materialstoday
Connecting the materials community

Intelligent Nanotechnology

Merging Nanoscience and Artificial Intelligence

Edited by Yuebing Zheng and Zilong Wu



ELSEVIER

INTELLIGENT NANOTECHNOLOGY

Materials Today: Connecting the Materials Community

INTELLIGENT NANOTECHNOLOGY

Merging Nanoscience and Artificial Intelligence

Edited by

YUEBING ZHENG
Walker Department of Mechanical Engineering and Texas Materials Institute, The University of Texas at Austin, Austin, TX, United States

ZILONG WU
Walker Department of Mechanical Engineering and Texas Materials Institute, The University of Texas at Austin, Austin, TX, United States



Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

Copyright © 2023 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

ISBN: 978-0-323-85796-3

For information on all Elsevier publications
visit our website at https://www.elsevier.com/books-and-journals

Publisher: Matthew Deans
Acquisitions Editor: Stephen Jones
Editorial Project Manager: Clodagh Holland-Borosh
Production Project Manager: Kamesh R
Cover Designer: Mark Rogers

Typeset by STRAIVE, India



Contributors

Trevon Badloe
Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

Yijun Bao
Department of Biomedical Engineering, Duke University, Durham, NC, United States

Keith A. Brown
Department of Mechanical Engineering; Physics Department and Division of Materials Science & Engineering, Boston University, Boston, MA, United States

Qian Chen
Department of Materials Science and Engineering; Department of Chemistry; Beckman Institute for Advanced Science and Technology; Materials Research Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, United States

Hongrui Cheng
College of Chemistry, Fuzhou University, Fuzhou; Fujian Science & Technology Innovation Laboratory for Optoelectronic Information, China

Liu Chen-Xu
Beijing Jiaotong University, School of Civil Engineering, Beijing, China

Frank Cichos
Leipzig University, Molecular Nanophotonics Group, Leipzig, Germany

Zhengyang Duan
Department of Electronic Engineering, Tsinghua University, Beijing, China

Zheyu Fang
School of Physics, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China

Paul S. Francis
School of Life and Environmental Sciences, Faculty of Science, Engineering and Built Environment, Deakin University, Waurn Ponds, VIC, Australia

Sheng Gao
Department of Electronic Engineering, Tsinghua University, Beijing, China

Yi Gao
Department of Electronic Engineering, Tsinghua University, Beijing, China

Yiyang Gong
Department of Biomedical Engineering; Department of Neurobiology, Duke University, Durham, NC, United States

Aldair E. Gongora
Department of Mechanical Engineering, Boston University, Boston, MA, United States

Grace X. Gu
Mechanical Engineering, University of California, Berkeley, Berkeley, CA, United States

Yu Gui-Lan
Beijing Jiaotong University, School of Civil Engineering, Beijing, China

Che-Lun Hung
Institute of Biomedical Informatics, National Yang Ming Chiao Tung University, Taipei, Taiwan

Santiago Muiños Landin
AIMEN Technology Centre, Smart Systems and Smart Manufacturing–Artificial Intelligence and Data Analytics Laboratory, Pontevedra, Spain

Chihun Lee
Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

Fushan Li
Institute of Optoelectronic Technology, Fuzhou University, Fuzhou, China

Yu Li
School of Physics, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing; Silicon Photonics R&D Center, CUMEC, Chongqing, China

Xing Lin
Department of Electronic Engineering, Tsinghua University, Beijing, China

Zhengchang Liu
School of Physics, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China

Cuicui Lu
Beijing Institute of Technology, Beijing, China

Yongfeng Lu
College of Chemistry, Fuzhou University, Fuzhou; Fujian Science & Technology Innovation Laboratory for Optoelectronic Information, China

G. Milano
Advanced Materials Metrology and Life Sciences Division, Italian National Institute of Metrology Research, Torino, Italy

Jaebum Noh
Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

Ravi Pradip
Universitat Politècnica De Catalunya, Barcelona, Spain

Junsuk Rho
Department of Mechanical Engineering; Department of Chemical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

C. Ricciardi
Applied Science and Technology Department (DISAT), Politecnico di Torino, Torino, Italy

Verda Saygin
Department of Mechanical Engineering, Boston University, Boston, MA, United States

Kelsey L. Snapp
Department of Mechanical Engineering, Boston University, Boston, MA, United States

Sunae So
Department of Mechanical Engineering; Graduate School of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

Zhongwei Xu
Institute of Optoelectronic Technology, Fuzhou University, Fuzhou, China

Charles Yang
Mechanical Engineering, University of California, Berkeley, Berkeley, CA, United States

Lehan Yao
Department of Materials Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States

Hongyi Yuan
Beijing Institute of Technology, Beijing, China

Jooyeong Yun
Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

Nianen Zhang
Beijing Institute of Technology, Beijing, China

Zhizhou Zhang
Mechanical Engineering, University of California, Berkeley, Berkeley, CA, United States

Yuanhui Zheng
College of Chemistry, Fuzhou University, Fuzhou; Fujian Science & Technology Innovation Laboratory for Optoelectronic Information, China

Preface

Nanoscience has been an active research field over the past several decades, triggered by the rising interest in exploring new science and technologies with materials at the nanoscale. Artificial intelligence (AI) has led to revolutionary development in many applications such as communication, computer vision, and speech/image recognition. Bridging the link between nanoscience and AI can provide a new generation of technologies that will have large impacts on our society. Some of the AI tools are being implemented by researchers to advance nanoscience and nanotechnology through AI-enhanced design, manufacturing, measurements, and applications of nanomaterials and nanodevices. Meanwhile, nanotechnology provides enormous data sets and new hardware platforms to advance the training of Al models and the implementation of AI algorithms.

With great contributions from some of the top researchers working in the interdisciplinary field, *Intelligent Nanotechnology: Merging Nanoscience and Artificial Intelligence* provides an overview of the advances in science and technology made possible by the convergence of nanoscience and AI. Section 1 focuses on AI-enhanced design, characterization and manufacturing of nanomaterials, and the use of AI to improve important material properties, with an emphasis on mechanical, photonic, phononic, electronic, and magnetic properties. The use of AI in the acquisition and analysis of data in experiments is explored as well. Section 2 includes chapters on AI technologies that have been enhanced through nanotechnology platforms. For example, memristors, neuromorphic computing, and artificial neural networks are included. Finally, Section 3 reviews the advances in applications enabled by the merging of nanotechnology and artificial intelligence, including examples from biomedicine, chemistry, anticounterfeiting, and automated research. We hope that this book will provide a bridge to further collaborations among researchers from different fields.

Yuebing Zheng
Zilong Wu

Acknowledgments

The editors acknowledge the financial support from the National Science Foundation (NSF-ECCS-2001650) and the National Institute of General Medical Sciences of the National Institutes of Health (DP2GM128446).

CHAPTER 1

Inverse design meets nanophotonics: From computational optimization to artificial neural network

Jaebum Noh[a,*], Trevon Badloe[a,*], Chihun Lee[a,*], Jooyeong Yun[a,*], Sunae So[a,b,*], and Junsuk Rho[a,c]

[a]Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea
[b]Graduate School of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea
[c]Department of Chemical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea

1. Computational inverse design

The methods for discovering optimized designs for photonic platforms using computational algorithms broadly fall into one of two categories, namely, gradient-based or evolutionary-based algorithms. Indeed, neural networks are trained using gradients and backpropagation, so are fundamentally related in this way. However, a key difference occurs in the fact that traditional computation inverse design methods are generally defined by a specific objective function to be optimized. This could be any required optical property, for example, a resonance at a specific wavelength or maximizing the broadband efficiency of a device. The design space can be as free or as limited as the researcher requires, which could be directly related to the fabrication constraints of certain materials or techniques. In this section, we will focus on the application of computational optimization without the integration of neural networks as an introduction to computational methods. We will describe the overall goals and methodologies of gradient and evolutionary-based algorithms and provide examples of recent results that showcase the power of numerical optimization techniques in the design of nanophotonic devices. A simple visual description of each method is provided in Fig. 1.

1.1 Gradient-based

Gradient-based optimization techniques rely on the computation of the local gradient of the objective function in relation to the designated design parameters. This gradient then guides the optimization algorithm in a certain direction toward a minimum (or maximum) for the objective function. A problem arises in that there is no guarantee that this local point of inflection in the gradients of the objective function is the global minimum,

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00001-9
Copyright © 2023 Elsevier Inc.
All rights reserved.

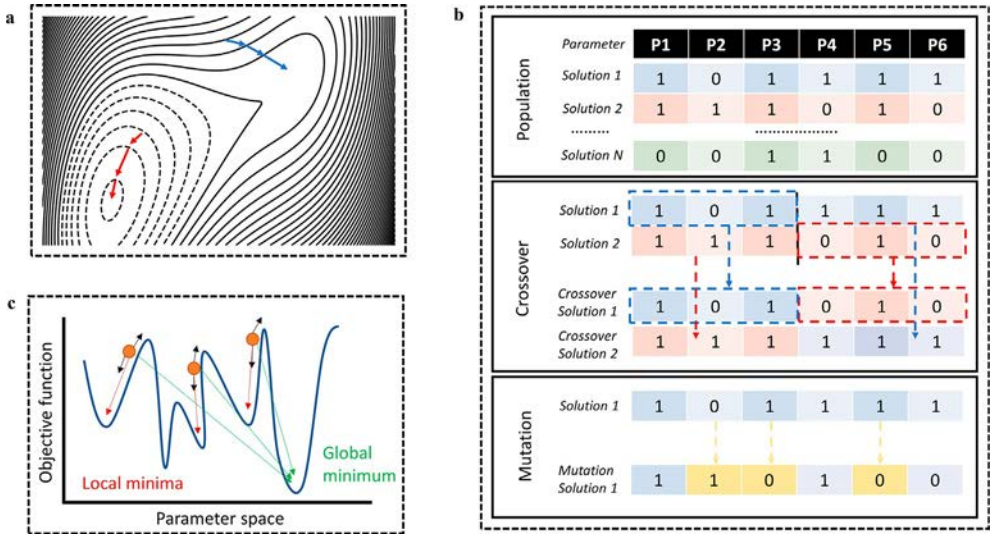


Fig. 1 Visual representations of numerical optimization methods. (A) Gradient descent method. The *blue* (gray in the print version) and *red* (dark gray in the print version) *arrows* represent optimization paths from different starting points, leading to different local minima. (B) Genetic algorithm. The population is made up of *N* unique solutions to the problem, here with six different parameters. After each has been given a score and ranked in terms of the fitness function, the best solutions are used to perform crossover, where the genes of two-parent solutions are spliced together to create a new child solution. With a specified random chance, some solutions also undergo mutation to create new solutions. This allows for the optimization to have some chance of escaping local minima. (C) Particle swarm. Three particles (*orange circles* (gray in the print version)) with a certain position and momentum explore the parameter space independently and are updated based on their local information as well as the information of the population as a whole, to hopefully end in the global minimum, i.e., the optimal solution to the problem. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which is, in general, what we wish to uncover to produce the most optimized design. If the objective function is convex, however, all local minima are also global minima, meaning that globally optimized solutions can be found. As described in Fig. 1A, after the gradient has been calculated at a specific location, the optimization algorithm takes a step in that direction. We need to determine how big of a step to take in the given direction, and how frequently to measure the gradient. The size of the step is an important factor, as too large of a step could lead to overshooting the minimum, while the frequency of measuring the gradient is determined by computational limitations. Of course, ideally, we would like to know the gradient at every point with infinitesimally small steps in order to uncover the optimum global solution, but this of obviously unrealistic in practice. Another important factor in gradient-based optimization methods is the starting guess in the parameter space. Due to these limitations, it is common to do multiple runs

with different starting points and step sizes, increasing the computational load. In an attempt to lighten the computational load on calculating the gradients of the objective function of high-dimensional parameter's spaces, the adjoint method has been introduced. This allows the gradients to be calculated using only a singular additional adjoint simulation, significantly reducing the computational load [1,2].

1.1.1 Adjoint optimization

Since the goal of optimization is to find the parameters in the design space that maximizes the achievable performance of the given objective, the traditional method of designing individual structures and arranging them in periodic arrays is unnecessary. In addition, by treating the device as a whole, with the only constraint being on the final output objective, issues such as coupling and interactions between neighboring structures are eradicated. This can allow adjoint optimization to find structures and layouts that avoid suboptimal designs. There have been various applications of adjoint optimization in the field of nanophotonics for designing large-scale and multilayered devices for applications such as waveguides [3–5] and grating couplers [6,7]. Some of the recent demonstrations will be introduced and the results of the devices designed using adjoint optimization will be discussed in this section.

Adjoint optimization allows for only one additional simulation in order to calculate the gradients of the objective function with regard to the design parameters. This allows for numerous structures to be optimized in a single pass. Using this framework, Backer numerically designed a double metalens system, holographic element, and an all-optical neural network, proving the versatility and robustness of adjoint optimization (Fig. 2A and B) [8]. Using a square TiO_2 nanostructure on the glass library, all of the aforementioned applications were realized. The achromatic, polarization-independent, doublet metalens was designed with an aperture of 800 μm and a back focal length of 2 mm. It showed a bandwidth of 160 nm across the visible spectrum. To confine the design parameter space to ensure that the nanostructures can be realistically fabricated, limits of the minimum and maximum size of the nanostructures were enforced. If the optimization gradient specified an unrealistic structure that is outside of the designated limits, then it was truncated to the limit. Full-wave 3D simulations with FDTD are computationally expensive, so all the propagations in both the forward and adjoint simulations were carried out in a 2D cross section of the total metalens to produce a library of results, and Fourier optics was used to propagate the resulting waves to the far field. Since the concept of the adjoint method is similar to backpropagation methods used in conventional machine learning algorithms, the system can also produce optical neural networks. Using an array of 10 detectors, it was shown that an adjoint optimized metasurface can successfully classify images of numbers with an accuracy of 84%. Unfortunately, these results were purely numerical; however, the power that adjoint optimization can bring to multiple tasks with large parameter spaces was clearly proven.

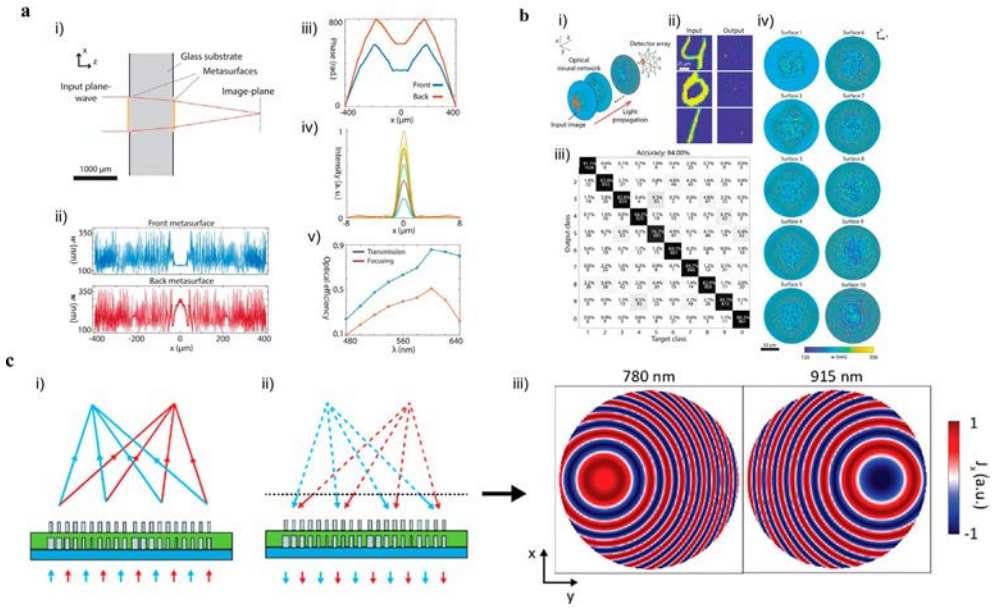


Fig. 2 Examples of nanophotonic devices designed using the adjoint method. (A) Inverse design of an achromatic doublet metalens. (i) Schematic of the design. (ii) The widths of the structures that make up the metalens. (iii) Phase profile across metasurface apertures at $\lambda = 640$ nm. (iv) Cross section of focal plane intensity at all designed wavelengths. (v) The optical efficiency of the metalens. (B) Optical neural network for handwritten digit classification. (i) Schematic of the design. (ii) Simulated test inputs and outputs of the optimized design. (iii) Confusion matrix for the MNIST dataset. (iv) The optimized metasurface designs. (C) A multiwavelength metalens design. (i) Schematic representation of the forward, (ii) the adjoint simulations in the adjoint optimization technique, and (iii) color maps of the surface electric current densities used as excitation sources in the adjoint simulations. *Panels A and B are adapted with permission from Mansouree, M., et al., Multifunctional 2.5D metastructures enabled by adjoint optimization. Optica 7 (2020) 77–84, copyright 2020 Optical Society of America. Panel C is adapted with permission from Backer, A.S., Computational inverse design for cascaded systems of metasurface optics. Opt. Express 27 (2019) 30308–30331, copyright 2019 Optical Society of America.*

In another demonstration of the adjoint optimization technique, the design spaces of metasurfaces have been increased from their general two-dimensional limitations, into 2.5D devices [9]. A 2.5D metalens that focuses different wavelengths of light at different focal points was designed and fabricated. By including the layers of metasurfaces into the design, the complexity is increased dramatically, as interactions between neighboring nanostructures in the same plane are also joined by interactions with the layers above and below. This could be an extremely arduous design to optimize for a researcher to optimize by hand; however, despite the increased degrees of freedom and complexity of the design space, utilization of the adjoint optimization technique produced devices with considerably higher efficiencies than those designed using conventional methods.

An FDTD solver was used to simulate the electric fields of plane waves of the desired wavelengths incident on the metasurface and the corresponding adjoint simulations, which are defined as the same structure illuminated by sources that are equal to the complex conjugate of the desired output fields (Fig. 2C). This can be understood as propagating the desired output fields back toward the metasurface. Through the calculated gradients, the dimensions of nanoposts were manipulated to find the optimized parameters. This was proved for both two and three layers of metasurfaces, with focusing efficiencies of over 50%. As the fabrication process of multilayer metasurfaces is much more involved, the performance increase must be worthwhile in practice. Here, the increase of focusing efficiency from 35% to 48%, to 60% for the single, bilayer, and trilayer metasurfaces proved the added complexity of fabrication is indeed worth the extra effort in order to produce optimized devices.

Expansions of adjoint optimization include using improved update rules [10,11], automatic differentiation [12], and extension into the nonlinear regime [13]. Limitations of adjoint optimization come from the discretization of the structure and the meshing accuracy of the full-wave simulations. Higher accuracy comes with a higher computational cost for both the forward and adjoint simulations. Although this optimization technique does indeed simultaneously optimize the whole device at each update step, full-wave simulations of large-scale devices require servers or clusters with large enough memory, or other simulation tricks must be utilized, such as making use of symmetry or simulating a 2D cross section.

1.1.2 Topology optimization

Topology optimization (TO) has recently been introduced to the field of nanophotonics, after being widely used in numerous fields for the design of physical systems and mechanical structures [14,15]. It is defined by the allocation of material either existing or not at specific points in a defined area, in order to satisfy an objective function. This means that each grid point is a design parameter, leading to the geometry being optimized in a pixel-like fashion. Again, in order to update the design over each iteration, a gradient needs to be calculated in relation to the parameter space, so TO is often coupled with the computationally efficient adjoint method, as described in the previous section. TO was introduced to the field of nanophotonics to design low-loss waveguide bends [16] and is now a staple method for the optimization of a magnitude of different nanophotonic devices [17]. See Ref. [18] for an in-depth review and description of the fundamentals and procedures of the optimization method, and Ref. [19] for a recent tutorial that includes FEM model examples.

Using TO, Ref. [20] designed reconfigurable metalenses that focus light at wavelengths that are an order of magnitude apart, and also an active metalens that shifts their focal length depending on the state of an active material. This was achieved using a 10-layer, multi-height TO method that has around 10 [4] degrees of freedom (Fig. 3A). By modeling

the problem in an axisymmetric domain using Maxwell's equations with a time-harmonic behavior, the full 3D equations were replaced with their simpler 2D counterparts. This allows the computational complexity to be reduced at the sacrifice of restricting the geometric freedom of the device. To further simplify the procedure, far-field transformations were applied rather than simulating the entire spatial domain between the lens and the focal point. The figure of merit was defined as the intensity of the electric field at the focal point, and the gradients were calculated using adjoint sensitivity analysis [21]. As an experimental validation of the technique, a single-layer metalens made up of variable heights was fabricated. The final design focused light with a wavelength of 1550 nm with a numerical aperture of 0.4 with an absolute power transmission of around 93%. Although this work shows the promise of TO, in reality, the fabrication of a 10-layer metasurface with variable heights is a great challenge, but one that could become a reality with improved nanofabrication methods such as 2.5D lithography or multi-photon polymerization.

Other work has shown that TO can be used to tailor the phase front of the light at will. In particular, using freeform optimization, Sell et al., proved numerically and experimentally that the optimization of a device for multiple wavelengths at the same time is possible [22] (Fig. 3B). In principle, the technique can be scaled to a large number of wavelengths at the cost of added computational effort. Using silicon, a beam steering device capable of steering N wavelengths into N unique diffraction orders, with an efficiency that scales as $1/N^{0.5}$ was designed. The optimization is carried out over the whole periodicity of the metasurface, meaning that the space that is being optimized is much larger than the wavelength it is being designed for. This is in stark contrast to the conventional method of designing subwavelength nanoantennas to build the total surface. This also allows the freedom of the structures to be of any shape that is required, without any constraint on local periodicities. This freedom in design space allows for curves and interconnected geometries that produce optical coupling and resonances that would otherwise be nearly impossible to design. One limitation of this work was the use of rigorous coupled-wave analysis (RCWA) solver for the simulations, as it requires the structures to be periodic. This could easily be overcome through the use of other methods of solving Maxwell's equations that do not rely on periodic boundary conditions [23]. Examples of multiwavelength and large-angle aberration-free metalenses have also been demonstrated using TO [24].

TO is able to uncover designs for optical properties that are very rare to find in nature. In particular, Shi et al. produced single-layered optical elements that show a considerable birefringence for different polarizations of incident light [25] (Fig. 3C). Using TO allows the device to have arbitrary geometries, which allow the degrees of freedom to be as big as the resolution of the individual pixels to be optimized. Elliptical birefringence is achieved in a single planar device by exploiting the incident angle of the light to break the symmetry of the structure.

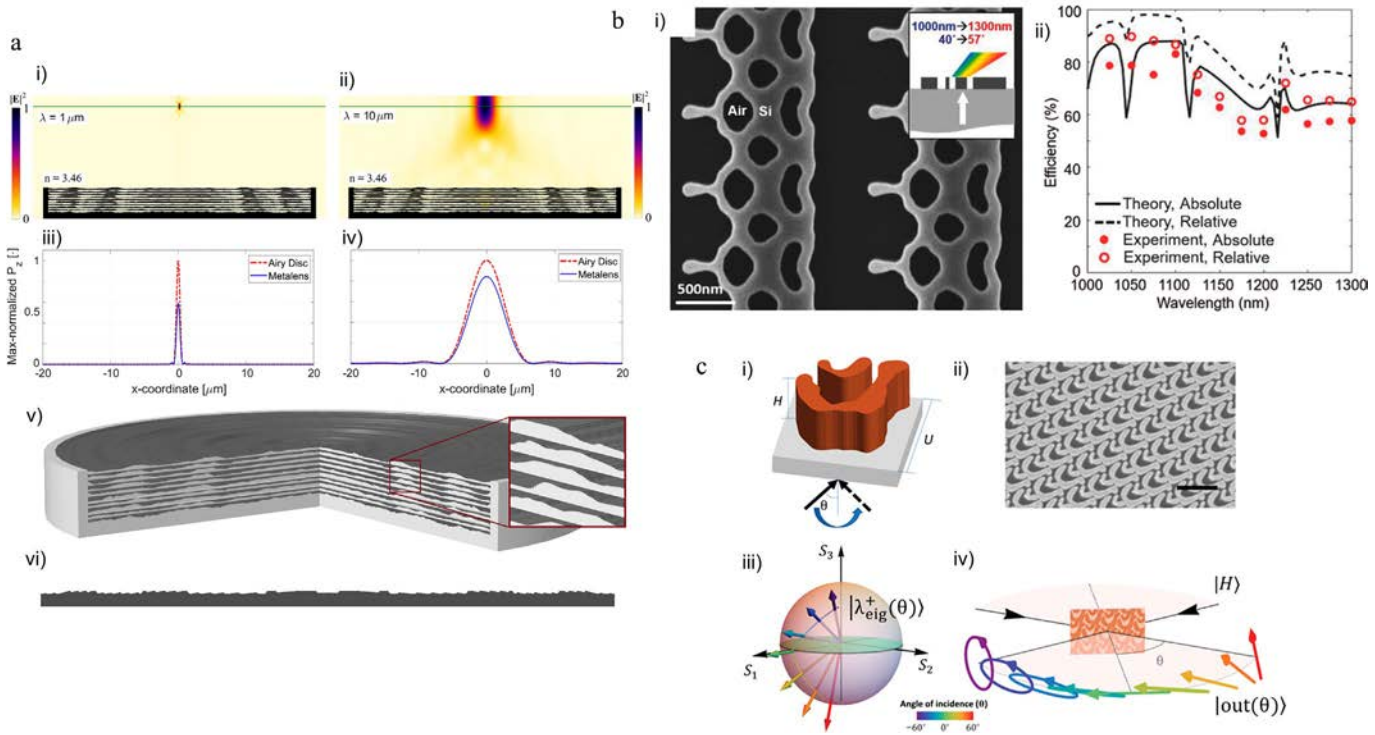


Fig. 3 Examples of nanophotonic devices designed using topology optimization. (A) Normalized electric field in the *x*–*z* plane through the center of the metalens for (i) = 1 μm and (ii) = 10 μm. (iii) The power in the *z*-direction through the focal plane normalized to the maximum of the Airy disc for = 1 μm and (iv) =10 μm. (v) A 3D illustration of the 10-layer metalens design. (vi) Cross section of single-layer reference design. (B) Experimental characterization of the broadband blazed metagrating. (i) Scanning electron microscopy image of the blazed metagrating operating between 1000 and 1300 nm for TM-polarized normally incident waves. (ii) Plot of the theoretical and experimental efficiencies vs wavelength for the device. (C) (i) Schematic of the optimized unit structure. (ii) SEM image of a fabricated sample. Scale bar, 1 μm. (iii) The *arrows* represent the angle-dependent eigen-polarization states of the device. *Different colors* correspond to different angles of incidence. As the angle of incidence varies, the device can be continuously tuned between linear and elliptical birefringence. (iv) Angle-dependent polarization generation. For a fixed incident polarization, the output polarization state changes continuously from right circular polarization through horizontal linear polarization to 45 degree linear polarization for varying angles of incidence. *Panel A is adapted with permission from Christiansen, R.E., et al., Fullwave maxwell inverse design of axisymmetric, tunable, and multi-scale multi-wavelength metalenses. Opt. Express. 28 (2020) 33854–33868, copyright 2020 Optical Society of America. Panel B is adapted with permission from Sell, D., Yang, J., Doshay, S., Fan, J.A., Periodic dielectric metasurfaces with high-efficiency, multiwavelength functionalities. Adv. Opt. Mater. 5 (2017) 1700645, copyright 2017 John Wiley and Sons. Panel C is adapted with permission from Shi, Z., et al., Continuous angle-tunable birefringence with freeform metasurfaces for arbitrary polarization conversion. Sci. Adv. 6 (2020) eaba3367, copyright 2020 CC BY-NC.*

Contrary to intuition, it has been shown that random initial geometries produce better results than using existing nanophotonic designs as the starting point [26]. TO usually has no constraints on the constituent parts of the final design, which could possibly lead to unrealistic designs. For example, if there are voids or floating elements, this may produce an output that maximizes the objective function, but it would be impossible to realize experimentally. Therefore, consideration of fabrication feasibility has also become a key point of improving devices designed using TO [27].

1.2 Evolutionary algorithms

In contrast to gradient-based optimization methods, evolutionary algorithms take inspiration from nature to reach designs that meet a certain design objective. The most well-known and often used examples of evolutionary algorithms are the genetic algorithm (GA) and particle swarm optimization (PSO). Both are known as metaheuristic approaches, meaning that they are able to find a reasonable solution to an optimization problem with limited information or computational cost. Although there is limited mathematical evidence that evolutionary algorithms converge to global optima, they do often reach extremely good results [28,29].

1.2.1 Genetic algorithm

The GA is inspired by Charles Darwin's theory of natural evolution, by focusing on the *selection*, *mutation*, and *crossover* of a population of solutions. A visual representation is shown in Fig. 1B. The optimization relies on the knowledge of good solutions within the population, in order to reach better solutions, while also keeping an amount of random mutation to allow for new solutions outside of the current direction to be accessible. The GA starts with a population of possible solutions, in nanophotonic design, this could be a set of geometric parameters for each individual. Following the lead of natural selection, the parameters are called genes, while a set of genes that form the solution for one individual is known as a chromosome. Each solution is then ranked in terms of its performance, with higher scores having a higher chance of being *selected* for the next iteration or generation. *Crossover* is another important part of the GA, whereby the genes of parent solutions are mixed together to form new solutions. This allows new child solutions to be formed from high-performance solutions, which should result in new, better solutions. *Mutation* relates to some genes being changed at random, allowing for a measure of diversity in the population, and avoiding premature convergence to a possible suboptimal solution. The important parameters of the GA are the population size, informative fitness function, and the amount of mutation and crossover. The population size is usually limited by the available computational resources, so it needs to be chosen to appropriately find a good solution in a reasonable amount of time. This simple yet intuitive technique has been applied to the field of nanophotonics with great success to design waveguides and photonic crystals [30], and some examples [31] will be discussed here.

In Ref. [32] the GA was utilized with some modifications in order to design phase-controlled metasurfaces. In this work, the concept of the crossover was omitted completely, in its place only mutation ensued. In order to reach an optimal solution, the amount of mutation was controlled to be smaller and smaller as the iterations were concluded. An elaborate fitness function was produced to create the exact desired response for the final solution. Furthermore, physical constraints were included to ensure that unphysical results were eradicated during the optimization process. This kind of supervision of the mutation is essential to keep the results in the desired domain. In another work, strong chiral dichroism in a plasmonic metasurface was demonstrated, using a micro-genetic algorithm [33]. The key feature of this adaptation is that instead of having mutation, there is a complete repopulation of the worst-performing solutions with completely new random solutions for faster convergence. Since the GA has no bounds in terms of the number of parameters that it can optimize in a single iteration, it owes well to freeform designs where the surface is discretized into individual pixels that can be chosen to include or exclude material, similar to the TO method [34]. But since no gradients are needed, there is no limitation on the fitness function to be differentiable. Here, the surface was discretized into a binary pattern as shown in Fig. 4A. Circular dichroism of 0.63 and 0.60 was reported for fabricated samples. The GA has also been used to choose the photonic elements from a library in order to produce the desired response [35]. The design of the individual structures was done by hand, but the phase profile was achieved using the optimization power of the GA. Using an adaptive GA allows for the problem to be split into subproblems that are solved in sequence, to reach the final solution [36]. The authors note that the large number of solutions that are produced with results that are close to the designed fitness function could also be employed later in data hungry methods such as a deep learning models. In a final example of using the GA to optimize the performance of nanophotonic applications, it was used to design perfect absorbers and color filters for single nanostructures and unit cells with multiple nanostructures (Fig. 4B) [37]. This study showed the limitations that single nanoparticles have when trying to optimize a solution with limited degrees of freedom. The colors achieved after increasing the unit cell to include multiple structures were greatly improved, as were the broadband absorption properties. For all of the examples presented here, it is interesting for researchers to delve into the underlying physics of the optimized solution to understand why certain designs are better than others, which in turn, helps researchers to uncover new ideas that can be applied successfully in new applications.

1.2.2 Particle swarm optimization

Particle swarm optimization gets its influence from the movement of birds and schools of fish. Each solution is treated as a particle that has a position and a momentum that is updated at each iteration. The update depends on the particular solution's local optimum, as well as the optima of the whole population as a whole, as demonstrated in Fig. 1C [38].

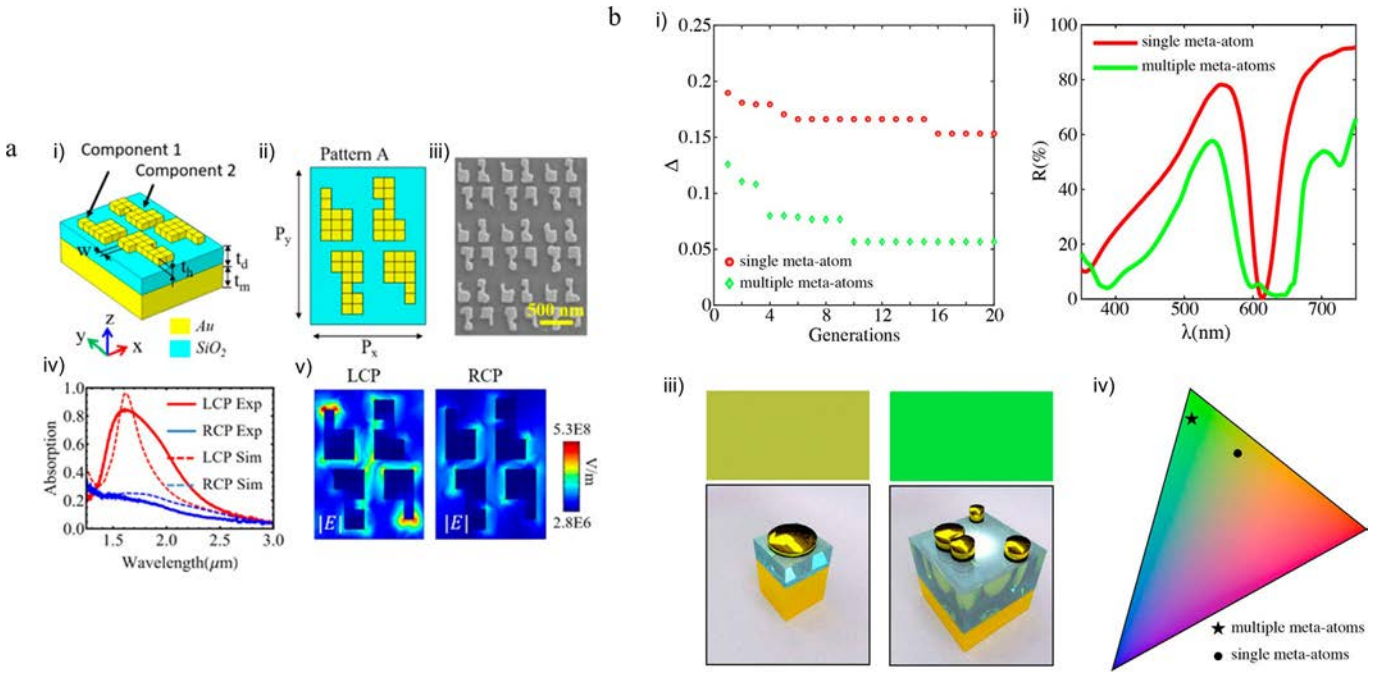


Fig. 4 Examples of nanophotonic devices designed using genetic algorithms. (A) (i) Schematic of the binary pattern for the chiral plasmonic metasurface. (ii) Top view of the pattern. (iii) SEM image of the fabricated chiral metasurface. (iv) Experimental and simulated absorption spectra under LCP and RCP at normal incidence. (v) Normalized electric field distributions at the resonant wavelength of 1.62 μm under LCP and RCP incidence. (B) (i) The distance between the target and designed color generated in the CIE diagram as a function of generations. *Red circles* (dark gray in the print version) represent the single-unit cell, *green diamonds* (gray in the print version) represent the unit cell with multiple nanostructures. (ii) Comparison of reflection spectra between the optimized samples of single (*green* (gray in the print version)) and extended (*red* (dark gray in the print version)) unit cells. (iii) Comparison of the generated colors and corresponding configurations. (iv) The generated colors on the CIE diagram. *Circles* denote the single nanostructure and *stars* represent the multiple nanoparticles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *Panel A is adapted with permission from Li, Z., Rosenmann, D., Czaplewski, D.A., Yang, X., Gao, J. Strong circular dichroism in chiral plasmonic metasurfaces optimized by micro-genetic algorithm. Opt. Express 27 (2019) 28313–28323, copyright 2019 Optical Society of America. Panel B is adapted with permission from Liu, C., Maier, S.A., Li, G., Genetic-algorithm-aided meta-atom multiplication for improved absorption and coloration in nanophotonics. ACS Photonics 7 (2020) 1716–1722, copyright 2020 American Chemical Society.*

The updates are governed by a learning rate that determines how much the velocity of the particle is changed at each update. This, along with the number of particles provides the most important parameters that need to be optimized for each problem and considered with regard to the available computing power.

In one example of using PSO for nanophotonic applications, Ref. [39] produced a freestanding metasurface of silicon for high-performance beam steering (Fig. 5A). A particle swarm algorithm was coupled with a full-wave simulation through the FDTD method. After optimization, both the forward transmissions were greatly increased, and the power of the deflected beam in the desired direction was improved dramatically. However, it was noted that the combination of 3D simulations is a huge bottleneck in this type of technique, as the total time needed for a single optimization was around 850 h. In another work using the PSO method, a silicon nitride *Y*-junction was developed with the goal of converting linearly polarized light into its opposite polarization (Fig. 5B) [40]. Conversion efficiencies of over 97% for both polarizations were reported and confirmed with numerical simulations. Many other examples have also been demonstrated, such as waveguides [41], and pixelated surfaces [42] with multiple objective functions [43].

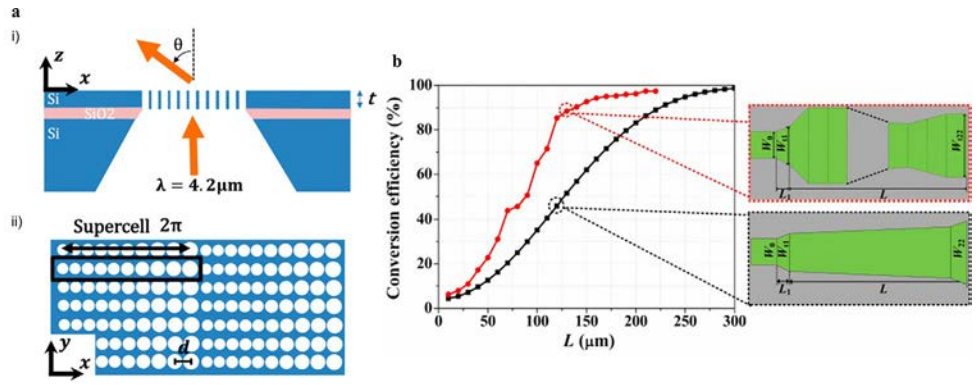


Fig. 5 Examples of nanophotonic devices designed using particle swarm. (A) (i) Schematic of a freestanding silicon nanohole array metasurface acting as a beam deflector with a deflection angle θ. (ii) Top-down view of the silicon nanohole array beam deflector metasurface constructed using repeating supercells. (B) Conversion efficiency as a function of the length *L* for the designed mode-conversion section and the adiabatic taper structure. *Panel A is adapted with permission from Ong, J.R., Chu, H.S., Chen, V.H., Zhu, A.Y., Genevet, P. Freestanding dielectric nanohole array metasurface for mid-infrared wavelength applications. Opt. Lett. 42 (2017) 2639–2642, copyright 2017 Optical Society of America. Panel B is adapted with permission from Zhang, B., et al., Particle swarm optimized polarization beam splitter using metasurface-assisted silicon nitride Y-junction for mid-infrared wavelengths. Opt. Commun. 451 (2019) 186–191, copyright 2019 Elsevier.*

2. Deep learning-based inverse design

Despite the achievements of computational optimization methods, as the design complexity increases, new challenges to inverse design arise. Specifically, solving highly complex problems may require numerous iterations to approach the optimum solution. In parallel, the sudden rise of artificial intelligence (AI), in particular machine learning (ML), over the past decade has had profound impacts on various fields of science. DL, powered by neural networks, has risen as one of the most widely used machine learning architectures [44]. The combination of the nanophotonics inverse design and DL is a notable approach that can solve the challenging problems of inverse design in nanophotonic based on data-driven ways.

Neural networks computationally model how learning happens in the neurons of the brain. As neurons in biological brains are connected in a complex manner with the strength of the connections being determined by certain external stimuli, the neurons computationally imitate this function. The simplest network, known as a perceptron, consists of several input neurons and one output neuron. Every adjacent neurons are connected by weight. Each node in the previous layer is multiplied by the weight and summed as one value. Next, by comparing the output of network and the output of actual data, the weights are adjusted in a direction to reduce this distance. The distance function between the perdition value of the network and the actual true value is named the cost function, and the method of updating the weights based on differentiating is named backpropagation. The detailed mathematical methods of forwarding and backpropagation are shown in Ref. [45]. As the connections are linear, only simple problems can be solved. Through the introduction of a non-linear activation function, which decides whether a neuron is activated or not by calculating the weighted sum of previous neurons, it becomes possible to learn nonlinear relationships.

To solve increasingly more complex problems, deep and wide multilayer perceptrons (MLPs) were introduced, known as DL. The adjective "deep" in DL refers to the use of multiple layers and nodes. In general, the more deep structure, the more complex relationships can be carried. Dramatic improvements in DL algorithms have been realized in recent years by overcoming chronic limitations such as vanishing gradients [46] and overfitting [47]. Along with the growth of accessible big data and advancements in computing power, ML has emerged as an extremely useful tool in various fields of computer science, including language processing [48], computer vision [44], image generation [49], and speech recognition [50]. There have also been attempts to combine DL with other disciplines, including manufacturing [51], microscopy [52], and nanophotonics [53–58].

DL models effectively learn to approximate the complex nonlinear relationships between input and output data, even without an analytic physical model. Most applications of DL in nanophotonics relate the design parameters of the nanostructures, such as the material, spatial location, and geometry with their optical responses such as the phase,

transmittance, or scattering power. Predicting the optical properties from the design parameters is known as forward modeling, which is generally calculated through computationally expensive full-wave simulations based on Maxwell's equations. After a model has been well trained, it learns to approximate these equations and can achieve extremely fast calculation times with considerable accuracy. On the other hand, inverse design has also been proven in nanophotonics using DL, by acquiring the geometric parameters that produce the desired optical properties. For inverse design using analytic or numerical methods, a compromise might be made between high-design feasibility and high spectral complexity; however, using DL can act as a bridge between the two, as shown in Fig. 6A [55]. In this section, we report the recent progress in DL-based inverse design in nanophotonics. The DL models used for inverse design can be divided into two types, deterministic and generative. The former always produces the same output for a given input, while the latter generates an output based on a probabilistic model which can open the possibility to design beyond the intuition of the researcher. In this section, we focus on the methodology of deterministic models and present the latest research. We then introduce examples of generative models such as autoencoders and generative adversarial networks for nanophotonic inverse design.

2.1 Deterministic neural network-based inverse design

Deterministic model is a neural network that does not contain randomness, once it has been trained, it acts as a function which produces the output for the given input. Representative deterministic neural networks are MLP, convolution neural network (CNN), and recurrent neural network (RNN). Each network has a specialized data type based on its structure and algorithms. For example, CNN is used to analyze images, and RNN is mainly used to analyze time-series data. In this way, researchers select the best network that can properly describe the given paired data. Before solving the inverse design problem, researchers evaluate whether the network can simulate Maxwell's equation properly through a forward model. In this section, we report the latest examples of forward and inverse models using MLPs and CNNs in nanophotonics.

2.1.1 Multilayer perceptron

MLPs consist of fully connected layers where every node of each adjacent layer is connected. It is the most basic architecture of artificial neural network (ANN). In Ref. [59], it was demonstrated that MLPs can be used to predict the scattering spectra of multilayered nanoparticles with variable thicknesses [60]. The relationship between the particle design and the scattering spectrum was sufficiently trained using a simple shallow network with a dataset of 50,000 samples. This research showed that well-trained MLPs can be indeed utilized as an approximator of Maxwell's equations to a reasonable accuracy with an increase in the speed, compared to conventional analytic simulations.

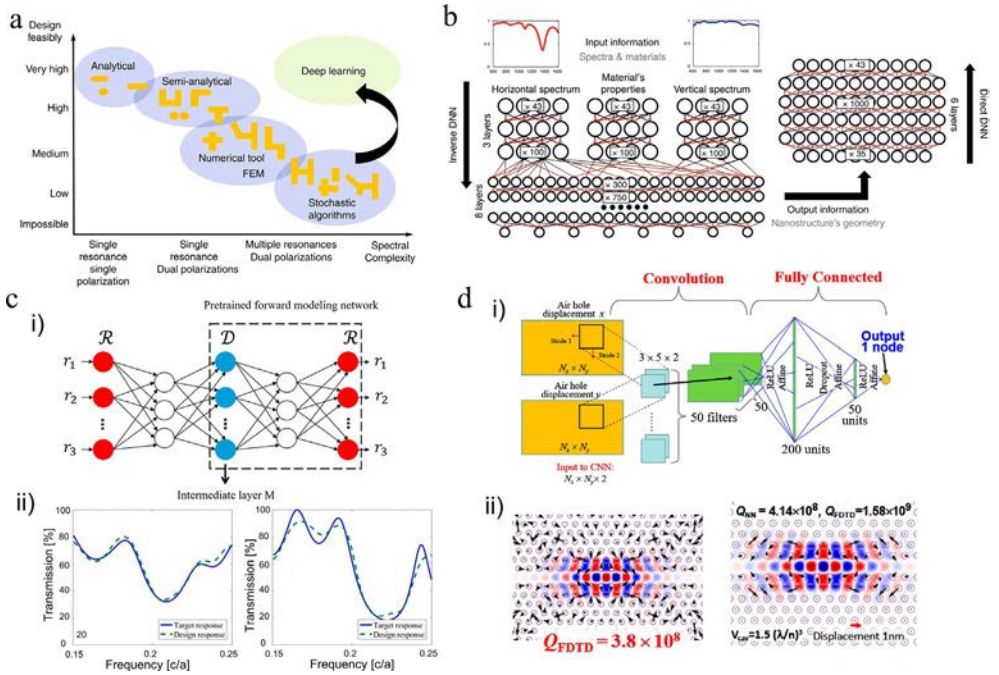


Fig. 6 Advantages of deep learning in photonics and photonic designs by deterministic neural networks. (A) Computational tools generally decrease the design feasibility as the optical response becomes more complex. Inverse design methods through deep learning can overcome this gap. (B) Two-phase network finds and verifies the inverse design from spectrum and material information to desired design parameters. The inverse network merged the expanded features of spectrum and material properties to consider corresponding geometry. After that, the direct network is also trained to improve the accuracy of the inverse network. (C) (i) The structure of tandem network which connects the inverse network and forward network sequentially. *R* denotes the optical response and *D* indicates the design parameters. To train the tandem network, a pre-trained forward network is necessary. (ii) The results of test data for the tandem network method. (D) 2D photonic crystal optimization by CNN. (i) The schematic of neural network which can learn between air hole displacement and Q factors. (ii) The heterostructure of 2D photonic crystal nanocavity. *Circle* denotes the air hole (110 nm radius) and *vector arrows* indicate the displacements of air holes. The *Q* factor before improved from 3.8×108 (*left*) and 1.58×109 (*right*) by deep learning optimization. The estimated MSEs are 2.64×10^{-3} (*left*) and 1.10×10^{-4} (*right*). *Panels A and B are adapted with permission from Malkiel, I., et al., Plasmonic nanostructure design and characterization via deep learning. Light: Sci. Appl. 7 (2018) 60, copyright 2018 CC BY 4.0. Panel C is adapted with permission from Liu, D., Tan, Y., Khoram, E., Yu, Z., Training deep neural networks for the inverse design of nanophotonic structures. ACS Photonics 5 (2018) 1365–1369, ACS. Panel D is adapted with permission from Asano, T., Noda, S., Optimization of photonic crystal nanocavities based on deep learning. Opt. Express 26 (2018) 32704–32717, copyright 2018 CC BY 4.0.*

Beyond forward modeling, MLPs were applied to inverse design. In Ref. [61], an inverse design method using a deep and wide was developed. The input was set to three types of series data consisting of 43 sampling points and the output to eight design features of an H-shaped antenna, including the existence of a leg, the angle of the structure, and the lengths of each part (Fig. 6B). With 15,000 simulated samples, the trained network was able to successfully design structures to satisfy the desired spectrum, which were verified using full-wave simulations.

Although MLPs have been shown to produce suitable candidate designs, without proper restrictions, the inverse design may be difficult to realize experimentally due to features that are too small or large, or have an unrealistically high aspect ratio [62]. This can be overcome by including extra constraints into the MLP cost function [63]. Before proceeding the inverse design, reasonable geometry design parameters were specified. After that, the cost function includes not only the difference between the prediction value and the true value, but also the difference between the recommended geometric parameters and pre-set geometric parameters. Through this cost function with geometrical constraint, it was possible to suggest an optimal design which was not too different from the pre-set geometry. The geometry vector includes the thickness, radius, and gap between the meta-atoms. As a result, the trained network was able to quickly recommend physically feasible meta-atoms that produce the required optical properties.

A fundamental issue of inverse design using DL occurs from the non-uniqueness of solutions; whereby various structures can produce the same optical properties. This hinders the training of the model as different solutions are given for the same inputs. In Refs. [64, 65], tandem networks were introduced. This works by combining a pre-trained forward modeling network with an inverse network to overcome the one-to-many problem [64,65]. As shown in Fig. 6C, the inverse network is sequentially connected to the pre-trained forward model. To train the inverse model, the inverse model error is combined with the error between the forward network result and the original target response. To find the proper multilayer film which can satisfy the desired transmission spectrum, the thicknesses of 20 layers were used as the design parameters, and 200 transmission spectral points were used as the optical properties [66]. The model was able to produce inverse designs that closely match the desired spectra, demonstrating that tandem networks can overcome the problem of non-uniqueness for inverse design in nanophotonics.

In parallel, researchers demonstrated that MLPs can be used to solve inverse design problems of various applications including integrated nanophotonic devices [67], color generation [68,69], photonic power splitters [70], absorbers [71], and core-shell particles [59,72]. In Ref. [72], an inverse design model was developed which considers both the thickness and material of core-shell nanoparticles using tandem networks. This result also demonstrates that DN-based methods can handle the multifunctional design by simply considering multi-design parameters (e.g., length, thickness, and material types) and

responses (e.g., amplitude, phase, and polarization) which is hard to achieve with purely guided by physical intuition.

2.1.2 Convolutional neural networks

Early studies expressed the design of nanostructures using several design features such as height, width, length, and angle. While this reduces the number of input features and promotes training, there is also a limitation that the results that inverse designs are restricted within the defined format. Therefore, to break these restrictions, the design can be considered as an image. The introduction of CNN can effectively consider spatial and temporal data [73]. Moreover, by sharing the weight with the filter, it can achieve better performance with fewer weights than MLPs. More specifically, the convolution layers, which calculate the convolution values of images and filters, were used to extract the key features. General conventional image filters have pre-defined values depending on their specific goals such as edge detection blurring, and denoising, but the filters in CNN do not have a specific form and it updates itself from the random state to reduce the errors of the cost function. With these trainable filters, it extracts the critical features from raw image data and uses it as an input for the following fully connected layers. This makes CNNs useful to analyze image data of nanophotonic structures or optical responses.

The combined CNN-based forward model and computational optimization methods were conducted to reduce the simulation time. In Ref. [74], polarization-insensitive plasmonic metasurface with 90% absorption was developed by combining GA and CNN. Another application is an example of maximizing the Q factor of photonic crystal [75]. To solve this Q factor maximize problem, more than 1000,000 simulations were needed. However, by replacing the simulation with a pre-trained CNN forward model, only 1000 simulations were conducted for gathering the training data (Fig. 6D).

There were also attempts to inverse design directly by using spectrums as input, and design parameters as output [76]. To optimize the plasmonic metasurface, 25,000 samples were used to describe the relationship between 300 spectral points and six design parameters. Particularly, this study also compares the effects of DL algorithms by changing network structures (MLP or CNN), and normalization methods (batch normalization is used or not). As a result, CNN-based model can recommend the proper geometry which satisfies the user suggested target absorption spectrum. It also demonstrates that the CNN with batch normalization has superior performance compared to the MLP-based model.

Finding the proper thin-multilayer-film metamaterial's thickness, which represents the desired spectral response, an inverse CNN model was applied [77]. This research used multiple spectra including reflectance and transmittance at various angles as input and the thickness of the film as output. To describe the relationship between input and output, CNN was developed with 200,000 samples. Moreover, by comparing CNN and computational optimization methods such as the least square method and GA, this study

demonstrated that CNN has advantages in terms of consuming time as the complexity of the problem increases (the number of film layers increases).

2.2 Generative neural network-based inverse design

Beyond determining the output from the trained network, generative models probabilistically create output from randomly generated noise. Moreover, generative networks can manage the one-to-many mapping problem without design parameter regularization or tandem networks. Here, we introduce the inverse design of nanophotonic structures using generative networks including autoencoders (AE) and generative adversarial networks (GAN).

2.2.1 Autoencoders

AEs are a type of neural network that is trained in an unsupervised manner, meaning that there is no requirement for prelabeled data. Generally, AEs are used to compress high-dimensional data into a lower dimensional space, known as the latent space. The latent variables are expected to contain only relevant and important features and neglect any extraneous information. As the word "latent" implies, it is not something that is explicitly observable but "hidden" and has to be implicitly learned through the framework of deep learning. A latent space is simply a probabilistic distribution of these latent variables. The notion of "space" can be understood as the configuration of such transformed data distribution. Conceptually, the closer the two data points in the data space are, the more likely they are to be considered similar. Under the name of "representative learning" in the data science field, studies on such latent spaces and ways to control and interpret them have gained a substantial understanding [78–81]. In addition, other constructive dimensionality reduction methods have also been highlighted in parallel [82,83]. Also in the field of nanophotonic design, attempts to construct a latent space into a desired distribution [84] or ways to effectively reduce the dimension of parameter space [85–88] have been explored.

AEs consist of an encoder, which compresses the input features, and a decoder, which recovers the original data from the latent space. Generally, AEs are made up of symmetric neural networks. The cost function is calculated as the difference between the restored output and the original input. The weights and biases of the connections in the network are adjusted to reduce the cost function. The latent variables of the AE were then used as the output of the network with the design parameters as the input. New designs can be generated by changing the latent variables uncovered by the AE, but in practice, there is low feasibility of the design as there are no bounds. Variational autoencoders (VAE) can overcome this low feasibility problem by introducing probabilistic perturbations [89]. Unlike AEs, the latent variables for VAEs consist of mean and standard deviation for each feature that can allow generating high-feasibility designs that follow the same distribution of the input data. In Ref. [90], three networks, such as a prediction model, a recognition

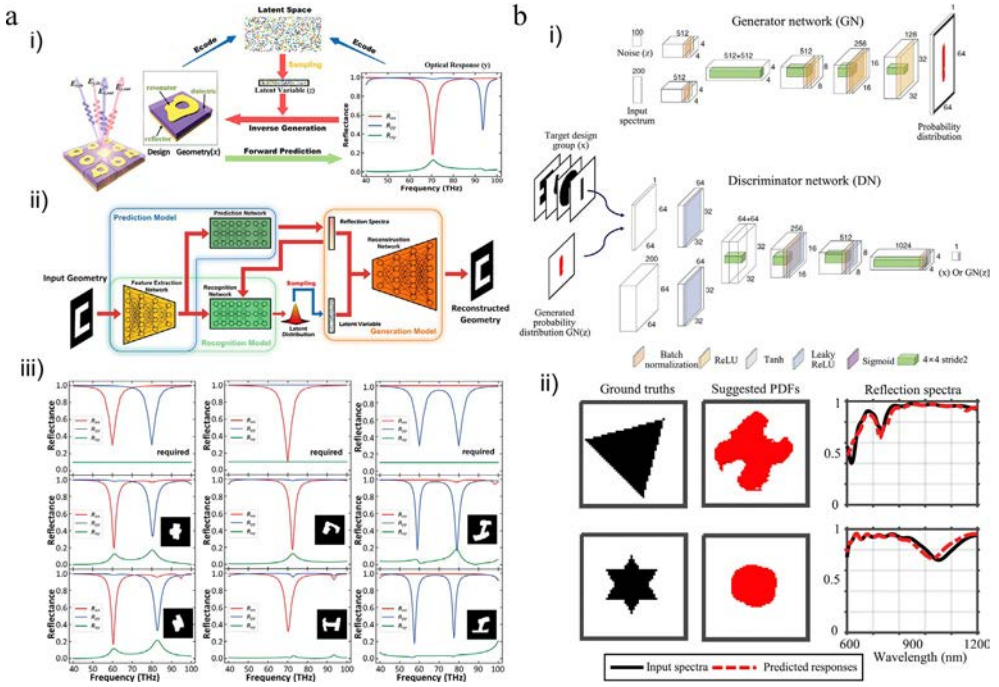


Fig. 7 Photonics designs by generative neural networks. (A) VAE-based meta-atom inverse design. (i) The framework of inverse generation and forward prediction with encoder and decoder. (ii) The architecture of deep generative model which consists of the prediction model, recognition model, and generation model. It can generate the geometry from desired reflection spectra. (iii) The verification result of on-demand inverse design. It contains various resonance cases including single resonance (*left*), only one resonance (*middle*), and dual resonances (*right*). The topmost figure indicates the required spectra and the rest two plots are inverse design generator results. (B) cDCGAN-based nanophotonic antennae design. (i) Schematic of the cDCGAN which consists of generative network and discriminate network. (ii) The inverse design model suggested geometry and its verification. *Red dot lines* (dark gray in the print version) show the prediction responses of the suggested probability distribution function (PDF) and *black solid lines* indicate the input spectra. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *Panel A is adapted with permission from Ma, W., Cheng, F., Xu, Y., Wen, Q., Liu, Y., Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. Adv. Mater. 31 (2019) 1901111, copyright 2019 John Wiley and Sons. Panel B is adapted with permission from So, S., Rho, J., Designing nanophotonic structures using conditional deep convolutional generative adversarial networks. Nanophotonics 8 (2019) 1255–1261, copyright 2019 CC BY 4.0.*

model, and a generation model, were used to solve the one-to-many mapping issues. The prediction model uses a one-to-one deterministic network to map the geometry to optical spectra and the recognition model encodes the spectra to latent variables as shown in Fig. 7A. The generation model uses both the optical spectra and latent variables to recommend designs that satisfy the required optical response. As a result, an on-demand

inverse design from the desired reflection response was achieved. However, for generative models based on VAEs, the generated images can be blurry as the distribution of the training data is described with probabilistic functions. For example, even if there is only a small error in the standard deviation of the trained VAE, it can deviate a lot from the sharp distribution of real data [91]. Therefore, to obtain sharp generated images, another approach is needed.

2.2.2 Generative adversarial networks

GANs, originally designed by Ian Goodfellow in 2014, are powerful generative models which consist of two neural networks, a generator, and a discriminator that contest each other as zero-sum game [92,93]. Firstly, the generator creates candidates from random noise to deceive the discriminator. The discriminator then tries to distinguish "real" results from the "fake" samples made by the generator. As a result, after the iterative and adversarial training of the two networks, GANs can generate samples that are indistinguishable from the original data. Though the original purpose of GANs is also unsupervised learning, it also has been proven that they can be trained in a semi-supervised [94] or fully supervised manner [95]. GANs also produce sharper images compared to VAEs [96]. In nanophotonic inverse design, supervised learning of GANs is generally applied to produce designs that satisfy a specific condition, i.e., a desired optical response. These GANs are known as conditional GANs (cGAN).

An example of using GANs in nanophotonics was conducted using three models: a generator, a simulator, and a discriminator [97]. The primary goal is to train the generator to inversely design structures that achieve the required target spectra. For this, the generator accepts a target response to create a pattern which becomes the input of the pretrained simulator. Next, the simulator evaluates the design. And then, the discriminator calculates the distance between the geometry of the generator. At that time, both of generator and discriminator used the response data to impose conditions. This guides the generator to produce similar patterns to the input geometric data. As a result, with the 8000 samples, GAN generated the geometry which can satisfy the randomly generated transmittance spectra. It also means that GAN demonstrated the possibility for high feasibility on-demand design even with large degrees of freedom. By utilizing convolution layers in GANs, a conditional deep convolutional generative adversarial network (cDCGAN) can improve their performance for image data [98]. This was applied to generate nanophotonic antennae that are not limited by predefined parameters. Both the generator and discriminator used a deep convolution network to increase the efficiency of pattern and probability distribution (Fig. 7B). cDCGAN proved that neural network-based inverse design methods can produce various and different designs which have a similar response (many to one) with high-fabrication feasibility from extremely large degrees of freedom of design ($2^{64\times64}$) [99].

3. Deep learning merged with computational optimization

The major contribution of ANNs is to serve as effective function approximators, which is a powerful tool in modeling the complicated relationships in data. When a well-trained network is appropriately placed into the inverse design pipeline, the process can be accelerated and designs with enhanced performance have been uncovered. Indeed, deep learning aided inverse design is an intricate process, as it requires the careful formulation of the loss function, preparation of the data, and selection of an appropriate model with the correct hyperparameters. Recently, a wide variety of hybridized frameworks coupling computational optimization methods with generative models such as GANs or different types of AEs have been actively introduced [84,100–102][103]. One approach is to train generative neural networks on sets of pre-optimized designs made by computational optimization methods. This affects the training efficiency of the model as the network exclusively learns from high-performance designs. Another intuitive method is to refine the output of a generative neural network with an iterative optimization technique, most commonly TO. While the two aforementioned schemes combine DL and computation optimization as a separate process, it is also possible to incorporate gradient-based optimization directly inside the network or to have a generative network construct a latent space and then have other computational optimization techniques to search within that space to produce an inverse design with the desired optical properties. In this section, we will describe some of these hybridized inverse design methods, especially those combined with generative neural networks that have shown notable advancements in terms of performance and efficiency. In addition, the pros and cons of each method will be discussed in detail.

3.1 Generative neural networks combined with topology optimization

In the early stages of adopting generative neural networks in nanophotonic inverse design, research was mostly focused on the functionality of the network to effectively encode the implicit topological geometries and decode on-demand input values into device designs [97] to systematically reduce the human intuition-driven procedure. While various new structures were uncovered within such frameworks, it is difficult to reach the goal of finding near-optimal performance designs of a specific objective. On the other hand, in the hybridized approach, the networks are trained only using high-quality data obtained through other optimization methods. This optimized training data helps the network to focus on learning geometric features of high-performance designs rather than spending resources on exploring other parts of the design space which are far from optimal.

A number of examples of designing free-form metagratings have been demonstrated. In Ref. [102], a framework using a cGAN was effectively trained on topology-optimized metagratings, enabling the rapid production of high-efficiency electromagnetic wave

deflection devices. The generator takes the operating wavelength and diffraction angle along with a random noise vector as inputs to produce 2D cross-sectional designs of the devices, while the discriminator decides if samples are "real" or "fake." The network was able to capture and generalize the topological features and efficiently generate devices operating over a continuous range of wavelengths and diffraction angles. This generated family of devices was further pushed toward its performance limit by applying TO once more (Fig. 8A). Application of TO on the output of the network not only refines its performance but also carries the aspect of providing additional data for GAN training, therefore, feeding the network with even higher quality data.

This method was developed further by progressively adding network layers and data during training [103]. Specifically, the network learned spatially coarse features from the low-resolution training set in its initial iterations and as additional neural layers were progressively added, it focused on the fine, detailed features from the augmented high-resolution training set. Additional high-performance network-generated devices were added to the training set over the course of multiple training cycles while removing relatively low-performance training data. The improved results can be found in Fig. 8B. This scheme of augmenting and refining the training set shares the same strategy of pushing up the performance level while still maintaining its ability to produce a diverse distribution of devices.

The provision of a pre-optimized initial dataset by TO and other methods has a fair amount of impact on the learning efficiency of the network. However, the requirement of multiple TO iterations in these setups remains the major computational bottleneck. To overcome this problem, ways to effectively generate initial training sets and frameworks that avoid the need for computationally heavy post-processing are required. In Ref. [104], a dataless training framework that directly integrates TO inside the conditional GAN pipeline was demonstrated. As described in Fig. 8C, in a showcase of designing silicon nano-ridge metasurface, the framework utilized an RCWA solver as the discriminator and utilized the adjoint variable method to calculate the gradients. By doing so, the network is able to calculate physics-driven gradients and directly incorporate them into the loss function. This method was shown to be capable of globally surveying the design space as the network maps the latent vector onto the original, full-dimension design space in a training course of sampling and shifting. Unlike previous models which are highly training-set dependent as they operate in a training scheme to fit the given data distribution, cumulative sampling of the network shifts and refines the distribution toward favorable distribution, continuously pushing it up to a globally optimal point. The proposed method was found to create designs with higher diffraction efficiencies with less computational cost compared to those created by conventional adjoint-based TO through statistical analysis. Further advancements such as solving problems with constraint re-parameterization [105] and multi-objective tasks [106] have been undertaken as subsequent research to demonstrate the power of this inverse design method.

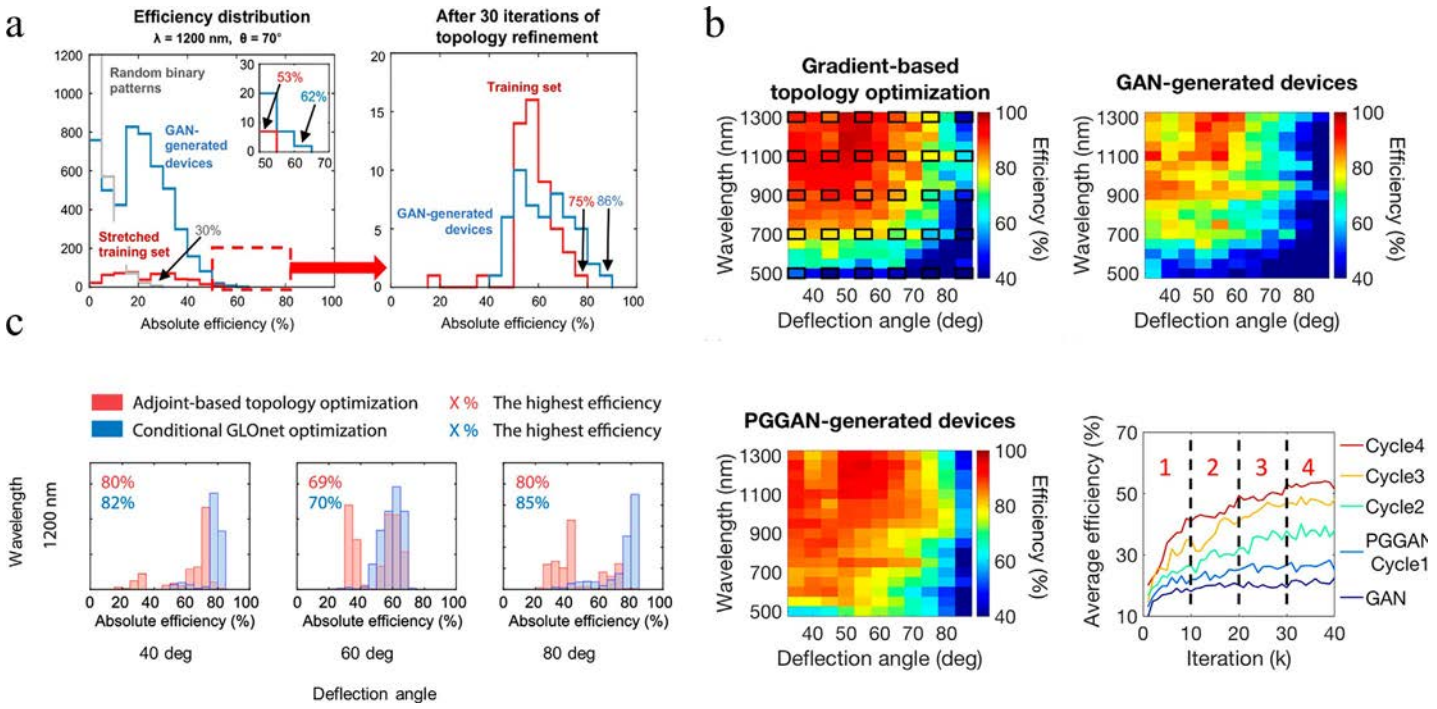


Fig. 8 Examples of nanophotonic device designed using generative neural network combined with topology optimization. (A) Efficiency histogram comparing randomly generated patterns and GAN-generated patterns and the refined patterns. (B) Plot of the highest efficiency devices at each wavelength-deflection angle pair for device designed using gradient-based topology optimization (*top left*), GAN (*top right*), and progressively growing GAN (*bottom left*). *Bottom right* is the average generated device efficiency as a function of training iteration. (C) Efficiency histogram comparing topology optimized patterns (*blue*) and conditional GAN generated patterns (*red*). *Panel A is adapted with permission from Jiang, J., et al., Free-form diffractive metagrating design based on generative adversarial networks. ACS Nano 13 (2019) 8872–8878, copyright 2019 American Chemical Society. Panel B is adapted with permission from Wen, F., Jiang, J., Fan, J.A., Robust freeform metasurface design based on progressively growing generative networks. ACS Photonics 7 (2020) 2098–2104, copyright 2020 American Chemical Society. Panel C is adapted with permission from Jiang, J., Fan, J.A., Global optimization of dielectric metasurfaces using a physics-driven neural network. Nano Lett. 19 (2019) 5366–5372, copyright 2019 American Chemical Society.*

3.2 Generative neural networks combined with metaheuristic optimization

Along with the development of generative models, the capability of neural networks to compress high-dimensional data into a low-dimensional latent space has been shown. If the latent space is well defined and captures the key features of the design parameters, performing computational optimization methods on this compressed space can be more efficient than performing on the original design space. Further, it is possible to perform global optimization on the latent space using various metaheuristic methods [107–109].

One example of this approach is to use an adversarial autoencoder (AAE). AAE is a combination of AE and GAN and is composed of three networks: an encoder, a decoder, and a discriminator [79]. The discriminator is responsible for performing adversarial learning. Unlike previously introduced VAE, where the latent space is set to be a standard normal distribution, the discriminator of an AAE is able to force the latent space into a user-defined distribution, resulting in the construction of a more compact, continuous latent space. The advantage of a continuous hyper-parametric space is that the interpolation of data points, which is necessary for the physical interpretation or the implementation of computational optimization methods, can be done with ease. A method of controlling and shaping the latent space using AAE was demonstrated to inversely design a high-efficiency metasurface-based thermal emitter [84] (Fig. 9A). The final output devices of this AAE generation with an extra level of TO refinement (AAE + TO) method showed excellent performance of thermal efficiency up to 98%, whereas only a maximum of 92% efficiency device was retrieved via direct TO. In a successive study on the same task, a realization of global optimization was achieved through a differential evolutionary (DE) algorithm [101]. DE, a population-based metaheuristic algorithm, uses multiple agents to search the solution space and evolutionally converge to a global solution [110]. Sets of corresponding efficiencies of the compressed design space vectors were used to update the position of the agents. The combination of a conditional AAE (c-AAE) and DE algorithm was able to greatly accelerate the design process to be almost 2.2 times faster than AAE + TO to produce designs with comparable performance. The authors took a step further to enhance the device efficiency by latent space regularization, performing a class labeling of high and low efficiency of devices (Fig. 9B). Over the course of training, the design distribution was clustered into two levels. Devices that were optimized by conducting DE only on the high-efficiency designs showed better performance than the un-regularized case. This example clearly shows the convenience that can be brought by a proper representation of the design space, implying a further possibility of other global optimizers or any kind of regularization method to be implemented. Expanded demonstrations of these include the adoption of a support vector machine (SVM), a classifying algorithm in machine learning, to perform additional analysis on the AE retrieved geometry manifold [111] and a merged framework using SVM with the PSO method to output a metasurface structure of the desired property [112].

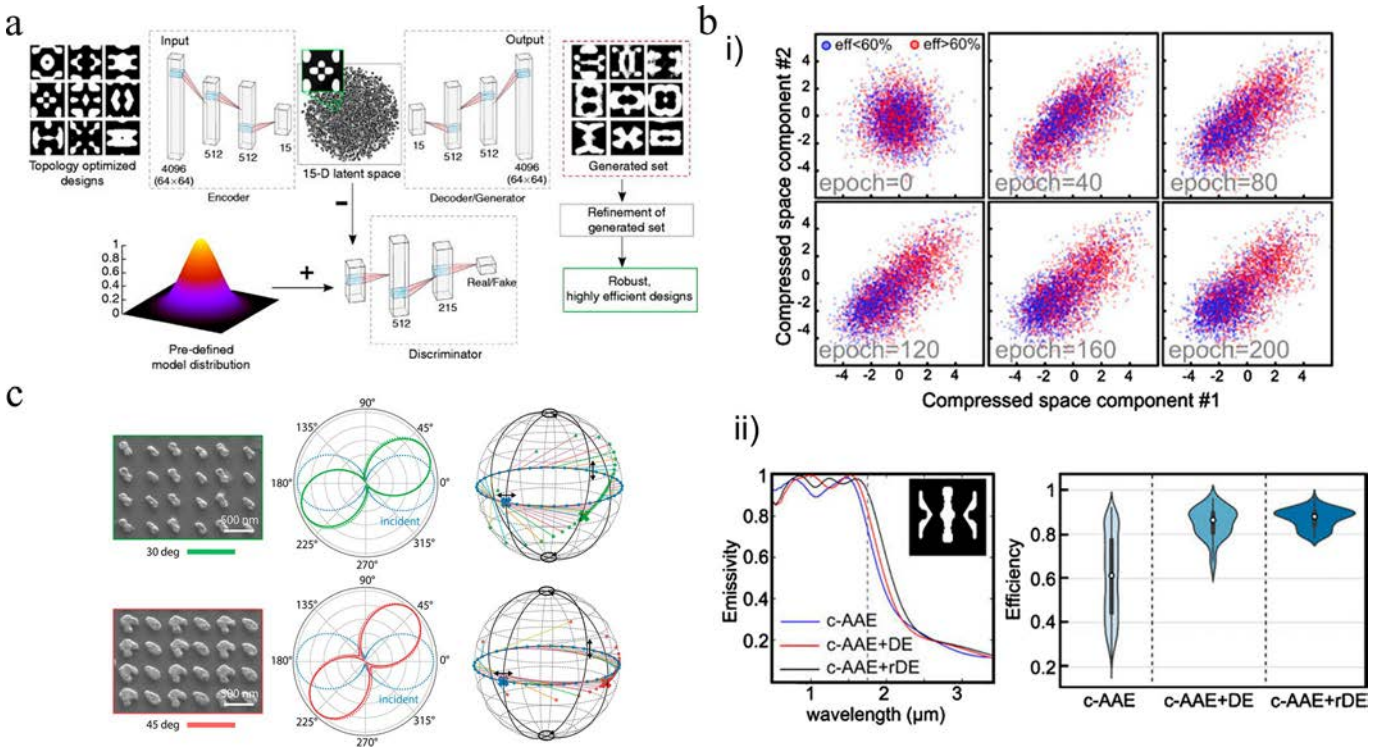


Fig. 9 Examples of nanophotonic device designed using generative neural network combined with metaheuristic searching methods. (A) The training schematic of AAE network model. The latent space is formed as a user-defined distribution by a discriminator. (B) (i) Evolution of regularized compressed design space during training. (ii) Comparison of spectral emissivity of the best design achieved by c-AAE, c-AAE + DE, and c-AAE + DE with regularized compressed design space (c-AEE + rDE) method (*left*). The violin plot of three design sets generated by c-AAE, c-AAE + DE, and c-AAE + rDE (*right*). (C) Fabricated, measured, and simulated results of polarization conversion by optimized metamolecules. *Panel A is reprinted with permission from Kudyshev, Z.A., Kildishev, A.V., Shalaev, V.M., Boltasseva, A., Machine-learning-assisted metasurface design for high-efficiency thermal emitter optimization. Appl. Phys. Rev. 7 (2020) 021407, copyright 2020 American Institute of Physics. Panel B is adapted with permission from Kudyshev, Z.A., Kildishev, A.V., Shalaev, V.M., Boltasseva, A. Machine learning-assisted global optimization of photonic devices. Nanophotonics 10 (2020) 371–383, copyright 2020 CC BY 4.0. Panel C is reprinted with permission from Liu, Z., et al., Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques. Adv. Mater. 32 (2020) 1904790, copyright 2019 Wiley.*

Other research has shown that metaheuristic methods can be well coupled with other generative neural networks and their hyper-parametric space. A concrete example of integrating an evolutionary algorithm-based searching method on a latent space was shown in a task of optimizing metamolecules in Ref. [113]. Metamolecules, a term for a unit set of multiple, differently shaped meta-atoms, possess features such as anomalous deflection which identical meta-atom consisting material cannot have. The goal of the work was to efficiently optimize the metamolecule configuration for arbitrary manipulation of the polarization and wavefront of light. In this framework, a generative network that is able to decode the high-quality 2D patterns from a latent variable in a pixel-by-pixel manner and an evolutionary algorithm were combined to globally search within the latent space. The algorithm performs selection, reproduction, and mutation in a round robin fashion for all species with defined fitness scores. The species are the latent vectors of meta-atoms and the fitness score is calculated based on the polarization state of the design, its ellipticity, and phase and intensity information of the produced electric field. Two cases of polarization conversion of two meta-atom metamolecules and metasurfaces with gradient phase distributions were demonstrated. As shown in Fig. 9C, the simulated and experimentally measured data were in excellent agreement, proving the efficacy of the proposed design framework.

Because the inverse design of nanophotonic devices is generally a highly complicated non-convex problem, performing gradient-based optimization method right from the start is not always the best policy as it is easy to get stuck at a local optimal point. To guarantee a globally optimal solution, therefore, global searches must be conducted. However, it is almost impossible to probe every point in the original design space. The methods introduced so far aim to reduce the high-dimensional parametric space into an appropriate hyper-parametric space, which is much more manageable. Numerous examples have shown these integrated approaches using generative neural networks and metaheuristic examples are capable of pushing up the performance level of the designs with considerable efficacy. Improvements could be made to the compressed design space representation and the effective searching techniques to provide a promising step toward the inverse design of globally optimized devices.

Acknowledgments

This work was financially supported by the National Research Foundation (NRF) grants (NRF-2019R1A2C3003129) funded by the Ministry of Science and ICT (MSIT) of the Korean government. S.S. acknowledges the NRF *Sejong* Science fellowships (NRF-2022R1C1C2009430) funded by the MSIT of the Korean government.

References

[1] T.W. Hughes, I.A.D. Williamson, M. Minkov, S. Fan, Forward-mode differentiation of Maxwell's equations, ACS Photonics 6 (2019) 3010–3016.

[2] R. Safian, N.K. Nikolova, M.H. Bakr, J.W. Bandler, Feasible adjoint sensitivity technique for em design exploiting Broyden's update, in: IEEE MTT-S International Microwave Symposium Digest, vol. 1, IEEE, 2003, pp. 299–302.

[3] C.M. Lalau-Keraly, S. Bhargava, O.D. Miller, E. Yablonovitch, Adjoint shape optimization applied to electromagnetic design, Opt. Express 21 (2013) 21693–21701.

[4] S.J. Kim, C. Lee, S. Jeon, J. Park, S.J. Kim, Subwavelength sorting of full-color based on anti-hermitian metasurfaces, Nanophotonics 10 (2021) 967–974.

[5] A.C.R. Niederberger, D.A. Fattal, N.R. Gauger, S. Fan, R.G. Beausoleil, Sensitivity analysis and optimization of sub-wavelength optical gratings using adjoints, Opt. Express 22 (2014) 12971–12981.

[6] A. Michaels, E. Yablonovitch, Inverse design of near unity efficiency perfectly vertical grating couplers, Opt. Express 26 (2018) 4766–4779.

[7] S. Hooten, et al., Adjoint optimization of efficient CMOS-compatible Si-SiN vertical grating couplers for DWDM applications, J. Lightwave Technol. 38 (2020) 3422–3430.

[8] A.S. Backer, Computational inverse design for cascaded systems of metasurface optics, Opt. Express 27 (2019) 30308–30331.

[9] M. Mansouree, et al., Multifunctional 2.5D metastructures enabled by adjoint optimization, Optica 7 (2020) 77–84.

[10] G.B. Hoffman, et al., Improved broadband performance of an adjoint shape optimized waveguide crossing using a Levenberg-Marquardt update, Opt. Express 27 (2019) 24765–24780.

[11] N. Zhao, S. Boutami, S. Fan, Accelerating adjoint variable method based photonic optimization with Schur complement domain decomposition, Opt. Express 27 (2019) 20711–20719.

[12] M. Minkov, et al., Inverse design of photonic crystals through automatic differentiation, ACS Photonics 7 (2020) 1729–1741.

[13] T.W. Hughes, M. Minkov, I.A.D. Williamson, S. Fan, Adjoint method and inverse design for nonlinear nanophotonic devices, ACS Photonics 5 (2018) 4781–4787.

[14] M.P. Bendsøe, O. Sigmund, O. Sigmund, Topology optimization by distribution of isotropic material, in: Topology Optimization, Springer Berlin Heidelberg, 2004, pp. 1–69.

[15] M.P. Bendsøe, Optimal shape design as a material distribution problem, Struct. Multidiscip. Optim. 1 (1989) 193–202.

[16] J.S. Jensen, O. Sigmund, Systematic design of photonic crystal structures using topology optimization: low-loss waveguide bends, Appl. Phys. Lett. 84 (2004) 2022–2024.

[17] D. Sell, J. Yang, S. Doshay, R. Yang, J.A. Fan, Large-angle, multifunctional metagratings based on freeform multimode geometries, Nano Lett. 17 (2017) 3752–3757.

[18] J.S. Jensen, O. Sigmund, Topology optimization for nano-photonics, Laser Photonics Rev. 5 (2011) 308–321.

[19] R.E. Christiansen, O. Sigmund, Inverse design in photonics by topology optimization: tutorial, J. Opt. Soc. Am. B 38 (2021) 496–509.

[20] R.E. Christiansen, et al., Fullwave maxwell inverse design of axisymmetric, tunable, and multi-scale multi-wavelength metalenses, Opt. Express 28 (2020) 33854–33868.

[21] D.A. Tortorelli, P. Michaleris, Design sensitivity analysis: overview and review, Inverse Prob. Eng. 1 (1994) 71–105.

[22] D. Sell, J. Yang, S. Doshay, J.A. Fan, Periodic dielectric metasurfaces with high-efficiency, multiwavelength functionalities, Adv. Opt. Mater. 5 (2017) 1700645.

[23] T. Phan, et al., High-efficiency, large-area, topology-optimized metasurfaces, Light: Sci. Appl. 8 (2019) 48.

[24] Z. Lin, C. Roques-Carmes, R.E. Christiansen, M. Soljačić, S.G. Johnson, Computational inverse design for ultra-compact single-piece metalenses free of chromatic and angular aberration, Appl. Phys. Lett. 118 (2021) 041104.

[25] Z. Shi, et al., Continuous angle-tunable birefringence with freeform metasurfaces for arbitrary polarization conversion, Sci. Adv. 6 (2020) eaba3367.
[26] J. Yang, J.A. Fan, Topology-optimized metasurfaces: impact of initial geometric layout, Opt. Lett. 42 (2017) 3161–3164.
[27] Y. Augenstein, C. Rockstuhl, Inverse design of nanophotonic devices with structural integrity, ACS Photonics 7 (2020) 2190–2196.
[28] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, ACM Comput. Surv. 35 (2003) 268–308.
[29] G. Angeris, J. Vučković, S. Boyd, Heuristic methods and performance bounds for photonic design, Opt. Express 29 (2021) 2827–2854.
[30] J. Jiang, J. Cai, G.P. Nordin, L. Li, Parallel microgenetic algorithm design for photonic crystal and waveguide structures, Opt. Lett. 28 (2003) 2381–2383.
[31] S. So, et al., Highly suppressed solar absorption in a daytime radiative cooler designed by genetic algorithm, Nanophotonics 11 (2022) 2107–2115.
[32] Y. Fan, et al., Phase-controlled metasurface design via optimized genetic algorithm, Nanophotonics 9 (2020) 3931–3939.
[33] Z. Li, D. Rosenmann, D.A. Czaplewski, X. Yang, J. Gao, Strong circular dichroism in chiral plasmonic metasurfaces optimized by micro-genetic algorithm, Opt. Express 27 (2019) 28313–28323.
[34] M.H. Heidari, S.H. Sedighy, Broadband wide-angle polarization-insensitive metasurface solar absorber, J. Opt. Soc. Am. A 35 (2018) 522–525.
[35] H. Li, G. Wang, T. Cai, H. Hou, W. Guo, Wideband transparent beam-forming metadevice with amplitude- and phase-controlled metasurface, Phys. Rev. Appl. 11 (2019), 014043.
[36] S. Jafar-Zanjani, S. Inampudi, H. Mosallaei, Adaptive genetic algorithm for optical metasurfaces design, Sci. Rep. 8 (2018) 11040.
[37] C. Liu, S.A. Maier, G. Li, Genetic-algorithm-aided meta-atom multiplication for improved absorption and coloration in nanophotonics, ACS Photonics 7 (2020) 1716–1722.
[38] M.R. Bonyadi, Z. Michalewicz, Particle swarm optimization for single objective continuous space problems: a review, Evol. Comput. 25 (2017) 1–54.
[39] J.R. Ong, H.S. Chu, V.H. Chen, A.Y. Zhu, P. Genevet, Freestanding dielectric nanohole array metasurface for mid-infrared wavelength applications, Opt. Lett. 42 (2017) 2639–2642.
[40] B. Zhang, et al., Particle swarm optimized polarization beam splitter using metasurface-assisted silicon nitride Y-junction for mid-infrared wavelengths, Opt. Commun. 451 (2019) 186–191.
[41] Y. Zhang, et al., A compact and low loss Y-junction for submicron silicon waveguide, Opt. Express 21 (2013) 1310–1316.
[42] M.J. Haji-Ahmadi, V. Nayyeri, M. Soleimani, O.M. Ramahi, Pixelated checkerboard metasurface for ultra-wideband radar cross section reduction, Sci. Rep. 7 (2017), 11437.
[43] A. Lalbakhsh, M.U. Afzal, K.P. Esselle, Multiobjective particle swarm optimization to design a time-delay equalizer metasurface for an electromagnetic band-gap resonator antenna, IEEE Antennas Wirel. Propag. Lett. 16 (2016) 912–915.
[44] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: a brief review, Comput. Intell. Neurosci. 2018 (2018) 7068349.
[45] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[46] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertainty Fuzziness Knowledge Based Syst. 6 (1998) 107–116.
[47] D.M. Hawkins, The problem of overfitting, J. Chem. Inf. Comput. Sci. 44 (2004) 1–12.
[48] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (2018) 55–75.
[49] X. Yi, E. Walia, P. Babyn, Generative adversarial network in medical imaging: a review, Med. Image Anal. 58 (2019) 101552.
[50] A.B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: a systematic review, IEEE Access 7 (2019) 19143–19165.

[51] C. Lee, et al., Development of artificial neural network system to recommend process conditions of injection molding for various geometries, Adv. Intell. Syst. 2 (2020) 2000037.
[52] Y. Rivenson, et al., Deep learning microscopy, Optica 4 (2017) 1437–1443.
[53] S. So, T. Badloe, J. Noh, J. Bravo-Abad, J. Rho, Deep learning enabled inverse design in nanophotonics, Nanophotonics 9 (2020) 1041–1057.
[54] W. Ma, et al., Deep learning for the design of photonic structures, Nat. Photonics 15 (2021) 77–90.
[55] Q. Zhang, H. Yu, M. Barbiero, B. Wang, M. Gu, Artificial neural networks enabled by nanophotonics, Light: Sci. Appl. 8 (2019) 42.
[56] J. Noh, et al., Design of a transmissive metasurface antenna using deep neural networks, Opt. Mater. Express 11 (2021) 2310–2317.
[57] S. So, D. Lee, T. Badloe, J. Rho, Inverse design of ultra-narrowband selective thermal emitters designed by artificial neural networks, Opt. Mater. Express 11 (2021) 1863–1873.
[58] J. Yun, S. Kim, S. So, M. Kim, J. Rho, Deep learning for topological photonics, Adv. Phys. X 7 (2022), 2046156.
[59] J. Peurifoy, et al., Nanophotonic particle simulation and inverse design using artificial neural networks, Sci. Adv. 4 (2018) eear4206.
[60] C.F. Bohren, D.R. Huffman, Absorption and Scattering of Light by Small Particles, John Wiley & Sons, 1998, https://doi.org/10.1002/9783527618156.
[61] I. Malkiel, et al., Plasmonic nanostructure design and characterization via deep learning, Light: Sci. Appl. 7 (2018) 60.
[62] X. Xu, et al., An improved tandem neural network for the inverse design of nanophotonics devices, Opt. Commun. 481 (2021) 126513.
[63] S. An, et al., A deep learning approach for objective-driven all-dielectric metasurface design, ACS Photonics 6 (2019) 3196–3207.
[64] J. Jiang, M. Chen, J.A. Fan, Deep neural networks for the evaluation and design of photonic devices, Nat. Rev. Mater. 6 (2020) 679–700.
[65] C. Zhang, J. Jin, W. Na, Q.-J. Zhang, M. Yu, Multivalued neural network inverse modeling and applications to microwave filters, IEEE Trans. Microwave Theory Tech. 66 (2018) 3781–3797.
[66] D. Liu, Y. Tan, E. Khoram, Z. Yu, Training deep neural networks for the inverse design of nanophotonic structures, ACS Photonics 5 (2018) 1365–1369.
[67] K. Kojima, et al., Deep neural networks for inverse design of nanophotonic devices, J. Light. Technol. 39 (2021) 1010–1019.
[68] I. Sajedian, T. Badloe, J. Rho, Optimisation of colour generation from dielectric nanostructures using reinforcement learning, Opt. Express 27 (2019) 5874–5883.
[69] O. Hemmatyar, S. Abdollahramezani, Y. Kiarashinejad, M. Zandehshahvar, A. Adibi, Full color generation with fano-type resonant HfO2 nanopillars designed by a deep-learning approach, Nanoscale 11 (2019) 21266–21274.
[70] L. Deng, X. Li, Machine learning paradigms for speech recognition: an overview, IEEE Trans. Audio Speech Lang. Process. 21 (2013) 1060–1089.
[71] S. So, Y. Yang, T. Lee, J. Rho, On-demand design of spectrally sensitive multiband absorbers using an artificial neural network, Photonics Res. 9 (2021) B153–B158.
[72] S. So, J. Mun, J. Rho, Simultaneous inverse design of materials and structures via deep learning: demonstration of dipole resonance engineering using core–shell nanoparticles, ACS Appl. Mater. Interfaces 11 (2019) 24264–24268.
[73] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2323.
[74] R. Sarma, et al., Optimization and prediction of spectral response of metasurfaces using artificial intelligence, Crystals 10 (2020) 1114.
[75] T. Asano, S. Noda, Optimization of photonic crystal nanocavities based on deep learning, Opt. Express 26 (2018) 32704–32717.
[76] R. Lin, Y. Zhai, C. Xiong, X. Li, Inverse design of plasmonic metasurfaces by convolutional neural network, Opt. Lett. 45 (2020) 1362–1365.

[77] A. Lininger, M. Hinczewski, G. Strangi, General inverse design of layered thin-film materials with convolutional neural networks, ACS Photonics 8 (12) (2021) 3641–3650.
[78] D.P. Kingma, M. Welling, An introduction to variational autoencoders, arXiv Prepr. arXiv1906.02691 (2019).
[79] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv Prepr. arXiv1511.05644 (2016).
[80] P. Turaga, R. Anirudh, R. Chellappa, Manifold learning, in: Computer Vision, Springer International Publishing, 2020, pp. 1–6, https://doi.org/10.1007/978-3-030-03243-2_824–1.
[81] Y. Long, J. Ren, H. Chen, Unsupervised manifold clustering of topological phononics, Phys. Rev. Lett. 124 (2020) 185501.
[82] Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing 184 (2016) 232–242.
[83] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 374 (2016) 20150202.
[84] Z.A. Kudyshev, A.V. Kildishev, V.M. Shalaev, A. Boltasseva, Machine-learning-assisted metasurface design for high-efficiency thermal emitter optimization, Appl. Phys. Rev. 7 (2020), 021407.
[85] Y. Kiarashinejad, S. Abdollahramezani, M. Zandehshahvar, O. Hemmatyar, A. Adibi, Deep learning reveals underlying physics of light–matter interactions in nanophotonic devices, Adv. Theory Simul. 2 (2019) 1900088.
[86] Y. Kiarashinejad, S. Abdollahramezani, A. Adibi, Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures, NPJ Comput. Mater. 6 (2020) 1–12.
[87] D. Melati, et al., Mapping the global design space of nanophotonic components using machine learning pattern recognition, Nat. Commun. 10 (2019) 4775.
[88] Z. Liu, Z. Zhu, W. Cai, Topological encoding method for data-driven photonics inverse design, Opt. Express 28 (2020) 4825–4835.
[89] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv Prepr. arXiv1312.6114 (2003).
[90] W. Ma, F. Cheng, Y. Xu, Q. Wen, Y. Liu, Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy, Adv. Mater. 31 (2019) 1901111.
[91] X. Hou, L. Shen, K. Sun, G. Qiu, Deep feature consistent variational autoencoder, in: Proceedings—2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Institute of Electrical and Electronics Engineers Inc, 2017, pp. 1133–1141, https://doi.org/10.1109/WACV.2017.131.
[92] I.J. Goodfellow, et al., Generative adversarial networks, arXiv Prepr. arXiv1406.2661 (2014).
[93] E.V. Mazumdar, M.I. Jordan, S.S. Sastry, On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games, arXiv Prepr. arXiv1901.00838 (2019).
[94] T. Salimans, et al., Improved techniques for training GANs, arXiv Prepr. arXiv1606.03498 (2016).
[95] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 1125–1134, https://doi.org/10.1109/CVPR.2017.632.
[96] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, 4, International Machine Learning Society (IMLS), 2016, pp. 1558–1566.
[97] Z. Liu, D. Zhu, S.P. Rodrigues, K.-T. Lee, W. Cai, Generative model for the inverse design of metasurfaces, Nano Lett. 18 (2018) 6570–6576.
[98] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: 4th Int. Conf. Learn. Represent. ICLR 2016—Conf. Track Proc, 2015, pp. 1–16.
[99] S. So, J. Rho, Designing nanophotonic structures using conditional deep convolutional generative adversarial networks, Nanophotonics 8 (2019) 1255–1261.
[100] Z.A. Kudyshev, A.V. Kildishev, V.M. Shalaev, A. Boltasseva, Machine-learning-assisted topology optimization for highly efficient thermal emitter design, in: Conference on Lasers and Electro-Optics *FTh3C.2*, OSA, 2019, https://doi.org/10.1364/CLEO_QELS.2019.FTh3C.2.

[101] Z.A. Kudyshev, A.V. Kildishev, V.M. Shalaev, A. Boltasseva, Machine learning-assisted global optimization of photonic devices, Nanophotonics 10 (2020) 371–383.
[102] J. Jiang, et al., Free-form diffractive metagrating design based on generative adversarial networks, ACS Nano 13 (2019) 8872–8878.
[103] F. Wen, J. Jiang, J.A. Fan, Robust freeform metasurface design based on progressively growing generative networks, ACS Photonics 7 (2020) 2098–2104.
[104] J. Jiang, J.A. Fan, Global optimization of dielectric metasurfaces using a physics-driven neural network, Nano Lett. 19 (2019) 5366–5372.
[105] M. Chen, J. Jiang, J.A. Fan, Reparameterization to enforce constraints in the inverse design of metasurfaces, OSA, 2020, pp. 1–2.
[106] J. Jiang, J.A. Fan, Multiobjective and categorical global optimization of photonic structures based on ResNet generative neural networks, Nanophotonics 10 (2020) 361–369.
[107] O. Andrzej, K. Stanislaw, Evolutionary Algorithms for Global Optimization. Global Optimization, Springer US, 2006, https://doi.org/10.1007/0–387-30927-6_12.
[108] The Motivation for Differential Evolution. Differential Evolution, Springer-Verlag, 2006, https://doi.org/10.1007/3–540-31306-0_1.
[109] C.F.M. Toledo, L. Oliveira, P.M. França, Global optimization using a genetic algorithm with hierarchically structured population, J. Comput. Appl. Math. 261 (2014) 341–351.
[110] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2010) 4–31.
[111] Y. Kiarashinejad, et al., Knowledge discovery in nanophotonics using geometric deep learning, Adv. Intell. Syst. 2 (2020) 1900132.
[112] X. Shi, T. Qiu, J. Wang, X. Zhao, S. Qu, Metasurface inverse design using machine learning approaches, J. Phys. D. Appl. Phys. 53 (2020) 275105.
[113] Z. Liu, et al., Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques, Adv. Mater. 32 (2020) 1904790.

CHAPTER 2

Machine learning for solid mechanics

Charles Yang, Zhizhou Zhang, and Grace X. Gu
Mechanical Engineering, University of California, Berkeley, Berkeley, CA, United States

1. Introduction

Solid mechanics, as a branch of continuum mechanics, is the study of the deformation or motion of solids under external applied loadings including forces, displacements, temperature changes, or other agents. Solid mechanics allows engineers to mathematically describe the physical behavior of solid materials and builds the foundation for material design and diagnosis [1]. There are two basic components needed to model the mechanics of a solid material: First, the fundamental physical laws that all continuums (including fluids) must obey, for example, the balance of linear momentum. Second, a constitutive equation that describes the materials' response to the external loadings, for example, a stress–strain relationship. In most cases, the constitutive relationship is either measured from experiments or derived from first principles (sometimes a combination of the two), which is then combined with the physical laws to form a set of complete governing equations. Countless governing equations have been developed in the past hundreds of years to tackle different application scenarios. Many classical theories are based upon many assumptions and simplifications so that the results can be calculated easily, for example, composite beam theory, Hertz contact theory, and Euler's critical load formula [2,3].

However, these classical theories are only applicable in very limited situations, making them insufficient for more complicated applications. Instead, more generalized governing equations for complex 3D geometries under various boundary conditions need to be solved. To approach complex problems in modern industry, various numerical methods have been developed among which the finite element method (FEM) is the most widely used for solid mechanics problems [4–8]. The finite element method basically discretizes the space into small elements where the solution of each element is approximated by a base function and a finite number of degrees of freedom. The approximation is realized by minimizing the associate error from the variational form of the governing equations. However, numerical methods including FEM have two major drawbacks: First, the computational cost scales drastically with the number of elements/nodes especially in 3D geometries. Second, complex numerical systems may lack available gradient or sensitivity information which are extremely important for material or structural design problems.

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00002-0
Copyright © 2023 Elsevier Inc.
All rights reserved.

Recently, artificial intelligence (AI), or more specifically machine learning (ML) methods, which are empirical, data-driven models, have risen to prominence as a promising method of simulating and solving physical engineering problems. This meteoric rise of ML in scientific domains is inspired by the success such methods have shown in a variety of other applications, such as image recognition, the challenging board game Go, and natural language understanding [9–14]. A specific family of machine learning models called neural networks is responsible for much of the recent success in applications that were previously believed to be decades away for algorithms to solve.

Already, ML is being applied and implemented in a variety of scientific domains, across a variety of disciplines. Such methods are proposed to accelerate density functional theory (DFT) calculations of molecular properties, model using molecular dynamics (MD), predict the state of health of batteries, and understand protein folding [14–18]. ML methods are also being applied to sift through the incredibly large amounts of data generated by astronomical observatories to find new planets [19]. Physicists are even using deep learning to search for new exotic particles in the debris left behind by powerful particle colliders [20]. While extensive research and studies have been conducted concerning the application of ML to different scientific disciplines that consider natural phenomena at a variety of length scales, this contribution will provide an overview of the recent research progress of ML on solid structural and materials (such as advanced composites) engineering. Computationally, solid material and structural design problems have mostly been approached through FEM as discussed above. FEM is of particular interest due to its widescale industrial application in a variety of engineering fields that are of significant industrial importance such as the design analysis and assessment of wind turbine blades, airplane wings, and automotive truck chassis [21–23]. As with other important fields of science like chemistry and physics, ML holds the potential to fundamentally reshape the toolkit scientists and engineers use to solve complex design problems and change the way we approach numerical simulations.

In this chapter, several segmented case studies of different ML techniques in a variety of applications are covered, including binary composites, auxetic metamaterials, and graphene configurations. Finally, we will discuss the future opportunities of ML applied to solid mechanics, as well as important considerations as the field matures.

2. Case studies

In order to conceptually organize AI research for solid mechanics, we group work into two types of modeling approaches: forward and inverse designs. Forward design maps the material structure to properties; this functional mapping matches the direction that FEM models use; hence they are referred to as modeling the "forward" direction. Data-driven models applied in this direction are also referred to as surrogate models, as they act as surrogates for existing numerical models. Inverse design is more closely related to what

engineers do, which is to map a set of desired properties to the final material structure or design. This approach is much more challenging because such mappings are usually not 1-to-1 and due to the lack of prior knowledge for the property to structure relationships. In addition to providing examples from both of these modeling paradigms in this section, an overview of two advanced ML model classes is provided: physics-informed neural networks and graph neural networks.

2.1 Surrogate modeling for materials

Finite element method has often been used to solve real-life solid mechanics problems. However, FEM is often computationally intensive and unscalable for problems with large design spaces. One solution to this design problem is to use a surrogate model, one that can stand-in for FEM with relatively similar accuracy but is much faster. This solution framework is very common due to the relatively simple model-building process. The standard procedure for surrogate modeling is outlined in Fig. 1: data preprocessing,

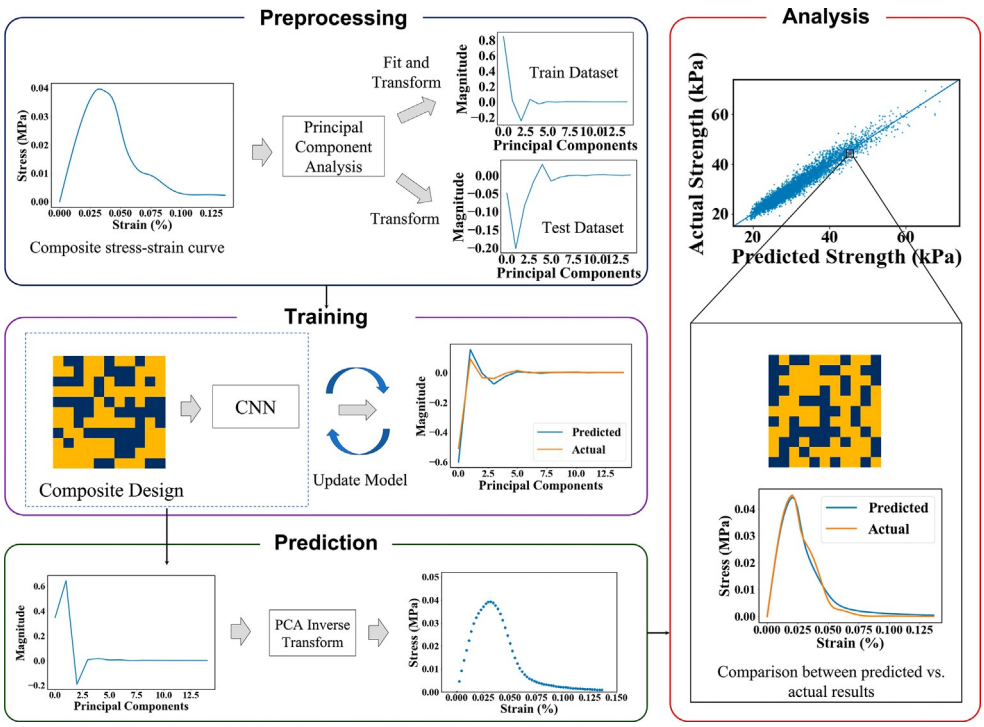


Fig. 1 In a standard surrogate modeling pipeline, the data are first preprocessed. Then a model is trained on the data; the trained model is used to make predictions, which are then analyzed to determine the model's performance. *Figure adapted from Yang, C., Kim, Y., Ryu, S., Gu, G.X., Prediction of composite microstructure stress-strain curves using convolutional neural networks. Mater. Des. 189 (2020) 108509.*

model training, model prediction, and error analysis. In this section, we provide a retrospective of how to approach problem setup and exploration when using ML models. Several applications of ML methods to different design scenarios are also covered.

2.1.1 Case study: Prediction of material properties of binary composites

This section provides a case study of how to progressively tackle a fundamental problem in solid mechanics with machine learning. ML is a new tool in the engineering toolkit—we hope this walkthrough of how to set up problems will provide insight into how to successfully frame problems in a way that allows us to leverage AI capabilities.

Predicting material properties is a difficult problem that is becoming increasingly important as material design spaces are exploding in complexity. For instance, composite materials are one class of materials that have combinatorically large design spaces, but, with the right design, can possess desirable mechanical properties, such as low density and high mechanical toughness. As a result, they have already demonstrated significant promise in helping reduce carbon emissions in a variety of automotive and aerospace applications. Developing an ML surrogate model might help accelerate our ability to search the vast design space efficiently.

In our problem setup, we seek to identify binary composite designs with a desirable combination of mechanical properties. We begin with a simple binary composite plate as the design space, since it is a both a general problem setup with a large design space, but simple enough to serve as a versatile testbed for the possibilities, ML offers to improve our modeling capabilities. Rather than directly trying to predict property values, research has shown that convolutional neural networks (CNN)can be used to rank the relative ranking between different composite designs [24]. A surrogate model, a convolutional neural network in this case, is trained to predict the properties of composites drawn from a large design space and demonstrates superior design discovery. The CNN is also compared to both a baseline of a linear model and the FEM model itself; feature importance from the CNN is also used to interpret the result. Having demonstrated the ability to relatively rank composite material properties, CNN can also be used to predict the actual mechanical properties of binary composites [25]. Again, the CNN outperforms a linear model, as well as a random forest in most cases, and demonstrates significant speed up as a surrogate model. In addition, more advanced gradient-based feature visualization is used to better discern what the CNN is learning and to develop more advanced architectures that can take advantage of modulus values when calculating strength and toughness, since modulus is easier to calculate with FEM. Finally, having demonstrated the ability of a CNN to predict modulus, strength, and toughness, the entire stress–strain curve of binary composites is predicted with a CNN. A linear decomposition of the stress–strain curve is also used to improve model training and is visualized for interpretability [26].

Here, we would like to highlight several important rules of thumb to consider when applying ML to material's problems. The most important lesson is to start simple: ML is a fairly advanced and novel framework that is not guaranteed to succeed on all problems—identifying the base case will help build confidence and experience, as well as provide a sanity check. It is also helpful to frame problems in such a way that already existing ML models can be easily applied and transferred. In the case of binary composites, their design parameterizations, i.e., binary matrices, are very similar to images, and as a result, we can easily apply standard computer vision models to the problem. However, adaptation may also be required: for instance, a "two-headed" neural network was developed because of our domain knowledge about the relative ease with which modulus values can be obtained and their correlation to other mechanical properties. It is also important to set up baselines with the original, or "oracle" model, which is usually FEM, as well as with simpler models, to ensure that unnecessary complexity is not being built in. And finally, interpretability is an important component of building surrogate models, particularly when there exists physical understanding of and intuition for the system being modeled. While not all models are equally interpretable, being able to understand what the model is learning is useful for incorporating prior knowledge and to ensure the model is not picking up spurious correlations.

2.1.2 Examples of surrogate models

As surrogate modeling, which essentially attempts to serve as an application-specific drop-in for FEM, is fairly straightforward, this problem-solving framework has seen many different applications, particularly for material's design with unique properties. For instance, auxetic metamaterials are structures that have negative Poisson's ratio that expand under tension. In Wilt et al., the authors develop a neural network to identify honeycomb lattices that display auxetic properties [27]. Zhang et al. used convolutional neural networks to predict deformation of smart composite materials [28]. Kim et al. also used CNN's to design adhesive pillar interfaces and use a genetic algorithm to iteratively search for the optimal design, enabled by the much faster deep learning surrogate model [29]. Machine learning can also be used to accelerate the design and understanding of more traditional applications as well. For instance, Nie et al. used convolutional neural networks that are also used to predict stress fields in cantilevered structures [30]. Many of these surrogate models define a set of design parameters and use deep learning methods to learn the underlying mechanics by training on a large corpus of simulated training data. However, oftentimes, scientists and engineers are striving to optimize for a set of material parameters. In these circumstances, surrogate models can often be paired with an optimizer, either gradient-based or evolutionary-based.

2.2 Inverse material design

Material structure design has long been a popular topic among mechanical/material engineers in both industry and research fields. Typically, evaluating the performance of a material design is considered a forward problem, and searching for a material design that best matches a user-defined objective is considered an inverse problem. And a good forward problem solver is always required for any attempt at the inverse problem. As discussed in the introduction, numerical methods have been widely used as a forward problem solver to simulate/predict material properties. In this section, we expand on how ML techniques are built upon numerical simulations to design and optimize material properties.

Classical material design requires iteratively solving and updating some design candidates (Fig. 2). In most cases, numerical simulations serve as the forward solver to evaluate the current candidate performance. And the updating methods are generally categorized into two big families: gradient-based and gradient-free. For instance, Hamel et al. used the genetic algorithm (a well-known gradient-free method) to optimize a smart composite beam for a target deformation [31]. Such methods are easy to implement, but often face challenges under large input space dimensions. On the other hand, gradient-based methods tend to be more robust and efficient in general applications. Gaynor et al. optimized a multi-material compliant inverter using the Solid Isotropic Material with Penalization (SIMP) method [32]. The SIMP method basically interpolates the discrete properties among candidate base materials through a smooth function, which offers gradient information to guide the change of material distribution. However, depending on the design features and objective, gradient information is not always available from the numerical solutions of a set of PDEs. Therefore, researchers applied ML techniques to generate gradients to help optimize material designs.

As introduced in the previous section, researchers have been training surrogate ML models to replace the role of numerical simulations to forward predict material

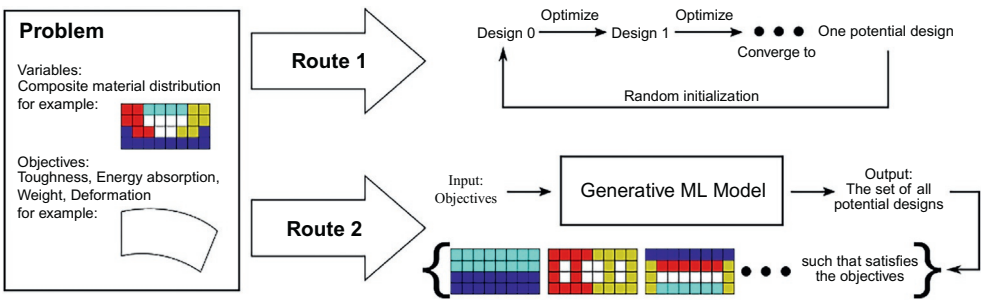


Fig. 2 A schematic showing two different material design approaches. Route 1 represents the classical iterative optimization, while route 2 illustrates a generative ML model that directly produces design candidates as its output.

properties. Any gradient-free design optimization techniques implemented on numerical simulators would also be applicable on ML surrogate models. And it turns out that many of these ML models (e.g., linear regression, neural networks) are differentiable, meaning that the model not only predicts a single design point, but also provides information about the local neighborhood in the design feature space. This feature allows the application of gradient-based optimization techniques, for example, gradient descent, newton's method, etc. Researchers have developed a material design framework using gradient-based optimization on an actively trained neural network [33]. The similar approach is extended to other numerical scenarios [34]. In most cases, neural networks are preferable than linear models in model-based optimization tasks for their ability to fit the details of a complex response function. One thing to note is that when optimization is conducted on ML surrogate models rather than a numerical simulator, each iteration can be significantly accelerated with its performance becomes relatively inferior, since the material property prediction is an approximation instead of being exact. As a tradeoff, major computational cost would instead lie in the model training phase where ground truth data are generated from simulations.

Researchers have also investigated a second route: using generative ML models to produce design candidates in one shot (Fig. 2). Different from surrogate models, generative ML models directly solve the inverse problem by taking a target material property as the input and outputting the candidate designs. However, a specific set of design objectives may correspond to multiple potential design solutions. Therefore, such generative models typically map a standard uniform distribution to the distribution of potential designs (more details can be found in Ref. [35]). This technique has been successfully used for generating artificial images that are almost indistinguishable from real ones [36]. Learning from these experiences in computer vision research, Mao et al. added a scoring system to the generative adversarial network which can then generate material designs with better expected properties [37]. Kyle et al. adopted a similar approach to generate porous graphene configurations based on desired energy levels [38]. Despite these successful applications, generative ML models are still in their infancy in terms of material designing, with plenty of room for improving the accuracy and training efficiency.

2.3 Physics-informed neural network

In previous sections, we have discussed various ML techniques to understand and design composite materials based on ground truth data generated from numerical simulations or experiments. Despite the essentiality of ground truth data for ML models, it is sometimes too expensive or even impossible to obtain data in real-life problems. To overcome this data shortage issue, researchers have been trying to explore different forms of prior knowledge that neural networks can learn from. And it turns out that in many engineering problems, there are perfect candidates serving as the prior knowledge: physics laws.

In this section, we will introduce the fundamentals of physics-informed neural networks and how they are utilized to solve material-related problems.

Most neural networks fed with ground truth data use a quadratic loss function so that the optimizer minimizes the distance between the predicted outputs and the true labels. Now suppose we want to train a neural network to represent the solution to the following equation: $u_t + uu_x = 0$, where u is a function of t and x (assuming an analytical solution cannot be found). A straightforward approach is to first sample a certain amount of data tuples (t, x, u) by numerically solving the equation. Then train the neural network with these ground truth data points as shown in Fig. 3. However, Raissi et al. introduced the physics-informed neural network which directly penalizes the residual of the governing equation [39]. In the case of our sample problem, the expression of the loss function would be $(\widehat{u}_t + \widehat{u}\widehat{u}_x)^2$, where $\widehat{u}$ represents the prediction from the neural network. By minimizing this loss function, the neural network is forced to make predictions that obey the governing equation. This method also has an elegant implementation since any partial derivatives in the governing equation can be easily computed through backpropagating the neural network.

Physics-informed neural networks have been very successful in fast visualization of flow profiles where direct measurements or numerical calculations may not be possible [40]. Unfortunately, this approach is not generally applicable since many problems are not governed by clear equations. However, many of the material learning and design problems do involve well-defined mechanics that can serve as prior physics knowledge. For example, Zhang et al. trained a neural network to predict the deformation field of a digital composite sheet [41]. Instead of generating ground truth labels from finite element simulations, the loss function was set to be the elastic energy over the entire material

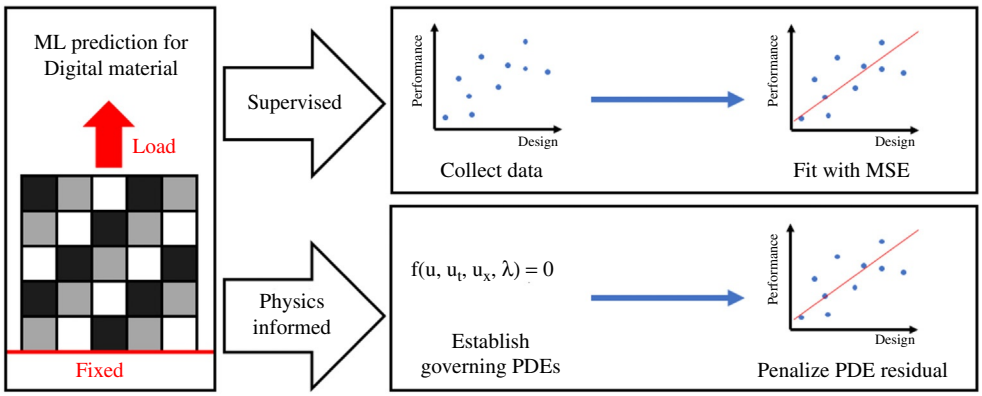


Fig. 3 A schematic comparing the workflow of a typical supervised learning approach which learns from ground truth data, and the physics-informed approach which learns from governing equations. *Figure adapted from Z. Zhang, G. Gu, Physics-informed deep learning for digital materials. Theor. Appl. Mech. Lett. 11 (2021) 100220.*

domain: $\min \frac{1}{2}\int(\varepsilon^T)[E](\varepsilon)\mathrm{d}\Omega$ where E stands for the material elasticity tensor and ε denotes the strain, which is known to be minimized under the static equilibrium state. Chen et al. applied the physics-informed neural network to elastography where material elastic properties are predicted from a measured strain distribution. This neural network is directly supervised by the linear elastic material constitutive law [42]. In most of the literature, the authors report comparable accuracy between supervised neural networks and physics-informed neural networks.

Despite the growing popularity of the physics-informed neural network, there has been limited attention in estimating the total computational cost. It is clear that making a prediction using a neural network is much faster than solving differential equations. This fast response also manifests when the model serves to produce gradient information. Nevertheless, the training cost of neural networks should never be ignored. Unlike typical supervised models whose training cost is dominated by the data collection process (e.g., simulation time of all data points), physics-informed neural networks have their cost concentrated at the model parameter optimization process since each backpropagation step starts from a physics loss that is much more complex than a quadratic loss. Zhang et al. briefly discussed the scaling relationship between the system size and a single-training epoch computation cost [41]. However, it remains an open research question to exactly bound the cost of physics loss due to the selection of hyperparameters and stochasticity of neural network training.

2.4 Graph neural networks

Identifying the right model architecture for different data formats is one large challenge in modeling, particularly for unstructured data types. Recently, graph neural networks are a novel type of neural network architecture that can be applied to graph-like inputs. The most common graph neural network is based on message passing between nodes: each node takes in a message, i.e., a vector or matrix, from all adjacent nodes and computes an update to the latent feature vector associated with each node using a composition of a simple mathematical function. Multiple rounds of message passing are done; at the end, a readout function computes the label of the graph, given the feature vector of all the nodes. To see examples of different mathematical formulations of the message passing and readout steps, see Ref. [43]. The message passing framework for graph neural networks is visualized in Fig. 4.

Graph neural networks have already seen great success in scientific fields that deal with complex structures, such as modeling quantum properties of molecules [43], discovering new antibiotics "A Deep Learning Approach to Antibiotic Discovery" [44], and understanding high-energy particle interactions in detectors [45]. In this section, several examples of how this advanced neural network architecture can be used for different modeling situations in a generalized fashion are provided.

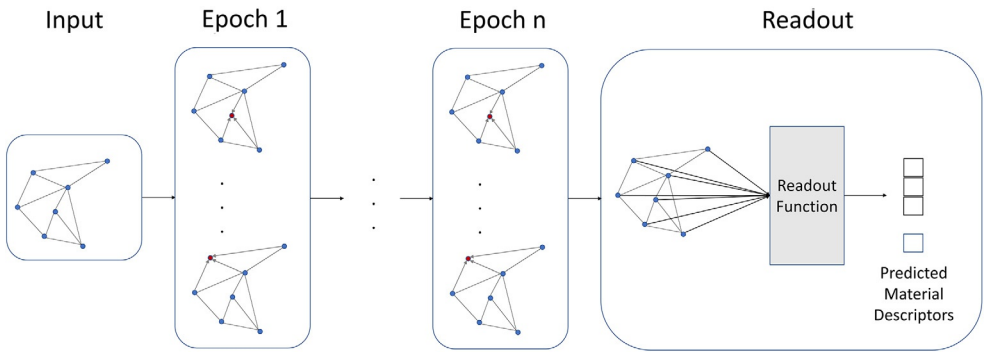


Fig. 4 A figure indicating the training process for message-passing graph neural networks. Each node in the graph receives a latent message vector from neighboring nodes and updates its own internal feature vector by applying non-linear functions. After *n* epochs of message passing for each node, a readout function specified by the user calculates the target material descriptor.

In Seff et al., they release a large-scale dataset of sketched CAD designs and demonstrate how graph neural networks can be used to aid the design process at different stages, ranging from generative modeling, image to CAD conversion, and autoconstraining [46]. Researchers at Autodesk also use graph neural networks for structural engineering of buildings and demonstrate that GNN's learn designs that use less material are more cost-effective, and safer. When benchmarking their graph neural network against standard numerical optimizers and human designers, they demonstrate that their model outperforms both, requiring significantly less compute or less time respectively, to achieve superior results [47]. Pfaff et al. at Google's DeepMind show that graph neural networks can accurately learn a variety of mesh-based simulation tasks, ranging from aerodynamics, structural mechanics, and cloth behavior [48].

Solid mechanics overall deal with a variety of numerical computations at different levels of abstraction, ranging from the differential equations at the core of FEM to the parameters that define a design. Graph neural networks, because of their ability to take in less formally structured inputs, are a promising model type for being able to generalize to a variety of FEM tasks, without being reliant on a specific set of design parameters.

3. Future opportunities and considerations

Solid mechanics is a well-poised field for AI adaption. It relies on robust, standardized numerical software that is already broadly adopted and trusted in industry. However, identifying optimal designs from these numerical models is challenging, often relying on human intuition and expertise, which is not always ideal or scalable. In particular, searching vast design spaces with numerically intensive FEM is a difficult task, yet a common theme. ML models, with their ability to learn from large-scale data efficiently, are a

promising tool to help solve this scalability problem. However, much of the progress in building AI and ML models have been driven by the concerns of software-defined technology companies in social media and ad recommendations. While AI can help solve important design problems in solid mechanics, we will need AI tools that are built for the job. There are many ways to use AI techniques in novel ways specific to problems in solid mechanics, many of which incorporate some element of domain or prior knowledge. There are also many important constraints in solid mechanics and FEM which are currently difficult to encode in ML models, such as manufacturing constraints and multiscale modeling. In order to build ML models capable of solving these domain-specific problems, we need "bilingual" people—those with the ability to speak the language of engineering and AI. Our hope is that this chapter will both provide a survey of the progress made and an incentive for others to learn how to use AI in their methodology and research.

In our experience, there are several important considerations when thinking about how to encourage AI adoption. In addition to the importance of "bilingual" scientists and engineers, we also need to consider the importance of compute and data, which are the sustenance of ML models. Indeed, much of the progress in AI is a result of open-sourced software (e.g., Pytorch, Tensorflow, Keras, and scikit-learn) and commonly agreed upon datasets (e.g., MNIST, CIFAR10, ImageNet) that acted as baselines for the research community. These open-sourced software and datasets are a great learning tool that can be applied not only for solid mechanics problems but also for broader engineering and science fields. AI has demonstrated itself to be a promising and potent new tool in the engineer's toolkit. But the work of adopting AI as a robust and accepted tool is only just beginning; indeed, there are still many open opportunities for researchers to help build a common set of software, data, and ML models tailored to the needs of the solid mechanics community.

4. Conclusions

In this chapter, an overview of AI applications to solid mechanics is provided, an important field of study with special relevance to our society's infrastructure. We survey its success in surrogate modeling and inverse design, as well as special advanced classes of ML models such as physics-informed neural networks and graph neural networks. We also consider the field as a whole and important considerations to think about when encouraging AI adoption. As outlined throughout this chapter, AI is a promising instrument for revolutionizing the way we design buildings, materials, and transportation. Such tools will be needed to confront a vastly exploding design space, driven by industrial advances in manufacturing. And indeed, the environmental and ecological challenges facing our society require the rapid, scalable development of designs that are sustainable and accessible. This will require lightweight cars and airplanes, new energy technologies, and more

environmentally friendly building designs. AI is a potent tool to help mechanical engineers and structural designers build a better future today.

References

[1] D.W. Rees, Mechanics of Solids and Structures, World Scientific Publishing Company, 2016.
[2] E.W. Jacobs, C. Yang, K.G. Demir, G.X. Gu, Vibrational detection of delamination in composites using a combined finite element analysis and machine learning approach, J. Appl. Phys. 128 (2020) 125104.
[3] D.L. Bartel, D.T. Davy, Orthopaedic Biomechanics: Mechanics and Design in Musculoskeletal Systems, Prentice Hall, 2006.
[4] T.I. Zohdi, Zohdi, Ditzinger, Finite Element Primer for Beginners, Springer, 2018.
[5] K. Demir, Z. Zhang, J. Yang, G. Gu, Computational and experimental design exploration of 3D-printed soft pneumatic actuators, Adv. Intell. Syst. 2 (2020), 2000013.
[6] X. Ma, S. Hajarolasvadi, G. Albertini, D.S. Kammer, A.E. Elbanna, A hybrid finite element-spectral boundary integral approach: applications to dynamic rupture modeling in unbounded domains, Int. J. Numer. Anal. Methods Geomech. 43 (2019) 317–338.
[7] S. Marimuthu, R. Eghlio, A. Pinkerton, L. Li, Coupled computational fluid dynamic and finite element multiphase modeling of laser weld bead geometry formation and joint strengths, J. Manuf. Sci. Eng. 135 (2013), 011004.
[8] G.X. Gu, M. Takaffoli, M.J. Buehler, Hierarchically enhanced impact resistance of bioinspired composites, Adv. Mater. 29 (2017) 1700060.
[9] He, K., Zhang, X., Ren, S. & Sun, J. n.d. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 770–778.
[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[11] V. Mnih, et al., Playing Atari With Deep Reinforcement Learning, arXiv, 2013. preprint arXiv:1312.5602.
[12] D. Silver, et al., Mastering the game of go without human knowledge, Nature 550 (2017) 354–359.
[13] Z. Jin, Z. Zhang, K. Demir, G.X. Gu, Machine learning for advanced additive manufacturing, Matter 3 (2020) 1541–1556.
[14] Z. Jin, Z. Zhang, J. Ott, G.X. Gu, Precise localization and semantic segmentation detection of printing conditions in fused filament fabrication technologies using machine learning, Addit. Manuf. 37 (2020), 101696.
[15] R. Nagai, R. Akashi, O. Sugino, Completing density functional theory by machine learning hidden messages from molecules, NPJ Comput. Mater. 6 (2020) 1–8.
[16] M.-F. Ng, J. Zhao, Q. Yan, G.J. Conduit, Z.W. Seh, Predicting the state of charge and health of batteries using data-driven machine learning, Nat. Mach. Intell. 2 (2020) 161–170.
[17] T. Jo, J. Hou, J. Eickholt, J. Cheng, Improving protein fold recognition by deep learning networks, Sci. Rep. 5 (2015) 1–11.
[18] C.-T. Chen, D.C. Chrzan, G.X. Gu, Nano-topology optimization for materials design with atom-by-atom control, Nat. Commun. 11 (2020) 1–9.
[19] C.J. Shallue, A. Vanderburg, Identifying exoplanets with deep learning: a five-planet resonant chain around kepler-80 and an eighth planet around kepler-90, Astron. J. 155 (2018) 94.
[20] P. Baldi, P. Sadowski, D. Whiteson, Searching for exotic particles in high-energy physics with deep learning, Nat. Commun. 5 (2014) 1–9.
[21] A. Albanesi, V. Fachinotti, I. Peralta, B. Storti, C. Gebhardt, Application of the inverse finite element method to design wind turbine blades, Compos. Struct. 161 (2017) 160–172.
[22] Das, S.K. & Roy, S. in n.d. IOP Conference Series: Materials Science and Engineering. 012077 IOP Publishing.
[23] Q. Mao, et al., Structural condition assessment of a bridge pier: a case study using experimental modal analysis and finite element model updating, Struct. Control. Health Monit. 26 (2019), e2273.

[24] G.X. Gu, C.-T. Chen, M.J. Buehler, De novo composite design based on machine learning algorithm, Extreme Mech. Lett. 18 (2018) 19–28.
[25] C. Yang, Y. Kim, S. Ryu, G.X. Gu, Using convolutional neural networks to predict composite properties beyond the elastic limit, MRS Commun. 9 (2019) 1–9.
[26] C. Yang, Y. Kim, S. Ryu, G.X. Gu, Prediction of composite microstructure stress-strain curves using convolutional neural networks, Mater. Des. 189 (2020) 108509.
[27] J.K. Wilt, C.X. Yang, G.X. Gu, Accelerating auxetic metamaterial design with deep learning, Adv. Eng. Mater. 22 (2020), 1901266.
[28] Z. Zhang, G.X. Gu, Finite element based deep learning model for deformation behavior of digital materials, Adv. Theor. Simul. 3 (2020), 2000031.
[29] Y. Kim, C. Yang, Y. Kim, G.X. Gu, S. Ryu, Designing an adhesive pillar shape with deep learning-based optimization, ACS Appl. Mater. Interfaces 12 (2020) 24458–24465.
[30] Z. Nie, H. Jiang, L.B. Kara, Stress field prediction in cantilevered structures using convolutional neural networks, J. Comput. Inf. Sci. Eng. 20 (2020), 011002.
[31] C.M. Hamel, et al., Machine-learning based design of active composite structures for 4D printing, Smart Mater. Struct. 28 (2019), 065005.
[32] A.T. Gaynor, N.A. Meisel, C.B. Williams, J.K. Guest, Multiple-material topology optimization of compliant mechanisms created via PolyJet three-dimensional printing, J. Manuf. Sci. Eng. 136 (2014), 061015.
[33] C.T. Chen, G.X. Gu, Generative deep neural networks for inverse materials design using backpropagation and active learning, Adv. Sci. 7 (2020), 1902607.
[34] Xue, T., Beatson, A., Adriaenssens, S. & Adams, R. n.d. in International Conference on Machine Learning. 10638–10647 PMLR.
[35] I.J. Goodfellow, et al., Generative Adversarial Networks, arXiv, 2014. preprint arXiv:1406.2661.
[36] A. Creswell, et al., Generative adversarial networks: an overview, IEEE Signal Process. Mag. 35 (2018) 53–65.
[37] Y. Mao, Q. He, X. Zhao, Designing complex architectured materials with generative adversarial networks, Sci. Adv. 6 (2020), eaaz4169.
[38] K. Mills, C. Casert, I. Tamblyn, Adversarial generation of mesoscale surfaces from small-scale chemical motifs, J. Phys. Chem. C 124 (2020) 23158–23163.
[39] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics Informed Deep Learning (Part I): Data-Driven Solutions of Nonlinear Partial Differential Equations, arXiv, 2017. preprint arXiv:1711.10561.
[40] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, Science 367 (2020) 1026–1030.
[41] Z. Zhang, G. Gu, Physics-informed deep learning for digital materials, Theor. Appl. Mech. Lett. 11 (2021), 100220.
[42] C.-T. Chen, G.X. Gu, Learning hidden elasticity with deep neural networks, Proc. Natl. Acad. Sci. 118 (2021), e2102721118.
[43] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O. & Dahl, G.E. n.d. in International Conference on Machine Learning. 1263–1272 PMLR.
[44] J.M. Stokes, et al., A deep learning approach to antibiotic discovery, Cell 180 (2020) 688–702.e613.
[45] I. Henrion, et al., Neural message passing for jet physics, in: Workshop on Deep Learning for Physical Sciences (DLPS 2017), NIPS 2017, Long Beach, CA, USA, 2017.
[46] A. Seff, Y. Ovadia, W. Zhou, R.P. Adams, Sketchgraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design, arXiv, 2020. preprint arXiv:2007.08506.
[47] Chang, K.-H. & Cheng, C.-Y. n.d. in International Conference on Machine Learning. 1426–1436 PMLR.
[48] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P.W. Battaglia, Learning Mesh-Based Simulation With Graph Networks, arXiv, 2020. preprint arXiv:2010.03409.

CHAPTER 3

Neural networks in phononics

Liu Chen-Xu and Yu Gui-Lan
Beijing Jiaotong University, School of Civil Engineering, Beijing, China

1. Introduction

Phononic crystals (PnCs) are periodic elastic composites and have broad application prospects in controlling and manipulating elastic waves due to the existence of phononic bandgaps. When elastic waves propagate in PnCs, a special dispersion relation or band diagram is generated, where the frequency domains covered by dispersion curves are called passbands and those uncovered are stopbands, namely bandgaps. Theoretically, elastic waves within bandgaps will be suppressed when propagating in PnCs, and those within passbands will spread without energy loss. If there are point defects or linear defects in PnCs, the elastic waves within the bandgaps will be localized at or propagated along the defects. Hence, PnCs can be used to block or control elastic waves.

The initial work concerning with PnCs can be traced back to 1971 when Narayanamurti et al. discovered for the first time that a layered periodic structure can be used to control the propagation of high-frequency waves [1], which was not known as PnCs at that time. In the early 1990s, Sigalas and Economou verified that there are bandgaps for elastic waves in a three-dimensional (3D) periodic structure [2] and two-dimensional (2D) fluid–solid system [3]. The concept of PnCs was first put forward in 1993 by Kushwaha et al. in the study of 2D solid periodic composites [4]. In 1995, Martinez Salar et al. conducted an acoustic experiment on a statue standing in a park in Madrid, Spain, which is a periodic structure called "Flowing Melody," and found that waves at some frequency ranges will experience attenuation when passing through the statue [5]. Vasseur et al. proved the existence of bandgaps in 2D solid–solid PnCs by experiment and the theoretical analysis was carried out [6]. Although wave propagation can be effectively controlled by tuning bandgaps of PnCs, however, the size of the structure must be very large to obtain bandgaps in low-frequency region owing to the Bragg scattering mechanism. In 2000, Liu et al. proposed a new PnCs based on local resonance [7], and the generation of its bandgaps is due to the resonance of scatterers, which is different from that induced by Bragg scattering. The wavelength corresponding to the first bandgap frequency is far greater than the size of the lattice, which marks another major breakthrough in the study of PnCs. The locally resonant PnC is also called metamaterial or metastructure because of its superpower of small size to control large wavelength.

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00003-2
Copyright © 2023 Elsevier Inc.
All rights reserved.

The traditional methods for calculating PnCs include the transfer matrix method, plane wave expansion method, finite element method, etc., and they will consume a lot of time and computing resources. It is well known that structures and their characteristics obey certain physical laws, which can be discovered and exploited. If there exists an approach that can predict the characteristics of PnCs quickly, the discovery and design of PnCs will be convenient. The key point is to find a method to simulate the physical laws of PnCs and output rapidly their characteristics. With the development of deep learning, artificial neural network (ANN) has entered the vision of many researchers. It can simulate an arbitrarily complex relationship through the combination of many simple functions, which is a powerful simulation algorithm. ANN provides new possibilities for PnCs exploration and design.

ANN is a method for representation learning and can approximate an arbitrary mapping relationship. Its appearance can be traced back to 1943, when McCulloch and Pitts [8] first proposed an ANN calculation model based on mathematics and threshold logic algorithms. In the late 1940s, Hebb created a typical unsupervised learning rule based on neuroplasticity, that is Hebb learning. In 1954, Farley and Clark [9] used a computer to simulate the Hebb network for the first time. In 1958, Rosenblatt created a modal recognition algorithm, which is a two-layer computer learning network realized by simple addition and subtraction. In 1974, Werbos [10] proposed a back-propagation algorithm, which is a key development in ANN and effectively solves the XOR problem. In 1982, Hopfield [11] proposed a recurrent neural network, which is more conducive to processing serialized data. In 1989, LeCun et al. [12] applied the back-propagation algorithm to a convolutional neural network (CNN). In 2006, Hinton et al. [13] used greedy layer-by-layer pre-training to effectively train a deep belief network. In 2012, the team led by Hinton won the championship with the excellent image classification effect of AlexNet in the ImageNet competition, and its recognition accuracy far exceeded the second place by 10 percentage points, which caused an upsurge in ANN. In recent years, excellent neural network models have emerged one after another, such as GoogleNet [14], ResNet [15], DenseNet [16], variational autoencoder [17], generative adversarial network [18], etc.

In 2019, Finol et al. used a CNN and a multilayer perceptron (MLP) to predict eigenvalues of 1D and 2D PnCs [19]. For 1D case, the CNN reaches 98% accuracy on the testing set when trained with 20,000 sets of data, whereas the MLP only has 85% accuracy, which was trained with 100,000 sets of data. In the CNN, they used a matrix to represent the distributions of materials in PnCs; for 1D PnCs, the input is a 2×50 matrix, where one column is elastic modulus with 50 elements and the other is the corresponding mass density of them; for 2D PnCs, the input is a 3D matrix with $128 \times 128 \times 5$ elements, where the five layers (each layer has 128×128 elements) represent elastic modulus, Poisson's ratio, mass density, and two components of wave vectors, respectively. In this research, they only studied the prediction of eigenvalues in the dispersion curves of PnCs.

The present authors compared the prediction accuracy of MLPs and radial basis function (RBF) neural networks for the dispersion curves of 1D PnCs [20]. Three cases are considered, that is, one-parameter prediction, two-parameter prediction, and three-parameter prediction. For one-parameter prediction, the input of neural networks is filling fraction, the training set has only 10 sets of data, and the two neural networks have great performance on the prediction of dispersion curves. For two-parameter prediction, shear modulus, and mass density are input into neural networks, and MLP has a better result. For three-parameter prediction, filling fraction, shear modulus, and mass density are taken as the input, and the prediction accuracy of MLP is better than that of the RBF neural network. We also used MLP and autoencoder (AE) with a pretrained decoder to design parameters of 1D PnCs [21]. The results illustrated that the AE has a better design performance compared with MLP. The difference between the two neural networks is that MLP is a supervised learning method, while AE is unsupervised. The MLP is to predict the designed 1D PnCs forwardly, and the AE is to search for a solution that satisfies the target by the AE's encoder according to the predicted values of AE's pretrained decoder.

Sadat and Wang discovered the properties of 2D PnCs with square lattice and cylindrical scatterer by machine learning approaches [22]. Three methods, linear regression, MLP, and random forest, are compared. The input is the diameter of scatterer, the elastic modulus and mass density of host and scatterer materials. The existence, center frequency, and width of the bandgap are predicted, and they found that the optimal algorithm among the three methods is random forest.

Li et al. proposed a deep learning model composed of a MLP and an AE to design the topological configurations of 2D PnCs [23]. The configurations of 2D PnCs are represented by 128×128 matrices, where there are two different numbers respectively indicating host and scatterer materials. A vector, which contains passbands and stopbands, has 1000 dimensions, where one value (0 or 1) determines whether a small frequency range is passband or not. The AE is used to compress the matrix representing the configuration of a PnC into a feature vector and restores it from the feature vector. The MLP can predict a feature vector according to a given target, and the feature vector is input into the AE's decoder to obtain the final designed topological configuration.

It can be seen that the application of neural networks to PnCs has made some progress. In order to understand this technology easily, the forward prediction and inverse design of 1D PnCs by neural networks will be introduced in this chapter. Firstly, the transfer matrix method for calculating the dispersion curves and energy transmission spectrums of 1D PnCs will be presented. Next, we will use neural networks to predict the dispersion curves and energy transmission spectrums of 1D PnCs. Finally, two types of neural networks will be compared for the inverse design of 1D PnCs.

2. One-dimensional phononic crystals and transfer matrix method

1D PnCs considered here are composed of two materials named as A and B, respectively. They are distributed periodically along the x direction as shown in Fig. 1. The lattice constant is equal to a, and the thicknesses of material A and material B are a_A and a_B, respectively. Here, we consider the oblique incidence of SH wave with an angle θ, and the materials are assumed linear and elastic. The governing equation can be expressed as

$$\rho(x,z)\frac{\partial^2 u(x,z,t)}{\partial t^2} = \mu(x,z)\left(\frac{\partial^2 u(x,z,t)}{\partial x^2} + \frac{\partial^2 u(x,z,t)}{\partial z^2}\right), \tag{1}$$

where u is displacement perpendicularly to the x–z plane; ρ is mass density, and μ is shear modulus; t is time.

When a harmonic wave is considered, the displacement in 1D PnCs can be written as

$$u(x,z,t) = U(x)\mathrm{e}^{\mathrm{i}k_z z - \mathrm{i}\omega t}, \tag{2}$$

where ω is angular frequency; k_z is wave number along the z direction, which is constant according to Snell's law. Substituting Eq. (2) into Eq. (1), the displacements and the stresses in material A and material B in the nth lattice are respectively written as

$$u_{nj}(x,z,t) = \mathrm{e}^{\mathrm{i}k_z z - \mathrm{i}\omega t}\left(C_{nj}\mathrm{e}^{\mathrm{i}\alpha_j x} + D_{nj}\mathrm{e}^{-\mathrm{i}\alpha_j x}\right), j = \mathrm{A},\mathrm{B} \tag{3}$$

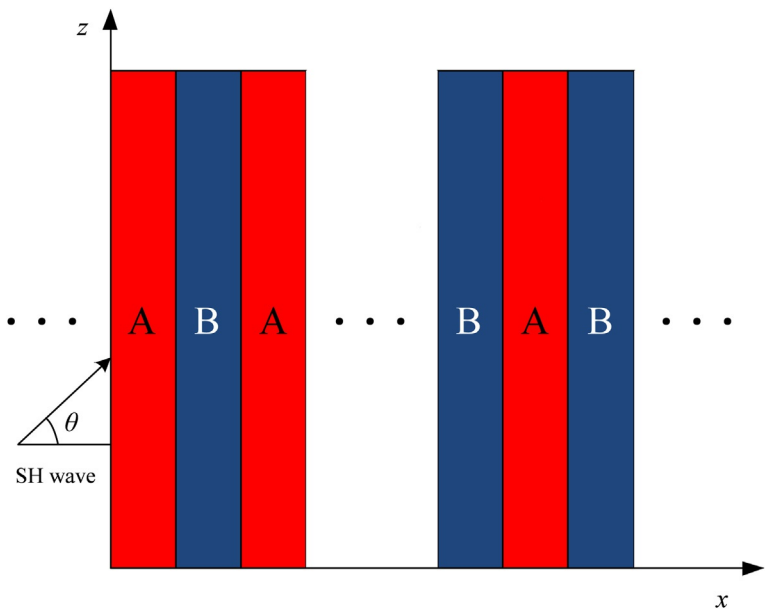


Fig. 1 One-dimensional phononic crystals where θ is incident angle.

$$\tau_{x\gamma}^{nj}(x,z,t) = \mu_j \alpha_j e^{ik_z z - i\omega t}\left(C_{nj}e^{i\alpha_j x} - D_{nj}e^{-i\alpha_j x}\right), j = \mathrm{A},\mathrm{B} \tag{4}$$

where, when j is equal to A, x belongs to $[(n-1)a, (n-1)a + a_\mathrm{A}]$, and when j is equal to B, x belongs to $[(n-1)a + a_\mathrm{A}, na]$; C_{nj} and D_{nj} indicate the amplitudes of transmitted and reflected waves, respectively; $\alpha_j = \sqrt{\left(\omega/c_j\right)^2 - (k_z)^2}$, where c_j is wave velocity. For simplicity, $\tau_{x\gamma}^{nj}$ is denoted by τ_{nj}.

The displacement and stress at the interface between material A and material B are continuous, thus we have

$$u_{n\mathrm{A}} = u_{(n-1)\mathrm{B}}, x = (n-1)a \tag{5}$$

$$\tau_{n\mathrm{A}} = \tau_{(n-1)\mathrm{B}}, x = (n-1)a \tag{6}$$

$$u_{n\mathrm{A}} = u_{n\mathrm{B}}, x = (n-1)a + a_\mathrm{A} \tag{7}$$

$$\tau_{n\mathrm{A}} = \tau_{n\mathrm{B}}, x = (n-1)a + a_\mathrm{A} \tag{8}$$

Therefore, the relationship between the amplitudes of incident and reflected waves in the nth lattice and the $(n-1)$th lattice can be obtained

$$\boldsymbol{\Psi}_{\boldsymbol{n}\mathbf{B}} = \boldsymbol{T}\boldsymbol{\Psi}_{(\boldsymbol{n-1})\mathbf{B}}, \tag{9}$$

where

$$\boldsymbol{\Psi}_{\boldsymbol{n}\mathbf{B}} = [C_{n\mathrm{B}}, D_{n\mathrm{B}}]^\mathrm{T}$$

$$\boldsymbol{T} = (\boldsymbol{K_2})^{-1}\boldsymbol{H_2}(\boldsymbol{H_1})^{-1}\boldsymbol{K_1}$$

$$\boldsymbol{H_1} = \begin{bmatrix} 1 & 1 \\ \mu_\mathrm{A}\alpha_\mathrm{A} & -\mu_\mathrm{A}\alpha_\mathrm{A} \end{bmatrix}$$

$$\boldsymbol{K_1} = \begin{bmatrix} e^{i\alpha_\mathrm{B}a} & e^{-i\alpha_\mathrm{B}a} \\ \mu_\mathrm{B}\alpha_\mathrm{B}e^{i\alpha_\mathrm{B}a} & -\mu_\mathrm{B}\alpha_\mathrm{B}e^{-i\alpha_\mathrm{B}a} \end{bmatrix}$$

$$\boldsymbol{H_2} = \begin{bmatrix} e^{i\alpha_\mathrm{A}a_\mathrm{A}} & e^{-i\alpha_\mathrm{A}a_\mathrm{A}} \\ \mu_\mathrm{A}\alpha_\mathrm{A}e^{i\alpha_\mathrm{A}a_\mathrm{A}} & -\mu_\mathrm{A}\alpha_\mathrm{A}e^{-i\alpha_\mathrm{A}a_\mathrm{A}} \end{bmatrix}$$

$$\boldsymbol{K_2} = \begin{bmatrix} e^{i\alpha_\mathrm{B}a_\mathrm{A}} & e^{-i\alpha_\mathrm{B}a_\mathrm{A}} \\ \mu_\mathrm{B}\alpha_\mathrm{B}e^{i\alpha_\mathrm{B}a_\mathrm{A}} & -\mu_\mathrm{B}\alpha_\mathrm{B}e^{-i\alpha_\mathrm{B}a_\mathrm{A}} \end{bmatrix}$$

Application of Floquet-Bloch boundary condition to the nth lattice yields

$$\boldsymbol{\Psi}_{\mathbf{nB}} = e^{ika}\boldsymbol{\Psi}_{(\boldsymbol{n-1})\mathbf{B}}, \tag{10}$$

where k is a Bloch wave vector.

Thus, a standard matrix eigenvalue problem can be obtained according to Eqs. (9) and (10)

$$\left|\boldsymbol{T} - \mathrm{e}^{\mathrm{i}ka}\mathbf{I}\right| = 0, \tag{11}$$

where $\mathbf{I}$ is a 2×2 unit matrix. Eq. (11) can be rewritten as

$$\begin{aligned}\cos(ka) = {} & \cosh(\alpha_{\mathrm{A}} a_{\mathrm{A}}) \cosh(\alpha_{\mathrm{B}} a_{\mathrm{B}}) \\ & + \frac{1}{2}\left(\frac{\alpha_{\mathrm{A}}\rho_{\mathrm{A}} c_{\mathrm{A}}^2}{\alpha_{\mathrm{B}}\rho_{\mathrm{B}} c_{\mathrm{B}}^2} + \frac{\alpha_{\mathrm{B}}\rho_{\mathrm{B}} c_{\mathrm{B}}^2}{\alpha_{\mathrm{A}}\rho_{\mathrm{A}} c_{\mathrm{A}}^2}\right) \sinh(\alpha_{\mathrm{A}} a_{\mathrm{A}}) \sinh(\alpha_{\mathrm{B}} a_{\mathrm{B}}),\end{aligned} \tag{12}$$

The dispersion curves of 1D PnCs can be obtained by solving the above equation.

Assuming that waves are incident from a semi-infinite material A, passing through the PnCs composed of material A and material B with m periods, and transmitted to a semi-infinite material B, then according to Eq. (9), the relationship between the amplitudes of incident and reflected waves in the mth lattice and the 0th lattice is as follows:

$$\boldsymbol{\psi}_{m\mathbf{B}} = \boldsymbol{T}^m \boldsymbol{\psi}_{0\mathbf{A}}, \tag{13}$$

where m represents the number of periods of the PnCs, and $\boldsymbol{T}^m$ can be abbreviated as follows:

$$\boldsymbol{T}^m = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$$

Then, the amplitude of the transmitted wave, denoted by $C_{m\mathrm{B}}$, can be obtained according to Eq. (13), which is expressed as

$$C_{m\mathrm{B}} = \frac{2\mathrm{e}^{-\alpha_{\mathrm{B}} m a}}{\frac{\mu_{\mathrm{B}}\alpha_{\mathrm{B}}}{\mu_{\mathrm{A}}\alpha_{\mathrm{A}}} T_{11} - \mu_{\mathrm{B}}\alpha_{\mathrm{B}} T_{12} - \frac{1}{\mu_{\mathrm{A}}\alpha_{\mathrm{A}}} T_{21} + T_{22}} C_0, \tag{14}$$

where C_0 is the amplitude of the incident wave passing through the PnCs with m periods. The energies of the incident wave and transmitted wave are, respectively,

$$P_{\mathrm{in}} = \frac{1}{2}\rho_{\mathrm{A}} c_{\mathrm{A}} \omega^2 |C_0|^2, \tag{15}$$

$$P_{\mathrm{out}} = \frac{1}{2}\rho_{\mathrm{B}} c_{\mathrm{B}} \omega^2 |C_{m\mathrm{B}}|^2, \tag{16}$$

where c_j is the velocity component of SH waves in the x direction in material j (j = A, B).

In order to illustrate the attenuation performance of the PnCs, the logarithm of the ratio of the transmitted wave energy to the incident wave energy is defined as the energy transmission coefficient, namely

$$\mathcal{T} = \ln\left(\frac{P_{\mathrm{out}}}{P_{\mathrm{in}}}\right), \tag{17}$$

Then, the energy transmission spectrum of 1D PnCs can be obtained according to the above equation.

Assuming that a 1D PnC is composed of aluminum ($\mu_A = 28.7\,\text{GPa}$ and $\rho_A = 2730\,\text{kg/m}^3$) and epoxy ($\mu_B = 1.59\,\text{GPa}$ and $\rho_B = 1180\,\text{kg/m}^3$), which are material A and material B, respectively. The thicknesses of the two materials are both $0.05\,\text{m}$, and the incident angle of SH waves, namely θ, is $0.5\,\text{rad}$. According to Eq. (12), the dispersion curves can be obtained as shown in Fig. 2A. The energy transmission spectrum for $m = 8$ is calculated by Eq. (17) and shown in Fig. 2B where $\Omega = \omega a/(2\pi c_A)$ is dimensionless frequency.

3. Forward prediction of phononic crystals with neural networks

In this section, the dispersion curve and energy transmission spectrum predictions of 1D PnCs are discussed. The generation of datasets and the choice of neural networks are introduced in details. Two simple neural networks, MLP and RBF neural networks, are adopted and compared. Three prediction cases are considered, such as one-parameter, two-parameter, and three-parameter predictions.

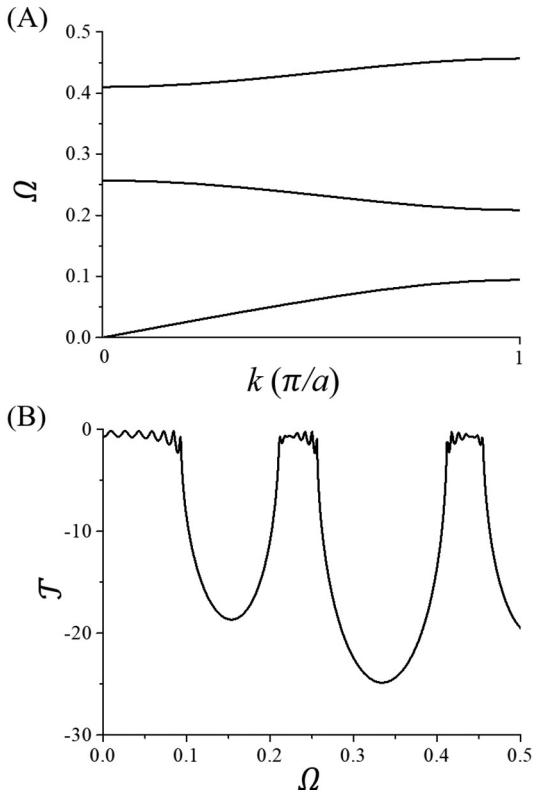


Fig. 2 Dispersion curves (A) and energy transmission spectrum (B).

3.1 Predicting the dispersion curves of one-dimensional phononic crystals

Dispersion curves of PnCs reveal the relationship of Bloch wave vectors and frequencies, from which some physical characteristics of PnCs, such as bandgaps which are located at none eigenvalue areas in dispersion curves, can be discovered and exploited. Hence, rapid prediction of dispersion curves is helpful for researchers to study the properties of structures.

3.1.1 Datasets for the prediction of dispersion curves

It is important for neural networks to establish suitable and reasonable datasets, which are a prerequisite for getting a good result. Generally, datasets are randomly divided into a training set, a validation set, and a testing set. Neural networks learn the inherent laws of data though the training set. The validation set is used to assist in adjusting neural networks during training, which can be used to judge whether the networks is overfitting or underfitting (overfitting means the performance of neural networks on the training set is far better than that on the validation set; underfitting means the neural networks perform very badly on both the training and validation sets). The testing set is responsible for evaluating the neural networks after being completed, which is never used during training and adjusting. The three datasets perform their duties to ensure the accuracy and versatility of neural networks.

From Eq. (12), one may notice that for SH waves, the dispersion curves of 1D PnCs are mainly related to geometry size, shear modulus, and mass density, and it is known that the shear modulus ratio ($\overline{\mu} = \mu_B/\mu_A$), mass density ratio ($\overline{\rho} = \rho_B/\rho_A$), and filling fraction ($\overline{a} = a_B/a$) are the key physical and geometric parameters affecting the bandgaps of PnCs [24]. Hence, these three parameters are considered when data are generated, and corresponding dispersion curves are calculated by Eq. (12). Assuming that material A, lattice constant, and incident angle are fixed, the shear modulus and mass density of material A is set to be 20 GPa and 3000 kg/m^3, respectively, the lattice constant is 1 m, and the incident angle is 0.5 rad. The geometric and physical parameters of material B are variables. For one-parameter (filling fraction $\overline{a}$), two-parameter (shear modulus ratio $\overline{\mu}$ and mass density ratio $\overline{\rho}$), and three-parameter ($\overline{a}$, $\overline{\mu}$, and $\overline{\rho}$) changes, there are corresponding three groups of datasets named as dataset A, dataset B, and dataset C, respectively. Dataset A includes training set A (10 sets of data), validation set A (2 sets of data), and testing set A (2 sets of data), where filling fraction varies from 0.3 to 0.75. Dataset B includes training set B (100 sets of data), validation set B (20 sets of data), and testing set B (20 sets of data), where shear modulus ratio and mass density ratio vary from 0.005 to 0.095 and 0.1667 to 0.5667, respectively. Dataset C includes training set C (1000 sets of data), validation set C (100 sets of data), and testing set C (100 sets of data), where filling fraction, shear modulus ratio, and mass density ratio vary from 0.3 to 0.75, 0.005 to 0.095, and 0.1667 to

0.5667, respectively. The first three bands in the dispersion curves are taken into account and each band is represented by 101 discrete points. It is worth noting that geometric or/and physical parameters of 1D PnCs are taken as the input of neural networks and dispersion curves are taken as the labels (labels are the true samples from datasets, which correspond to the output of neural networks) and there is no intersection among these training, validation, and testing sets.

3.1.2 Neural networks predicting dispersion curves

In this section, we introduce two types of neural networks, which are multilayer perceptron (MLP) and radial basis function (RBF) neural network. They are simple and easy for application.

MLP is composed of many fully connected layers as shown in Fig. 3A. In each layer, there are a lot of artificial neurons which are disconnected, and each neuron is connected to all neurons in the previous and back layers. A neuron consists of a weight vector, a bias, and an activation function, which is defined as

$$O = f(\boldsymbol{w}\boldsymbol{x} + b), \tag{18}$$

where $\boldsymbol{w}$ is weight vector; b is bias; f is activation function; $\boldsymbol{x}$ is input data; O is output value. A satisfied MLP can be obtained by adjusting weights and biases. It is general for linear regression problems to take mean squared error function as the loss function of MLP, which makes neural networks able to estimate posterior probabilities for finite samples with high accuracy [25,26]. Mean squared error function is written as

$$\mathcal{L}_{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left(y_{\mathrm{L}}^{(i)} - y_{\mathrm{O}}^{(i)}\right)^2, \tag{19}$$

where $y_{\mathrm{L}}^{(i)}$ is the ith set of label; $y_{\mathrm{O}}^{(i)}$ is the output of neural networks corresponding to $y_{\mathrm{L}}^{(i)}$; and M is the number of data. Adam optimizer [27] is used to improve the gradient descent

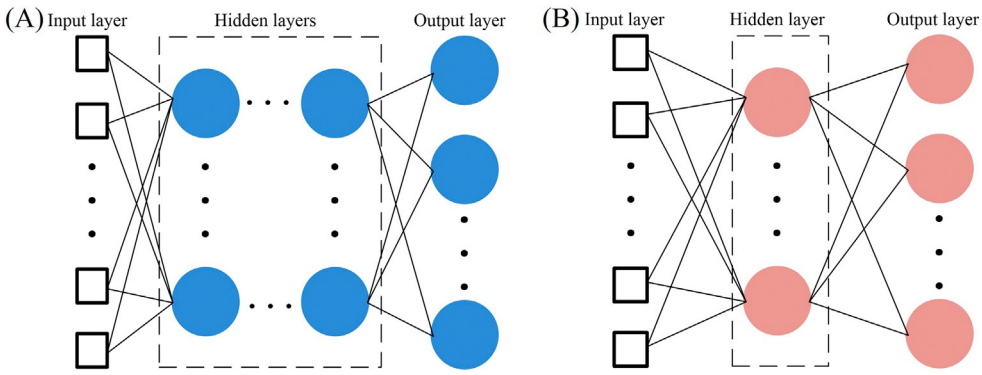


Fig. 3 Multilayer perceptron (A) and radial basis function neural network (B).

method, which is conducive to the convergence of neural networks. The activation function in each layer is "Tanh" as follows

$$f_{\tanh} = \frac{e^{p} - e^{-p}}{e^{p} + e^{-p}}, \tag{20}$$

where p is the preactivated value.

RBF neural network can approximate arbitrary nonlinear functions. With good generalization ability, it is able to learn complicated laws in a system, and its learning efficiency is remarkable. RBF neural network is composed of an input layer, a hidden layer, and an output layer as shown in Fig. 3B. For the RBF neural network, there is no weight connecting between the input layer and the hidden layer but the weight connecting between the output layer and the hidden layer. RBFs can calculate the distance or similarity between the inputs and the centers of the hidden layer. The farther the distance is or the lower the similarity is, the smaller the activation of a neuron is, and the less obvious its effect is. Take multivariate quadratic function as the radial basis function, which can be written as

$$f_{RB} = \sqrt{1 + \delta|\boldsymbol{x} - \boldsymbol{c}|^{2}}, \tag{21}$$

where $\boldsymbol{c}$ is the center of the radial basis function, and δ is an adjustable parameter. K-means algorithm is adopted to determine $\boldsymbol{c}$ which should obey the following condition:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{K} \sum_{\boldsymbol{x}\in\mathbf{S}_i} |\boldsymbol{x} - \mathbf{c}_i|^{2}, \tag{22}$$

where $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \ldots, \mathbf{S}_K\}$ is the collection of all input sample data, where $\mathbf{S}_i$ ($i = 1, 2, 3, \ldots, K$) is a subset of $\mathbf{S}$, which is composed of one or more samples; and $\mathbf{c}_i$ is the mean of $\mathbf{S}_i$. Here, linear regression method is used to calculate the weights between the hidden layer and the output layer. Compared to gradient descent method, linear regression method saves the training time and its model is simpler. The weight calculated by linear regression method is as follows:

$$\boldsymbol{w} = \left(\boldsymbol{y}_L^{\mathrm{T}}\boldsymbol{y}_{\mathbf{L}}\right)^{-1}\boldsymbol{y}_L^{\mathrm{T}}\boldsymbol{y}_{\mathbf{O}} \tag{23}$$

3.1.3 Choosing suitable neural networks

It is important to construct a reasonable and suitable neural network for data-driven work. If the neural network is too simple, it has a weak simulation performance; if too complicated, its learning ability may deteriorate and the calculation efforts will increase. In addition, choosing suitable activation functions is necessary for different problems, and adopting an appropriate algorithm to update the neural network parameters is conducive to speeding up the convergence of the network and avoiding falling into a local optimum. A validation set can be used to preliminarily evaluate the

performance of a neural network. Hence, judging whether the neural network is appropriate can refer to its performance on the validation set.

Firstly, in order to describe the accuracy of predicted dispersion curves by neural networks, Euclidean distance between the predicted dispersion curves and the label is adopted, which is as follows:

$$e = \sqrt{|\boldsymbol{\gamma}_{\mathbf{L}} - \boldsymbol{\gamma}_{\mathrm{O}}|^2}, \tag{24}$$

It should be noticed that the smaller the Euclidean distance is, the higher the prediction accuracy is.

In this section, the architectures of the MLPs and RBF neural networks are discussed for three cases in the validation sets, involving one-parameter, two-parameter, and three-parameter changes, respectively. The optimal architectures are determined by comparing the mean errors of corresponding validation sets. Here, the prediction error is represented with the Euclidean distance (Eq. 24) between labels and predicted values.

For the one-parameter prediction, the following four architectures of the MLPs are compared in validation set A:

MLP-1-1: 1–30–30–30–303
MLP-1-2: 1–30–30–30–30–303
MLP-1-3: 1–30–30–30–30–30–303
MLP-1-4: 1–30–30–30–30–30–30–303

For the two-parameter prediction, the following four architectures of the MLPs are compared in validation set B:

MLP-2-1: 2–300–100–100–100–303
MLP-2-2: 2–300–100–100–100–100–303
MLP-2-3: 2–300–100–100–100–100–100–303
MLP-2-4: 2–300–100–100–100–100–100–100–303

For the three-parameter prediction, the following four architectures of the MLPs are compared in validation set C:

MLP-3-1: 3–600–300–100–100–303
MLP-3-2: 3–600–300–100–100–100–303
MLP-3-3: 3–600–300–100–100–100–100–303
MLP-3-4: 3–600–300–100–100–100–100–100–303

Here "1," "2," or "3" is the dimension of the input layer, "303" is the dimension of the output layer, and others are the number of the neurons in their corresponding hidden layers. Fig. 4A shows the mean errors of the validation sets of the MLPs under three cases. It can be seen that for the one-parameter prediction, the mean error of "MLP-1-3" is the smallest, for the two-parameter prediction, "MLP-2-3" and "MLP-2-4" have a similar accuracy, but "MLP-2-3" has less hidden layers than "MLP-2-4," and for the three-parameter prediction, "MLP-3-3" is the best choice.

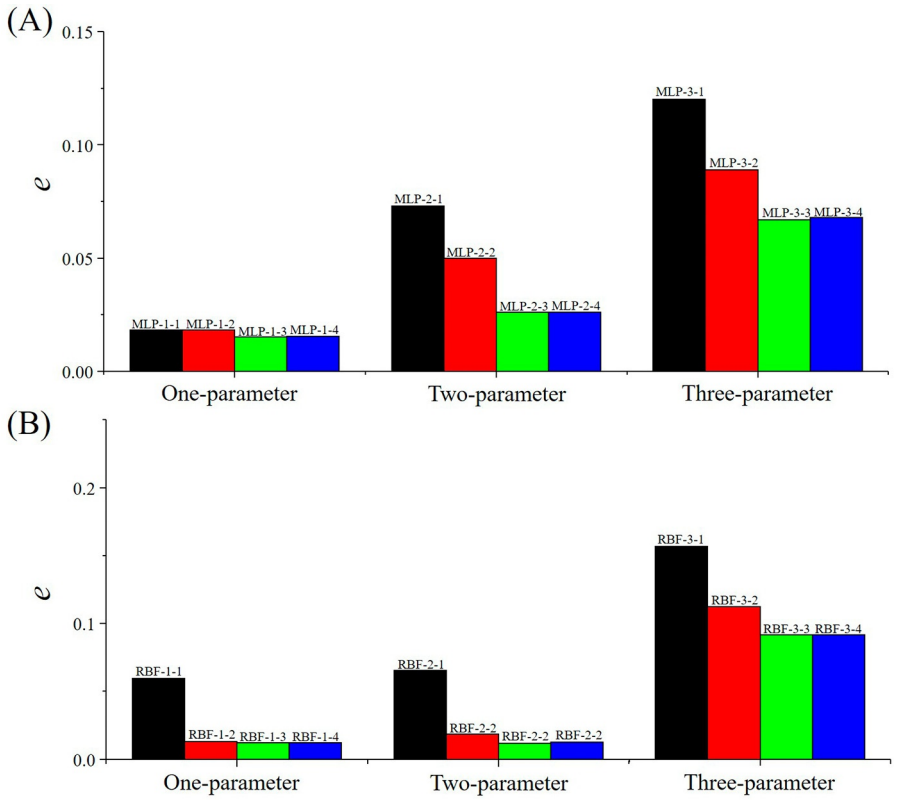


Fig. 4 Comparisons of the mean errors in the validation sets for different multilayer perceptrons (A) and radial basis function neural networks (B).

Similar to the comparison of MLPs, for the one-, two-, and three-parameter predictions, the following 12 architectures of the RBF neural networks are, respectively, compared in the validation sets:

RBF-1-1: 1–5–303
RBF-1-2: 1–10–303
RBF-1-3: 1–15–303
RBF-1-4: 1–20–303
RBF-2-1: 2–50–303
RBF-2-2: 2–100–303
RBF-2-3: 2–150–303
RBF-2-4: 2–200–303
RBF-3-1: 3–200–303
RBF-3-2: 3–400–303
RBF-3-3: 3–600–303
RBF-3-4: 3–800–303

Fig. 4B gives the mean errors of the validation sets of the RBF neural networks under three cases. It can be seen that "RBF-1-3," "RBF-2-3," and "RBF-3-3" are the optimal architectures for the first, second, and third cases, respectively.

It can be seen from the above comparisons that for MLP, too shallow hidden layers or too few neurons will make prediction accuracy worse. If the hidden layers are too deep or the number of neurons is too large, the performance of the MLPs in the validation sets will become poor or the amount of calculation will increase without the improvement of prediction accuracy. For RBF neural networks, too few neurons will lead to poor performance, and too many neurons will greatly increase the amount of calculation without improved results. Hence, it is very important to choose a properly constructed neural network model.

3.1.4 Predicted dispersion curves in the testing sets

Dataset A is used to train, validate, and test the neural networks for the one-parameter prediction, and it has 14 sets of data where 10 form training set A, 2 form validation set A, and 2 form testing set A. According to above analysis in Section 3.1.3, "MLP-1-3" and "RBF-1-3" are the optimal choices for one-parameter prediction. The predicted dispersion curves in testing set A by the two neural networks are shown in Fig. 5, where the black solid lines are the true dispersion curves, namely label, the red triangle dots are the predicted ones by the MLP, and the cyan dashed lines are those by the RBF neural network. It can be seen that the predicted dispersion curves are in good agreement with the labels. The mean prediction errors of the MLP and the RBF neural network are 0.02 and 0.01, respectively. The two neural networks exhibit good performances on predicting the dispersion curves of 1D PnCs with different filling fractions, and the RBF neural network is better.

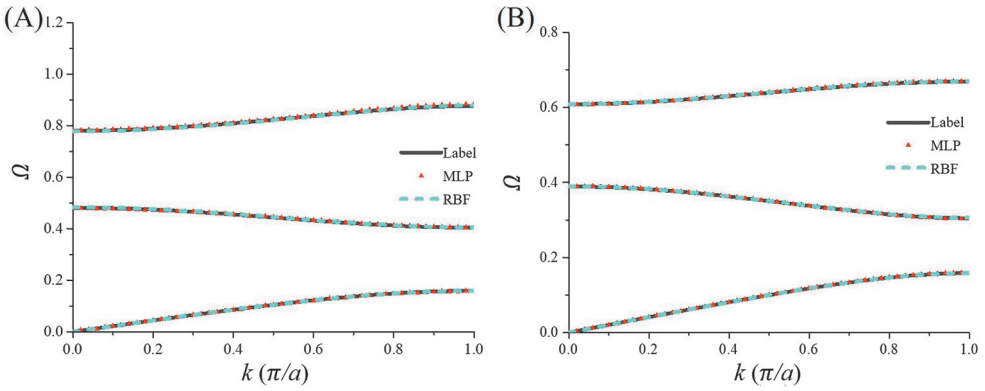


Fig. 5 Predicted dispersion curves by the multilayer perceptron and radial basis function neural network in testing set A.

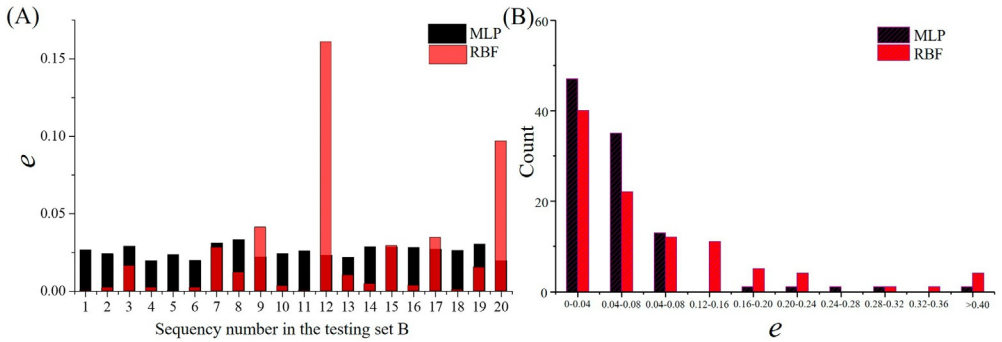


Fig. 6 Each prediction error in testing set B (A) and distributions of prediction errors in testing set C (B).

For the two-parameter prediction, the dispersion curves of 1D PnCs with different shear modulus ratios and mass density ratios are predicted. Here, the architecture of the MLP is "MLP-2-3," and the architecture of the RBF neural network is "RBF-2-3." Due to the space limit, predicted dispersion curves by the two neural networks in testing set B are not given, but the prediction errors are shown in Fig. 6A. It can be seen the prediction errors by the MLP are all close to 0.025, but those by the RBF neural network ranges from 0 to 0.17. Obviously, both the two neural networks present satisfied predictions with high precision and their mean prediction errors are 0.03 and 0.02, respectively. Although the RBF neural network has less mean error, the stability of its prediction performance is not as good as that of the MLP as shown in Fig. 6A where there are two prediction errors far greater than others.

Dataset C, which is composed of training set C, validation set C, and testing set C, is used to train, validate, and test neural networks. For the three-parameter prediction, i.e., the prediction of the dispersion curves for different filling fractions, shear modulus ratios, and mass density ratios, the architectures of the MLP and the RBF neural network are "MLP-3-3" and "RBF-3-3," respectively. In testing set C, the mean prediction errors of the two neural networks are 0.06 and 0.09, respectively, and the distributions of prediction errors are shown in Fig. 6B. It can be seen that the proportion of prediction errors less than 0.08 by the MLP is more than that by the RBF neural network. Hence, the performances of the two neural networks are still remarkable, and the MLP performs much better.

From the above results, it can be seen that RBF neural network is good at solving uncomplicated problems which have less data and simple rules, and it takes a very short time (0.01 s for the one-parameter prediction and 1.03 s for the two-parameter prediction) to complete learning without iteration because its parameters are calculated by a matrix operation given in Eq. (23). For a complex problem, such as the three-parameter prediction, MLP has greater advantages due to its powerful simulation and generalization ability caused by multilayer and many neurons. In summary, RBF neural network is a

good choice for a simple issue with small samples, and MLP is more suitable for a complex issue with a big data.

3.2 Predicting the energy transmission spectrums of one-dimensional phononic crystals

In the above section, we introduce the prediction of dispersion curves of 1D PnCs with infinite periods. For a PnC with finite periods, the energy loss from its input to output can better illustrate the physical phenomena it possesses. In this section, the prediction of the energy transmission spectrums of 1D PnC is presented in detail.

3.2.1 Datasets for the prediction of energy transmission spectrums

Energy transmission spectrum expresses the energy loss of the incident waves passing through the 1D PnC with finite periods over frequency. Similar to those in Section 3.1.1, changes in shear modulus ratio, mass density ratio, and filling fraction are considered, and the corresponding three prediction cases are taken into account. We assume that the number of basic units is 8, the periodic constant is 1 m, the incident angle is 0.5 rad, and material A is the same as that in Section 3.1.1. For one-parameter prediction, filling fraction is considered as input, which varies from 0.3 to 0.75, and there are training set D (10 sets of data), validation set D (2 sets of data), and testing set D (2 sets of data) which are collectively referred to as dataset D. For two-parameter prediction, shear modulus ratio and mass density ratio vary from 0.055 to 0.1 and 0.367 to 0.667, respectively, and the dataset is called dataset E which includes training set E (100 sets of data), validation set E (20 sets of data), and testing sets E (20 sets of data). For three-parameter prediction, dataset F is generated, which is composed of training set F (1000 sets of data), validation set F (100 sets of data), and testing set F (100 sets of data), where filling fraction, shear modulus ratio, and mass density ration vary from 0.3 to 0.75, 0.055 to 0.1, and 0.367 to 0.667, respectively. Energy transmission spectrum from 0 to 3000 Hz is termed as label and is represented by 100 discrete points, which can be obtained according to Eq. (17).

3.2.2 Neural networks predicting energy transmission spectrums

The performances of MLP and RBF neural network are compared for the prediction of the energy transmission spectrums. Here, the selection process of the two neural networks is omitted, which is similar with that in Section 3.1.3. For the one-parameter prediction, the architectures of the MLP and the RBF neural network are "1–100–60–30–20–60–100" and "1–15–100," respectively, where the first number "1" is the dimension of the input layer, the last number "100" is the dimension of the output layer, and others are the numbers of neurons in the corresponding hidden layers. For the two-parameter prediction, they are "2–100–60–30–20–60–100" and "2–100–100," respectively. For the three-parameter prediction, the architectures of the MLP and the RBF neural network are "3–300–100–80–80–80–80–60–60–60–60–100" and "3–1000–100," respectively. It should

be noticed that the activation functions of the neurons in the last layers of the three MLPs are all "Elu" activation function and others are "Tanh" activation function. "Elu" activation function is expressed as

$$f_{\text{elu}} = \begin{cases} p, p > 0 \\ \varepsilon(e^p - 1), p \leq 0 \end{cases}, \tag{25}$$

where ε is an adjustable parameter. The training methods of the two types of neural networks are the same as those in Section 3.1.

3.2.3 Predicted energy transmission spectrums in the testing sets

For the one-parameter prediction, only 10 sets of data in training set D are used for the learning of the neural networks. The predicted energy transmission spectrums in testing set D by the MLP and the RBF neural network are shown in Fig. 7 [28], where the black

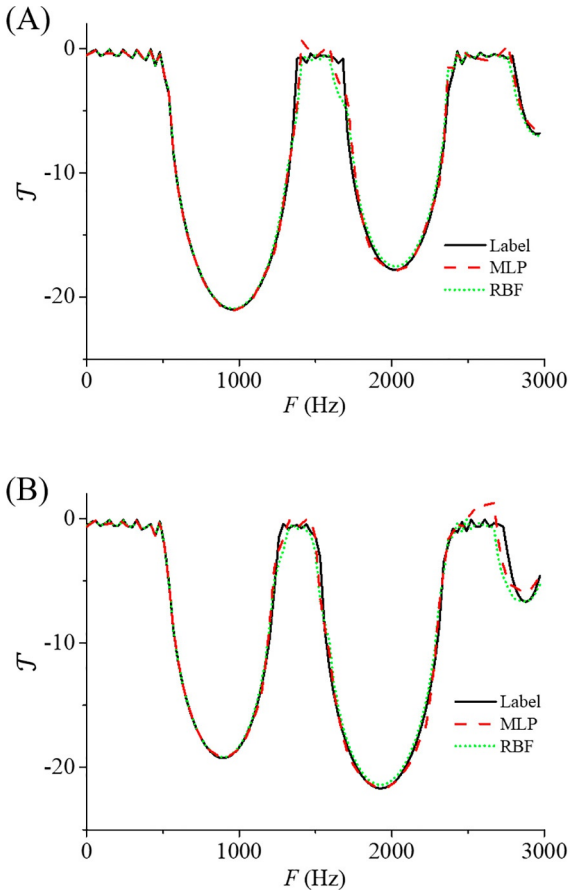


Fig. 7 Predicted energy transmission spectrums by multilayer perceptron and radial basis function neural network.

solid lines represent the true energy transmission spectrums, namely labels, the red dashed lines represent the predicted results by the MLP, and the green dotted lines represent those by the RBF neural network. Obviously, the predicted energy transmission spectrums are highly consistent with the labels, and the results by the two neural networks fit the labels better in lower frequency zone, e.g., from 0 to 1000 Hz. The prediction errors of the two results by the MLP are 0.39 and 0.48, and those by the RBF neural network are 0.36 and 0.47, which are calculated according to Eq. (24) in Section 3.1.3. The performance of the two neural networks is very close. However, the training time of the RBF neural network (0.3 s) is far less than that of the MLP (30.5 s).

Training set E is adopted to train the neural networks for the two-parameter prediction. In testing set E, the mean and variance of prediction errors between the predicted energy transmission spectrums by the MLP and labels are 0.13 and 0.0015, respectively, those by the RBF neural network and labels are 0.17 and 0.0016, respectively. Obviously, the MLP is better than the RBF neural network in the two-parameter prediction.

One thousand sets of data in training set F are used to train the MLP and the RBF neural network for the three-parameter prediction. The architectures of the two neural networks are given in Section 3.2.2. The mean and variance of errors in testing set F for the MLP are 0.2 and 0.0013, respectively, and those for the RBF neural network are 0.81 and 4.2972, respectively. It can be seen for the three-parameter prediction of energy transmission spectrums, which is a more complex problem, the performance of the MLP is far stronger than that of the RBF neural network.

To sum up, MLP is better than RBF neural network in general on the prediction of energy transmission spectrums. For the one-parameter prediction which is a relatively simple issue with a small data, the prediction performance of the two neural networks is similar. However, as the difficulty of the problem increases, the advantages of MLP become more and more obvious. Especially, MLP is far better than RBF neural network in dealing with the three-parameter prediction problem. Hence, MLP is a better choice than RBF neural network in the prediction of energy transmission spectrums.

4. Designing phononic crystals with neural networks

In the above section, the predictions of dispersion curves and energy transmission spectrums are introduced, both of which are forward problems. However, how to automatically and rapidly get a structure meeting the expectations is more worthy of being explored. In this section, the inverse design of 1D PnCs is discussed, and two types of neural networks used to design structures are compared, which are MLP and autoencoder (AE) with a pretrained decoder, respectively.

MLP can give results without iteration by learning the mapping relation from input to label. It usually takes a very short time to complete an output, generally on the order of milliseconds. For the design of 1D PnCs, the input is design targets and the label is

geometric or/and physical parameters, namely design parameters. As we all know, a design target usually corresponds to many different structures, which is a non-mapping issue. Using MLP to simulate a law about 1D PnCs design seems inappropriate. Hence, we may need a more complicated neural network to achieve the inverse design of 1D PnCs better.

It is well known that the physical characteristics are determined for a given structure. A simple neural network, like MLP, is able to learn well the relation from feature parameters of a PnC to its physical characteristics, which is discussed in Section 3. For realizing the design of 1D PnCs, the physical law can be firstly captured by an MLP. And then, a neural network is used to search the structure meeting the expectations according to the feedback of the MLP which simulates the physical law of 1D PnCs. The constructed model is an AE with a pretrained decoder. An AE is composed of an encoder and a decoder connected by a code layer. Its aim is to make output close to input. The pretrained decoder in the AE is the MLP which learnt the physical law of 1D PnCs. A targeted bandgap can be input into the AE, and the output is a predicted bandgap which can be close to the targeted by adapting neural network parameters in the encoder and code layer. Feature parameters of 1D PnCs are the outputs of the code layer, which are a bridge between the encoder and the decoder. The feature parameters making the predicted bandgaps by the decoder close to the targeted, which are given by the encoder, can be taken as the designed results. Hence, the AE with a decoder which captures the physical law is expected to implement the design of 1D PnCs well. The time required for designing a PnC by an AE with a pretrained decoder varies from less than 1 s to several seconds due to multiple updates of neural network parameters. Fig. 8 shows the architecture of the AE with a pretrained decoder.

Here, three design cases are discussed, and the lower and upper limits of bandgaps are termed as design targets. The first case is to design filling fractions, the second is to design shear modulus ratios and mass density ratios, and the third is to design the three parameters above simultaneously.

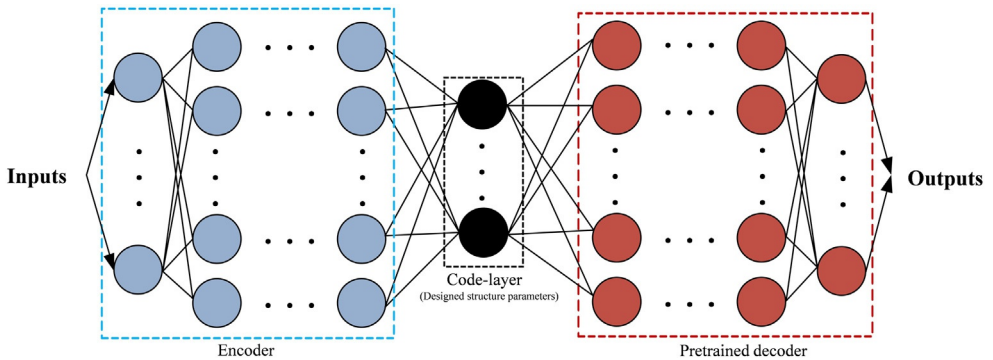


Fig. 8 Autoencoder with a pretrained decoder where the designed structure parameters are output in the code layer.

Table 1 Values of filling fraction, shear modulus ratio, and mass density ratio.

Filling fraction	Shear modulus ratio	Mass density ratio
0.30	0.005	0.1667
0.35	0.015	0.2000
0.40	0.025	0.2333
0.45	0.035	0.2667
0.50	0.045	0.3000
0.55	0.055	0.3333
0.60	0.065	0.3667
0.65	0.075	0.4000
0.70	0.085	0.4333
0.75	0.095	0.4667

4.1 Datasets for the inverse design of phononic crystals

We take 10 values for each of these three parameters shown in Table 1. Then, 1000 of PnCs with different parameters are obtained by iteratively traversing all combinations of these values. Without loss of generality, we take $a = 1\,\text{m}$, $\mu_A = 20\,\text{GPa}$, $\rho_A = 3000\,\text{kg/m}^3$, and θ is 0.5 rad. Then the dispersion curves of these PnCs can be calculated according to Eq. (12), and their bandgaps can be determined. Here, the first two bandgaps are taken into account. Hence, we get 1000 sets of data (training set G) for training MLP and AE's decoder. Each set of data consists of PnCs' parameters and the lower and upper limits of the corresponding bandgaps. The testing sets only contain design targets which are not in training set G.

4.2 One-parameter design

The 10 sets of data in training set G where shear modulus ratio and mass density ratio are 0.055 and 0.3333, respectively, are used to train the neural networks for the one-parameter design. The architecture of the MLP is "4–40–20–10–1" where the first number "4," is the dimension of the input or target, the last number "1," is the dimension of the output or filling fraction, and others are the numbers of the neurons in the corresponding hidden layers. The architecture of the AE with a pretrained decoder is "4–20–20–10–1–20–20–10–4" where the first number "4" and the last number "4" are respectively the input and output dimension, the middle number "1" is the number of the neuron in the code layer, which outputs filling fraction, and the remaining numbers are the numbers of the neurons in the corresponding hidden layers. After the two neural networks are trained, three sets of targeted bandgaps which are not appeared in training set G are input into the MLP and the AE with a pretrained decoder, respectively, and then the designed 1D PnCs with filling fractions which meet these three sets of targeted bandgaps, can be obtained. To verify whether the first two bandgaps of the designed PnCs by the two neural networks meet the requirements, the bandgaps of the designed PnCs are calculated by Eq. (12). Fig. 9 shows the lower and upper limits

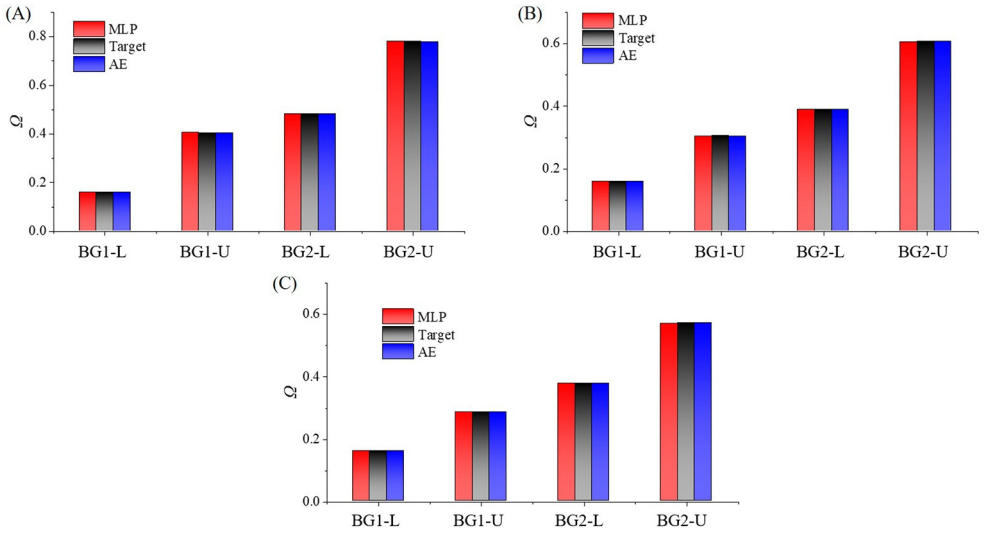


Fig. 9 Comparisons of the targeted bandgaps and the bandgaps of designed structures for the one-parameter design; in the abscissa, BG1-L and BG2-L refer to the lower limits of the first and second bandgaps; BG1-U and BG2-U refers to the upper limits of the first and second bandgaps; the filling fractions designed by the MLP and the AE with a pretrained decoder are 0.47798 and 0.48052, respectively (A); the filling fractions designed by the MLP and the AE with a pretrained decoder are 0.66283 and 0.66065, respectively (B); the filling fractions designed by the MLP and the AE with a pretrained decoder are 0.71116 and 0.71001, respectively (C).

of their bandgaps and the targeted bandgaps as comparisons. The filling fractions designed by the MLP and the AE with a pretrained decoder are 0.47798 and 0.48052 in Fig. 9A, 0.66283 and 0.66065 in Fig. 9B, and 0.71116 and 0.71001 in Fig. 9C, respectively. It can be seen that the bandgaps of PnCs designed by the two neural networks are highly consistent with the targets. Both the MLP and AE with a pretrained decoder exhibit good performances in the one-parameter design.

4.3 Two-parameter design

Investigation on the two-parameter design concerning the shear modulus ratio and mass density ratio is carried out. For a certain filling fraction which is set to be 0.5, 100 sets of data in training set G is used to train the MLP and the decoder of the AE for the two-parameter design. The architecture of the MLP is "4–100–50–20–2," and the architecture of the AE with a pretrained decoder is "4–60–30–20–2–100–50–20–4" where the middle number "2" is the number of the neurons in the code layer which outputs shear modulus ratio and mass density ratio. After the two neural networks are trained, 30 sets of targeted bandgaps, which are not appeared in training set G, are input into the MLP and the AE with a pretrained decoder, respectively, and the structure parameters constrained

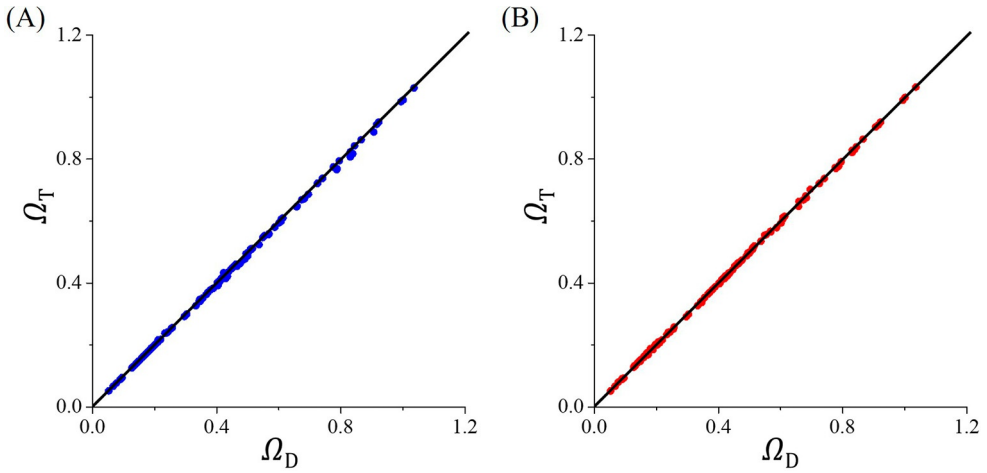


Fig. 10 Correlation between the targeted bandgaps and the bandgaps of designed PnCs by the MLP (A) and the AE with a pretrained decoder (B), respectively, for the two-parameter design.

by the target bandgaps are finally obtained. Fig. 10 shows the correlation between the targeted bandgaps and the bandgaps of the 30 designed PnCs by the MLP and the AE with a pretrained decoder, respectively, and the correlation coefficients in Fig. 10A and B are 0.9998 and 0.9999, respectively, where Ω_T represents the lower or upper limit of the targeted bandgap and Ω_D represents the lower or upper limit of the bandgap of designed 1D PnCs. It can be seen that the bandgaps of the designed PnCs by the two neural networks are highly correlated with the targeted bandgaps. The results of the two neural networks are excellent and stable in performance for the two-parameter design.

4.4 Three-parameter design

For the three-parameter design, the architecture of the MLP is "4–100–50–30–3," and the architecture of the AE with a pretrained decoder is "4–60–40–30–3–100–60–40–40–204" where "3" refers to the number of the neurons in the code layer. Training set G is adopted to train the MLP and the decoder of the AE. The 300 sets of targeted bandgaps, which are not in training set G, are input into the two neural networks after being trained, and the structure parameters can be obtained. Fig. 11 shows the correlation between the targeted bandgaps and the bandgaps of the 300 designed PnCs by the MLP and AE with a pretrained decoder, respectively. Obviously, the bandgaps of the PnCs designed by the AE with a pretrained decoder are far closer to the targeted bandgaps than those designed by the MLP. Besides, the correlation coefficients of the designed results by the MLP and AE with a pretrained decoder are 0.9986 and 0.9999, respectively. Compared with the two-parameter design, the correlation between the designed results

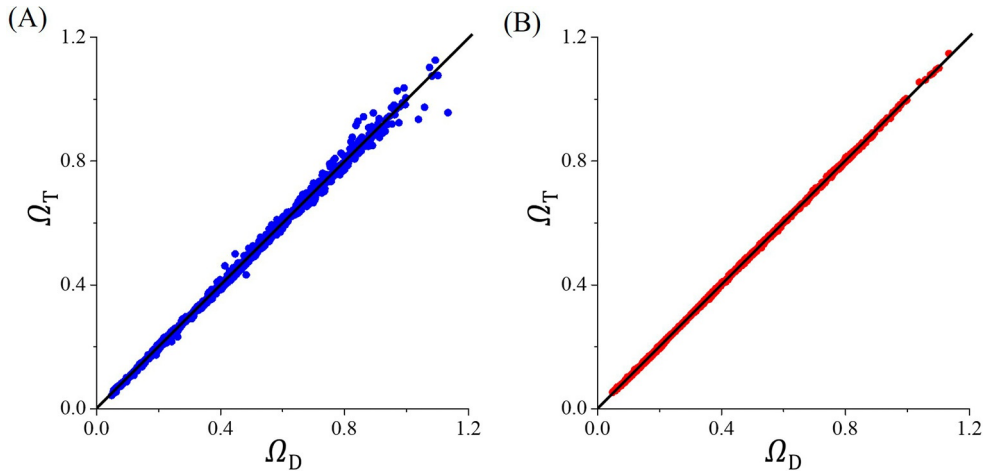


Fig. 11 Correlation between the targeted bandgaps and the bandgaps of designed PnCs by the MLP (A) and the AE with a pretrained decoder (B), respectively, for the three-parameter design.

by the MLP and targets are significantly weakened, but that by the AE with a pretrained decoder does not change. Hence, the AE with a pretrained decoder is far better than the MLP for the three-parameter design.

5. Summary

Two types of neural networks, multilayer perceptron (MLP) and radial basic function (RBF) neural network, are adopted to predict the dispersion curves and energy transmission spectrums of 1D PnCs in this chapter. Three conditions are taken into account, including the one-parameter prediction, two-parameter prediction, and three-parameter prediction. For the first two conditions which are relatively simple problems, MLP and RBF neural networks have a similar prediction performance and the latter has a faster training speed than the former. However, MLP has a clear advantage for a complex problem like the three-parameter prediction, and its accuracy is far higher than that of RBF neural network. Hence, simple and fast RBF neural networks are good at solving uncomplicated problems, and MLPs perform better on a big data with a certain degree of difficulty.

For design problems, MLP and AE with a pretrained decoder are used, where the one-parameter design, two-parameter design, and three-parameter design are considered, respectively. MLP achieves the design of 1D PnCs by establishing a mapping relationship from targets to structure parameters, which solves the design problem from data level. AE with a pretrained decoder firstly captures the physical laws of 1D PnCs by the pretrained decoder and then the encoder in the AE can give the designed structure

parameters by multiple iterations. The encoder of the AE plays the role of a scholar and the pretrained decoder of the AE a law. The design process of the AE is like the scholar (the encoder) searching the answer meeting the law (the pretrained decoder) based on a specific target. For the one-parameter and two-parameter design problems, the mapping relationships from targets to structure parameters are simple, and MLP can learn them well, which makes the designed results as good as that of the AE with a pretrained decoder. However, the three-parameter design problem is relatively complicated, leading to the complex of the mapping relationship from targets to structure parameters, and it is difficult for MLP to accurately capture the inherent law of the data. The pretrained decoder of the AE can learn well the physical law of calculating bandgaps according to given structure parameters, and makes the AE better realize the design of 1D PnCs. In summary, MLP is suitable for a simple design issue and the AE with a pretrained decoder for a complicated one.

The methods for the forward prediction and inverse design of 1D PnCs are also suitable for those of 2D or 3D PnCs. With a dataset containing the feature parameters of 2D or 3D PnCs and the corresponding physical characteristics, neural networks can be trained to realize some functions and help people to solve problems more easily and rapidly.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (11772040 and 12172037) and Beijing Natural Science Foundation (8222024).

References

[1] V. Narayanamurti, H.L. Störmer, M.A. Chin, A.C. Gossard, W. Wiegmann, Selective transmission of high-frequency phonons by a superlattice: the "dielectric" phonon filter, Phys. Rev. Lett. 43 (1979), https://doi.org/10.1103/PhysRevLett.43.2012.

[2] M.M. Sigalas, E.N. Economou, Elastic and acoustic wave band structure, J. Sound Vib. 158 (2) (1992) 377–382.

[3] M.M. Sigalas, E.N. Economou, Band structure of elastic waves in two dimensional systems, Solid State Commun. 86 (3) (1993) 141–143.

[4] M.S. Kushwaha, P. Halevi, L. Dobrzynski, B. Djafari-Rouhani, Acoustic band structure of periodic elastic composites, Phys. Rev. Lett. 71 (13) (1993) 2022–2025, https://doi.org/10.1103/PhysRevLett.71.2022.

[5] R. Martínez-Sala, J. Sancho, J.V. Sánchez, V. Gómez, J. Llinares, F. Meseguer, Sound attenuation by sculpture, Nature 378 (6554) (1995) 241.

[6] J.O. Vasseur, P.A. Deymier, B. Chenni, B. Djafari-Rouhani, L. Dobrzynski, D. Prevost, Experimental and theoretical evidence for the existence of absolute acoustic band gaps in two-dimensional solid phononic crystals, Phys. Rev. Lett. 86 (14) (2001) 3012.

[7] Z. Liu, X. Zhang, Y. Mao, Y.Y. Zhu, Z. Yang, C.T. Chan, P. Sheng, Locally resonant sonic materials, Science 289 (5485) (2000) 1734–1736.

[8] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.

[9] B.W.A.C. Farley, W. Clark, Simulation of self-organizing systems by digital computer, Trans. IRE Prof. Group Inform. Theory 4 (4) (1954) 76–84.
[10] P.J. Werbos, New Tools for Prediction and Analysis in the Behavioral Sciences, Harvard University, 1974.
[11] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (8) (1982) 2554–2558.
[12] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[13] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[16] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[17] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, 2013. https://arxiv.org/pdf/1312.6114.pdf.
[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, 2014. https://arxiv.org/pdf/1406.2661.pdf.
[19] D. Finol, Y. Lu, V. Mahadevan, A. Srivastava, Deep convolutional neural networks for eigenvalue problems in mechanics, Int. J. Numer. Methods Eng. 118 (5) (2019) 258–275.
[20] C.X. Liu, G.L. Yu, Predicting the dispersion relations of one-dimensional phononic crystals by neural networks, Sci. Rep. 9 (1) (2019) 1–10.
[21] C.X. Liu, G.L. Yu, G.Y. Zhao, Neural networks for inverse design of phononic crystals, AIP Adv. 9 (8) (2019), 085223.
[22] S.M. Sadat, R.Y. Wang, A machine learning based approach for phononic crystal property discovery, J. Appl. Phys. 128 (2) (2020), 025106.
[23] X. Li, S. Ning, Z. Liu, Z. Yan, C. Luo, Z. Zhuang, Designing phononic crystal with anticipated band gap through a deep learning based data-driven method, Comput. Methods Appl. Mech. Eng. 361 (2020) 112737.
[24] X.Z. Zhou, Y.S. Wang, C. Zhang, Effects of material parameters on elastic band gaps of two-dimensional solid phononic crystals, J. Appl. Phys. 106 (1) (2009), 014903.
[25] M.S. Hung, M.Y. Hu, M.S. Shanker, B.E. Patuwo, Estimating posterior probabilities in classification problems with neural networks, Int. J. Comput. Intell. Organ. 1 (1) (1996) 49–60.
[26] M.D. Richard, R.P. Lippmann, Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Comput. 3 (4) (1991) 461–483.
[27] D.P. Kingma, J. Ba, Ada A Method for Stochastic Optimization, 2014. https://arxiv.org/pdf/1412.6980.pdf.
[28] C.X. Liu, G.L. Yu, Prediction of energy transmission spectrum of layered periodic structures by neural networks, J. Shanghai Jiao Tong Univ. 55 (1) (2021) 88–95.

CHAPTER 4

Nanophotonic devices based on optimization algorithms

Cuicui Lu, Hongyi Yuan, and Nianen Zhang
Beijing Institute of Technology, Beijing, China

1. Introduction

Nanophotonic devices, which take photons as information carriers, are the essential components for photonic chips, and play a key role in optical computing, optical interconnection, and optical networks. However, most of the traditional methods that rely on human experience and physical inspiration for structural design and parameter optimization usually require many resources, and the performance of the designed device is limited. Several years ago, before 2016, we had to sweep each structure parameter related to a wavelength demultiplexer device manually [1], even day after day, cycle after cycle. With the development of computer computing capacity and algorithms, it is an exciting and significant thing that we combine optimization algorithms and conventional numerical methods to design nanophotonic devices. Compared with previous numerical methods, optimization algorithms can help us to find the best pattern, improve efficiency, and save time. The design of nanophotonic devices based on optimization algorithms can break the restrictions of traditional methods and predict new configurations, which is universal and efficient for different materials, different structures, different modes, different wavelengths, etc.

In this chapter, we introduce the typical nanophotonic devices based on the optimization algorithms from physical concepts to their applications. The nanophotonic devices based on the optimization algorithms will be illustrated in this chapter according to the related reports and our research experiences, which can bring inspiration for the efficient design of photonic devices. Here, the algorithms used to design nanophotonic structures are divided into two main types, gradient-based and heuristic.

Gradient-based algorithms used in the nanophotonic design include topology optimization, objective-first algorithm, and other gradient-based optimization algorithms. There are three kinds of topology optimization methods often used in the nanophotonic design, variable density method, level set method, and bi-directional evolutionary structural optimization (BESO). Gradient-based algorithms search for better solutions according to the gradients of the current solution, which usually brings a great computation cost in gradient calculation. To solve this difficulty, the adjoint method is introduced in

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00004-4
Copyright © 2023 Elsevier Inc.
All rights reserved.

gradient's calculation. The algorithms mentioned above will be introduced one by one in the first section.

Heuristic algorithms can be regarded as gradient-independent algorithms, which generate new solutions with different strategies based on randomness but not gradients information. Heuristic algorithms can be further classified into two kinds, the individual-based and the population-based. Individual-based heuristic algorithms include hill-climbing algorithm, direct binary search (DBS) algorithm, simulated annealing algorithm (SAA), and tabu search (TS). Population-based heuristic algorithms include genetic algorithm (GA), differential evolution (DE) algorithm, particle swarm optimization (PSO) algorithm, and ant colony algorithm (ACA). Actually, all these heuristic algorithms can be understood from a broader view of the Monte-Carlo method, whose underlying concept is to use randomness to solve problems that might be deterministic in principle [2,3].

2. Gradient-based algorithms

Gradient-based algorithms usually have good optimization efficiency and perform well in convergence. In recent years, in the design of nanophotonics, many optimization schemes based on gradients have been reported [4], including topology optimization, objective-first algorithm, and other gradient-based algorithms. Topology optimization can be further divided into variable density method, level set method, and bi-directional evolutionary structural optimization (BESO). The objective-first algorithm was proposed to design and experimentally realize on-chip nanophotonic devices by J. Vuckovic's group in recent years and has been widely applied in nanophotonic devices. Other gradient-based algorithms have various workflow but they are all based on the adjoint method, which is a method used for derivative calculation. This section will introduce these algorithms and their applications one by one. For each kind of algorithm, we will first give its main idea of optimization and then give some typical examples achieved with this algorithm.

2.1 Topology optimization

Topology optimization was initially presented by Bendsøe and Kikuchi in their pioneering work in 1988 [5], which aimed to answer the question of how to distribute material in a given area to obtain the best performance. After that, various schemes are proposed and applied in mechanical engineering. With the development of design methods in nanophotonics, different kinds of topology optimization are also introduced into this field, which include variable density method, level set method, and BESO. Here, we will concentrate on these three topology optimization methods including their main ideas and nanophotonic applications in order.

2.1.1 Variable density method

Introduction

Consider a planar binary structure composed of two kinds of materials. How can the material distribution corresponding to the best performance be obtained? One direct idea is dividing the design area into many finite elements with each element filled with one material. We can define a distribution function χ_e to represent this binary structure:

$$\chi_e = \begin{cases} 0, & \text{if material 1 in element;} \\ 1, & \text{if material 2 in element.} \end{cases}$$

The variable density method divides the design area into many finite elements. Each finite element is one design variable, so the parameter space can be extremely enlarged as long as the computation source can afford it. However, it is inconvenient to get the gradients of the discrete value function. Bendsøe et al. proposed a solution to this difficulty: interpolation and penalization [6].

First, introduce a continuous density ρ_e to connect the specific parameters of two materials:

$$0 \leq \rho_e \leq 1,$$

and then add penalization on intermediate density to let the optimized object converge to discrete structure, because the intermediate density has no practical meaning in physics. For example, one simple and successful linear interpolation method can be defined as [7]:

$$A_e = A_1 + \rho_e(A_2 - A_1),$$
$$B_e = B_1 + \rho_e(B_2 - B_1),$$

where A_1, A_2, B_1, and B_2 are the specific values of certain properties (e.g., the index) of two materials (e.g., dielectric and air). During the optimization process, the continuous variables can cause intermediate values in the design, which may bring challenges to fabrication. Therefore, it is necessary to give the algorithm a "taste" to prefer a structure with only 0–1 values, which is achieved by adding penalization to intermediate values. One common penalization scheme is solid isotropic material with penalization (SIMP) [6].

To ensure the optimization run, we also need to know the gradient of each variable in the current structure, which is a process called sensitivity analysis. With a significant number of variables defined on finite elements, it is impossible to calculate the gradient of each variable one by one. One commonly used method is the adjoint method [8]. The adjoint method uses an adjoint system deduced from the forward system to help calculate the gradients of each variable, which will be discussed in detail in Section 2.3. It should be pointed out that the adjoint method can also be used in other optimization schemes, including the level set method, objective-first method, and other gradient-based algorithms. Fig. 1 shows the flow chart of the variable density method.

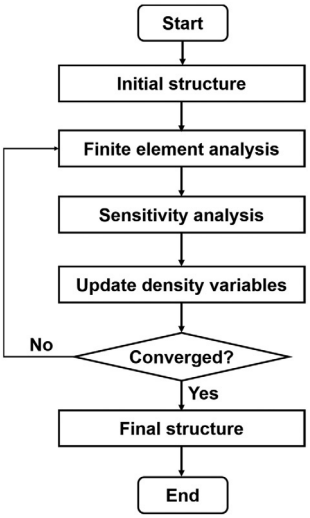


Fig. 1 Flow chart of variable density method.

Applications

Variable density methods have been used to optimize nanophotonic structures for years and many works have been reported, including different kinds of nanophotonic devices [9–17] and photonic crystals [18,19]. Below, these works will be introduced to help readers understand the application of the variable density method in nanophotonic design.

The simplest nanophotonic device could be photonic crystal waveguides and scattering loss usually occurs on the sharp corners, which limits its application in photonic crystal circuits. To solve this problem, Jensen et al. applied variable density method to design a 90 degree waveguide bend [9]. The optimization ideology is direct: select several unit cells near the photonic crystal waveguide bends and maximize the transmission from the input port to the output port. The same research group also designed photonic waveguides with 60 and 120 degrees bends based on their methods [10,11].

Waveguide can be used not only to guide light but also to slow light. This kind of waveguide is called a slow light waveguide which can enhance the interaction of the localized light modes. Wang et al. shows that this kind of waveguide can also be optimized with density-based topology optimization [12]. By using their methods, the authors demonstrated two waveguides facilitating slow light with group indexes of $n_g = 25$ and $n_g = 100$ and bandwidths of $\Delta\omega/\omega = 2.3\%$ and 0.3%, respectively. They also designed a novel waveguide with two different constant group index waveguide regions [12].

Apart from slow light waveguides, there is another kind of nanostructure used to localize light, photonic crystal cavity, which can also be optimized by using variable

density method. One aim of designing a light cavity is to seek stronger light intensity in a smaller volume, which is quantified using the figure of merit defined as Q/V, the ratio of the quality factor to the mode volume. By adopting density-based topology optimization, Wang et al. increased the Q/V ratio by up to two orders of magnitude relative to standard L1 cavities [13].

Devices with more complicated functions can also be designed with density-based topology optimization. Frandsen et al. designed a mode converter that can convert the fundamental even mode into the higher order odd mode of a photonic crystal waveguide through dispersion engineering [14]. The optimization area includes the conversion area around the center and the interface of the output waveguide. The footprint of the mode convertor is only $\sim 6.3\,\mu m \times \sim 3.6\,\mu m$. Frellsen et al. designed a mode multiplexer that could achieve low loss and broad bandwidth at the same time [15]. The loss is less than 1.2 dB in a bandwidth of 100 nm measured around 1570 nm. The footprint of this device is only $2.6\,\mu m \times 4.22\,\mu m$. These two devices are both designed using a density-based method combined with finite difference time domain (FDTD) method.

Works introduced above are limited to 2D devices. Density-based topology optimization can also be used for spatial light processing devices which have broad applications in images and sensors. Camayd-Muñoz et al. designed a spatial wavelength-polarization router using density-based topology optimization [16]. With their method, the designed wavelength-polarization splitting element can separate light with different wavelength-polarization information to different areas of the target plane. This designed structure is freeform in three dimensions and has a small volume of $2\,\mu m^3$. In addition to linear devices, density-based topology optimization can also be applied in nonlinear nanophotonic devices. Elesin et al. proposed a 1D photonic switch that is based on Kerr effect of nonlinearity [17]. The designed switch has no periodic structure but has a better performance than normal simple designs in the literatures before this work.

In recent years, topological photonics has become an attractive research field because of its novel physical phenomena. Traditional design methods of topological photonic crystal are based on intuition combined with trial-and-error. Christiansen et al. presented a scheme to design topological insulators with quantum spin Hall states using density-based topology optimization [18]. They constructed an intersection with unit cells to be optimized. Also, injected light from one port and optimized the transmission of the other three different ports. In this way, the structure has a significant possibility to converge to topological photonic crystal. Topological photonic crystal has been introduced in nanophotonic devices design because of its robustness. Yuan et al. applied density-based topology optimization in their scheme of topological wavelength router, which could guide light with different wavelengths to different topological channels [19]. They realized an on-chip topological nanophotonic wavelength router using topology optimization for the first time. The optimization results are shown in Fig. 2. The central optimization area is only $2.30\,\mu m \times 3.78\,\mu m$ and can guide light with central wavelengths

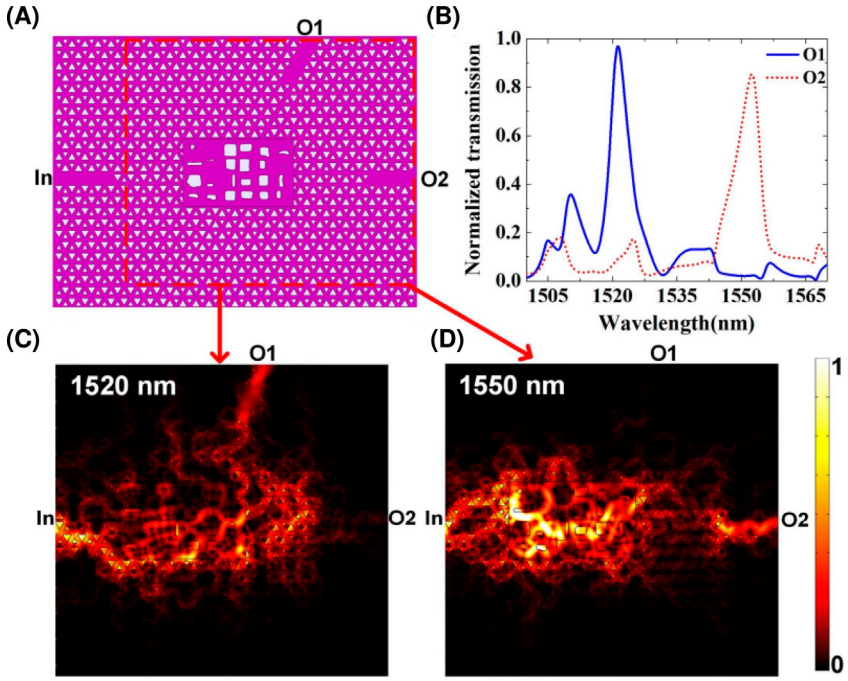


Fig. 2 Optimization results of on-chip topological nanophotonic wavelength router. (A) The designed structure; (B) the simulated spectrum; and (C and D) the simulated time-average power flow distributions at two target wavelengths.

of 1520 and 1550 nm to different topological channels, respectively. The structure is shown in Fig. 2A. Purple color represents silicon and white color represents air. The normalized transmission is shown in Fig. 2B. The signal-to-noise ratios of the two peaks are 11.20 and 15.76 dB, respectively. Fig. 2C and D shows the simulated time–average power flow distribution at 1520 and 1550 nm near the central optimized area.

The variable density method has been applied in many areas in nanophotonic design, which proves its generality. Variable density method regards each discrete element as a design variable and is convenient to integrate with other calculation methods like FEM and FDTD, which accelerates the design in nanophotonics.

2.1.2 Level set method

Introduction

Different from the variable density method, the level set method regards boundaries as the optimization object. The traditional parameterized method deals with structure boundary with coordinates of points on the boundary; however, this kind of method is inconvenient in situations where there are boundary's merging and splitting. How to overcome this difficulty? Level set method answers this question with a level set

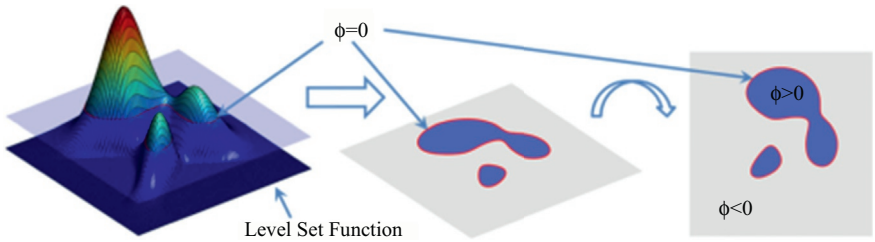


Fig. 3 Level set function representation of plane geometry.

function which is an artificial function defined on the design domain with one dimension higher than the actual structure. For example, if the design domain is a 2D area, the level set function can be a 3D surface, as shown in Fig. 3 [20].

The level set function has different specific forms, but the main idea is the same and clear: use a continuous function to represent the boundary implicitly. The general level set function ϕ can be expressed as follows:

$$\phi = \begin{cases} < 0 & \text{if in material 1} \\ > 0 & \text{if in material 2.} \end{cases}$$

The level set method does not deal with the structure boundary directly and the level set function evolves according to the level set equation [21]:

$$\phi_t + \vec{V} \bullet \nabla \phi = 0,$$

where ϕ_t is the derivative of ϕ to time t. $\vec{V}$ is the velocity. Here, time and velocity are two variables controlling the evolution of the level set function. Time is similar to step length, which should be decided by testing according to a specific optimization problem. Velocity is related to gradients and should be deduced from sensitivity analysis. The sensitivity analysis method commonly used by the level set method is the adjoint method. Fig. 4 shows the flow chart of the level set method.

Applications

Because of its flexible geometric shapes, level set method is broadly applied in nanophotonic structure optimization problems, including photonic crystals and nanophotonic devices. Kao et al. presented a work on maximizing the band gaps of two-dimensional photonic crystals [22]. They used the level set method to describe the interface between two materials with different dielectric constants. The interface is updated with generalized gradient ascent method. Their method shows that the band gap between two arbitrary adjacent bands can be enlarged. The optimized largest bandgap of GaAs in air is 0.4418 ($\omega a/2\pi c$) for the transverse magnetic (TM) mode and 0.2104 ($\omega a/2\pi c$) for the transverse electric (TE) mode.

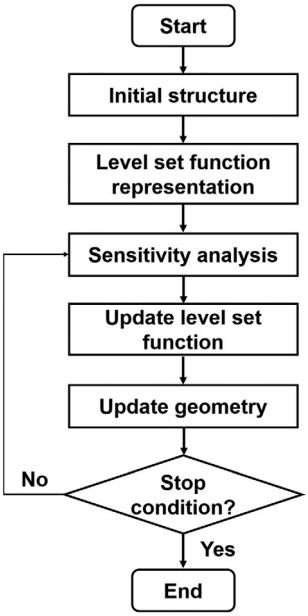


Fig. 4 Flow chart of level set method.

In addition to band gap optimization, level set method has also been introduced into the configuration optimization of nanophotonic devices. Lalau-Keraly et al. combined adjoint method with level set method to present an optimization scheme for electromagnetic design [23]. As an example, a Y-splitter is designed using their method. Because the adjoint method only needs two times of simulation to get the gradients of all variables, the proposed method has an obvious advantage in optimization efficiency. In their work, the authors also gave an interpretation of adjoint method from the perspective of physics.

Furthermore, level set method is adopted in other optimization schemes by adding constraints on designed structures. Based on their previous works [24–26], Piggot et al. added a level set method in their proposed objective-first method, to impose curvature constraints on device boundaries [27]. Through their methods, they achieved devices suitable for fabrication, including a three-channel power splitter, spatial mode demultiplexer, wavelength demultiplexer, and directional coupler. All of the structures consist of a fully etched 220-nm thick silicon (Si) layer with a silicon dioxide (SiO_2) substrate. In recent years, level set method has also been adopted in robust optimization of nanophotonic devices. Lebbe et al. combined Hadamard's boundary variation method with level set method [28]. With their method, they designed different kinds of nanophotonic devices including crossing device, mode convertor, power divider, and duplexer, which shows their method's generality. Moreover, they studied the robustness of the optimized devices concerning physical perturbation and geometrical uncertainties.

The level set method is suitable to handle complex boundary changes and is flexible to be integrated with other optimization algorithms. With level set method, it is convenient to add curvature limits to the optimized structure, which is beneficial to fabricate and enrich the function.

2.1.3 Bi-directional evolutionary structural optimization

Introduction

There is another kind of topology optimization method, bi-directional evolutionary structural optimization (BESO), which is developed from evolutionary structural optimization (ESO) method. The main idea of ESO is removing inefficient material elements slowly from the current structure, which dates back to the works of Xie and Steven in the early 1990s [29]. However, as Huang et al. have pointed out [30], this strategy has a main disadvantage that an element may be moved prematurely, leaving a "hole" which can be nonoptimal to the final structure. To overcome this disadvantage, BESO was proposed by Huang et al., which adds and removes elements at the same time during the evolution process [31]. This method deals directly with finite elements and is convenient to integrate with finite element analysis. Fig. 5 shows the flow chart of BESO.

Applications

The applications of BESO mainly concentrate on optimization of photonic crystals, such as bandgap optimization [32,33], design of photonic crystals with negative refraction [34], and inverse design of topological photonic crystals [35–38].

Maximizing the band gap is important in the design of photonic crystals. Meng et al. developed a new algorithm based on BESO to maximize the bandgap of photonic crystals [32]. At each finite element, they use discrete intermediate values instead of 0–1 values to

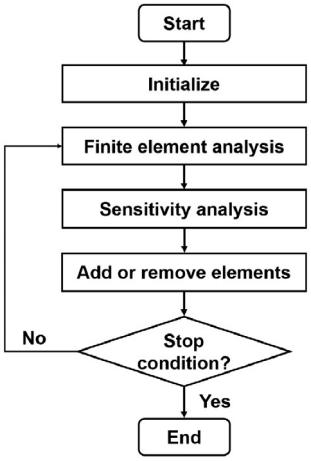


Fig. 5 Flow chart of bi-directional evolutionary structural optimization.

avoid extremely fine mesh. The variation of the design variable in each iteration is limited to 0.1. Because the intermediate values will reduce the refractive index contrast between materials, it is natural for the design to converge to a 0–1 structure. The authors show that using their method, the bandgap between two arbitrary adjacent bands can be opened and maximized. Furthermore, Chen et al. achieved complete bandgap optimization for TE and TM modes with BESO [33]. The mechanism of negative refraction for photonic crystals arises from the band structure of the periodic medium. Therefore, the design of photonic crystals with negative refraction can be optimized from the perspective of band engineering. With the definition of upper limit and lower limit for all-angle negative refraction (AANR) frequency range, Meng et al. used a modified objective function to expand the frequency range [34]. The optimization results show that optimized structures have a larger AANR frequency range than the designs based on intuition.

Apart from traditional photonic crystals, BESO is also applied to the optimization of topological photonic crystal (PC). In most recent years, Chen et al. developed an optimization approach based on BESO to design topological photonic crystals with wide bandgaps [35]. Their proposed method can construct doubly degenerate Dirac cones of PCs at any wanted frequency. They then optimized the bandgap overlap of topologically trivial and nontrivial PCs to achieve topological PCs with broad bands. Furthermore, they use their optimization approach to design higher order photonic topological insulators [36]. Their optimization results show that the designed structure can support gapped edge states and robust corner states at the same time. The group index of the proposed edge state is large. With the designed topological photonic crystal, they demonstrate four-channel photonic routers for edge states and three-channel routers for corner states. In addition, they show that their method can also design second-order topological photonic crystals with C_3-symmetry [37]. Most works on topological photonic crystals are for the TE mode or TM mode, and topological crystals simultaneously supporting two modes are rare. Chen et al. used BESO-based optimization approach to propose a structure which supports two kinds of modes [38]. They first used their optimization to design a photonic crystal with a complete bandgap for both TE and TM modes, and then used simple tight binding model to study the topological features of the structure.

BESO method is simple in concept, and reported works show that it can be applied in many areas in nanophotonic design, especially photonic crystal design. Structural design with this method usually has novel properties.

2.2 Objective-first algorithm

2.2.1 Introduction

The objective-first algorithm was first proposed by Lu et al. in 2013 and has been applied in many areas in nanophotonic design since then. Here, we first briefly introduce the main idea of objective-first algorithm [39] and review its applications in nanophotonic design.

Typical optimization (e.g., minimization) problems can be expressed as follows:

$$(\text{decrease})\, f(x)$$
$$\text{subject to}\, g(x,p) = 0,$$

where x is the field variable (e.g., electric field), p is the structure variable (e.g., the distribution of dielectric constant), $f(x)$ is the design object, and $g(x,p)$ is the governing equation of the system. $f(x)$ can be any function that can describe the performance of the device and $g(x,p)$ is usually an electromagnetic wave equation in nanophotonic design.

Different from the traditional optimization idea above, the objective-first algorithm expresses the optimization problem below:

$$(\text{decrease})\ \|g(x,p)\|^2$$
$$\text{subject to}\, f(x) = 0,$$

where $\|g(x,p)\|^2$ is called physics residual. From the expressions above, we can see that the former optimization objective is set as a hard constraint which is always satisfied during the optimization process. Meanwhile, the former governing equation is optimized to fit the constraint. Because the objective is given higher priority than the governing equation, this algorithm is named as objective-first algorithm. This is a strategy to chase solutions with high performance and high efficiency in convergence, at the expense of not perfectly satisfying the governing equation.

2.2.2 Applications

In recent years, there have been many works based on objective-first algorithms, which are primarily concentrated in on-chip nanophotonic devices, ranging from two-channel and three-channel wavelength multiplexers [25,40], spectrometer [40], optical diode [41], and metalens on an optical fiber tip [42].

One typical work is about an on-chip wavelength demultiplexer, proposed by Piggot et al. [25]. The designed wavelength demultiplexer splits two channels with central wavelength of 1300 and 1550 nm light from the input waveguide to two different output waveguides, respectively. The device has a low insertion loss (~2 dB), low cross talk (<−11 dB), and wide bandwidths (>100 nm). The pattern is freeform and the footprint is only $2.8 \times 2.8\,\mu m^2$. Fig. 6A and B is the top–down view and angled view of scanning electron microscopy (SEM) images. The simulated and measured scattering parameters (S-parameters) are shown in Fig. 6C and D. The design process can be divided into three different stages. The first stage is the optimization stage allowing permittivity ε to change continuously. In the second stage, the structure is converted into a binary level set presentation and optimized at two target wavelengths. The final stage is broadband optimization by specifying the device performance at 10 different wavelengths, with five frequencies equally spaced about each central frequency.

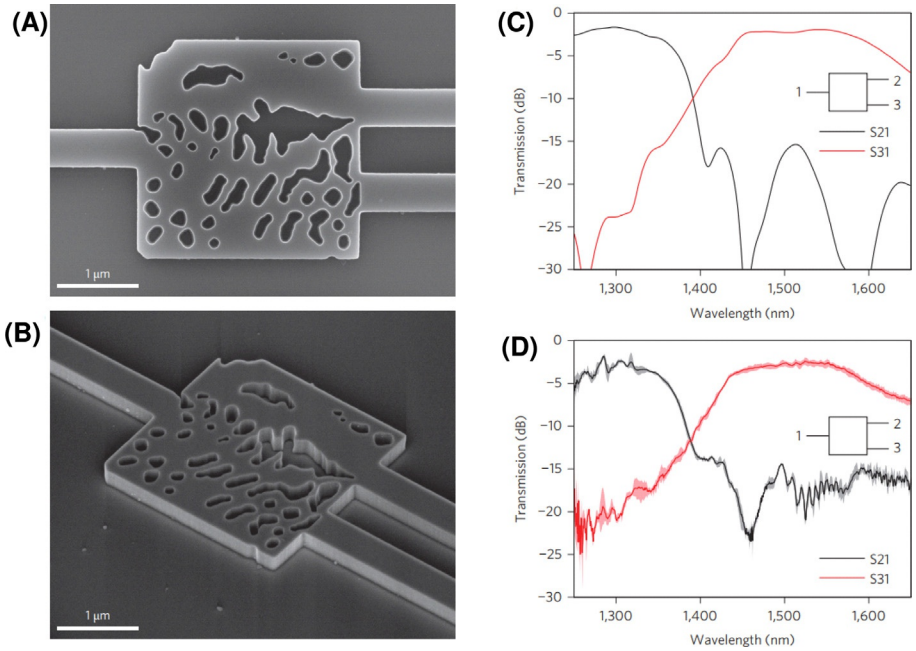


Fig. 6 (A and B) Top–down view and angled view of SEM images; (C and D) simulated and measured S-parameters of the designed final structure.

Furthermore, Su et al. improved the objective-first algorithm used above and designed a narrow-band three-channel wavelength demultiplexer [43]. The interval of three channels (1500, 1540, and 1580 nm) is only 40 nm. The footprint of the device is 24.75 μm^2. Hadibrata et al. also designed an eight-channel spectrometer working in the near-infrared regime using an objective-first algorithm [40]. The spectral resolution was 0.25 nm, with a bandwidth of 30 nm. The footprint of the device is $30 \times 12.8\,\mu m^2$. Apart from wavelength routers and spectrometers, an optical diode was also designed with the objective-first algorithm. An optical diode is a two-port device that allows light to propagate from the first port to the second port but prohibits propagation in the other direction. Callewaert et al. proposed an on-chip broadband optical diode with small size [41]. The silicon-based nanophotonic device is suitable for fabrication and integration.

The devices mentioned above are two-dimensional on-chip devices; however, the objective-first algorithm can also be used on spatial light processing devices, such as metalenses. Hadibrata et al. proposed a thin circular grating-like metalens which could transform a parallel wavefront into a spherical wavefront at the near-infrared range [42]. The designed structure was fabricated using a 3D printing method and measured. The focal length is about 8 μm and size of the focal spot is at the scale of 100 nm.

The objective-first algorithm is an efficient algorithm, which searches for high-performance solutions even at the expense of not fulfilling the governing equations. This algorithm has been applied in the design of numerous nanophotonic devices and has been proven to have good generality.

2.3 Other gradient-based algorithms

2.3.1 Introduction

With the development of algorithms, more and more optimization schemes in nanophotonic design based on gradients have been developed. In this section, we will introduce other algorithms based on gradient calculation but different from the ones introduced above. Although these algorithms cannot be classified into algorithm types mentioned directly, they still have a common feature, the calculation method of gradients.

Because of the large quantity of design variables in structure optimization, the traditional gradient calculation method using brute force is inefficient. The standard choice of gradient calculation is the adjoint method or adjoint variable method [8]. Below, we introduce the main idea of the adjoint method. Consider a system described below:

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{P}(\mathbf{x}). \tag{1}$$

Here, $\mathbf{x}$ is the design variable vector and $\mathbf{u}(\mathbf{x})$ is the response (e.g., electromagnetic field). $\mathbf{K}(\mathbf{x})$ and $\mathbf{P}(\mathbf{x})$ are the operators. We want to calculate the gradients of the following target function:

$$F(\mathbf{x}) = G(\mathbf{u}(\mathbf{x}), \mathbf{x}) \tag{2}$$

The calculation of gradients or sensitivity analysis computes the derivative of $F(\mathbf{x})$:

$$\frac{DF}{D\mathbf{x}}(\mathbf{x}) = \frac{DG}{D\mathbf{u}}(\mathbf{u}(\mathbf{x}), \mathbf{x})\frac{D\mathbf{u}}{D\mathbf{x}}(\mathbf{x}) + \frac{DG}{D\mathbf{x}}(\mathbf{u}(\mathbf{x}), \mathbf{x}) \tag{3}$$

The difficulty of calculating the expression above is that $D\mathbf{u}/D\mathbf{x}$ is implicitly defined in Eq. (1). The adjoint method handles this difficulty by eliminating $D\mathbf{u}/D\mathbf{x}$ in Eq. (3). This is achieved by introducing Lagrange multiplier λ to define the augmented function:

$$\widehat{F}(\mathbf{x}) = G(\mathbf{u}(\mathbf{x}), \mathbf{x}) - \lambda\bullet(\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) - \mathbf{P}(\mathbf{x})) \tag{4}$$

Differentiation of the above provides the following:

$$\begin{aligned}\frac{D\widehat{F}}{Dx_i}(\mathbf{x}) &= \frac{\partial G}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}), \mathbf{x})\frac{\partial \mathbf{u}}{\partial x_i}(\mathbf{x}) + \frac{\partial G}{\partial x_i}(\mathbf{u}(\mathbf{x}), \mathbf{x}) \\ &-\lambda\bullet\left(\frac{D\mathbf{K}}{Dx_i}(\mathbf{x})\mathbf{u}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\frac{D\mathbf{u}}{Dx_i}(\mathbf{x}) - \frac{D\mathbf{P}}{Dx_i}(\mathbf{x})\right)\end{aligned} \tag{5}$$

Rearranging Eq. (5), we have

$$\frac{D\widehat{F}}{Dx_i}(\mathbf{x}) = \left[\frac{\partial G}{\partial x_i}(\mathbf{u}(\mathbf{x}),\mathbf{x}) - \lambda\bullet\left(\frac{D\mathbf{K}}{Dx_i}(\mathbf{x})\mathbf{u}(\mathbf{x}) - \frac{D\mathbf{P}}{Dx_i}(\mathbf{x})\right)\right] + \frac{D\mathbf{u}}{Dx_i}(\mathbf{x})\bullet\left[\frac{\partial G}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}),\mathbf{x}) - \mathbf{K}^{\mathbf{T}}(\mathbf{x})\lambda\right], \tag{6}$$

where $(\,)^T$ denotes the transpose operator. Since λ is arbitrary, we can select an appropriate λ to let the coefficient of the $(D\mathbf{u}/Dx_i)(\mathbf{x})$ term be zero. In this way, we can get the derivatives from the first bracketed quantity in Eq. (6). We need to solve the following adjoint problem

$$\mathbf{K}^{\mathbf{T}}(\mathbf{x})\lambda(\mathbf{x}) = \frac{\partial G}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}),\mathbf{x}) \tag{7}$$

to get the adjoint response $\lambda(\mathbf{x})$. Now, λ is expressed as a function of the design variable vector and is no longer arbitrary. Once the $\lambda(\mathbf{x})$ is determined from Eq. (7), we can substitute $\lambda(\mathbf{x})$ into Eq. (6) to get derivative expression below:

$$\frac{D\widehat{F}}{Dx_i}(\mathbf{x}) = \left[\frac{\partial G}{\partial x_i}(\mathbf{u}(\mathbf{x}),\mathbf{x}) - \lambda(\mathbf{x})\bullet\left(\frac{D\mathbf{K}}{Dx_i}(\mathbf{x})\mathbf{u}(\mathbf{x}) - \frac{D\mathbf{P}}{Dx_i}(\mathbf{x})\right)\right] \tag{8}$$

When comparing Eqs. (1) and (7), we see that these two equations have the same structures. In many cases where the systems are symmetric, we have $\mathbf{K} = \mathbf{K}^{\mathbf{T}}$. Eq. (1) is called forward problem and Eq. (7) is called adjoint problem. In the adjoint method, each time we need to calculate the derivatives of the target function to the design variable vector, we need to solve only two equations which are usually two times of simulation in nanophotonic design. Compared to calculating derivatives to variables ono by one, the advantage of the adjoint method's efficiency is obvious because this method does not consider the number of design variables. Adjoint method can also be understood from the perspective of physics [23].

The adjoint method is widely used in many areas and is usually integrated with other optimization frameworks, including previously introduced algorithms. Most gradient-based optimization algorithms in nanophotonic design are gradient-based because the design variable number is usually vast. Next, we will introduce some typical gradient-based works in nanophotonic design, which are adjoint-based but are different from algorithms introduced in the former sections. For simplicity, we use adjoint-based algorithms to denote optimization schemes introduced below.

2.3.2 Applications

Adjoint-based algorithms have been used in various nanophotonic designs, such as linear devices [44], nonlinear devices [45], active devices [46], and metasurfaces [47–52].

Sideris et al. presented an adjoint-based algorithm used to design linear nanophotonic devices, which combined the adjoint method with the boundary integral equation method

[44]. In their method, integral equation method is used to solve Maxwell equation problems, which is convenient for dealing with discrete boundaries. With this method, the authors designed a nonadiabatic waveguide taper, a 1:2 power splitter and a perfectly vertical incidence grating coupler.

A nonlinear device designed with an adjoint-based algorithm was first presented by Fan's group [45]. They first extended the adjoint method to a nonlinear system and then applied it to Kerr effect nonlinearity. With this method, they designed an optical switch which can guide light to different ports when the light is in linear region and nonlinear region. The device is made of air and chalcogenide glass (Al_2S_3). In linear region, the device has a power transmission of 81.8% and 5.9% to the linear output port and nonlinear port, respectively, while in nonlinear region, the transmission values become 6.1% and 80.8%. Fan's group also presented an adjoint-based algorithm for active nanophotonic devices, which combined the adjoint method using multi-frequency finite-difference frequency-domain method [46]. As a proof of principle, they designed a nonmagnetic dynamic optical isolator, which reduced the length of the modulated regions by a factor of two while achieving good performance in isolation ratio and insertion loss.

Adjoint-based algorithms are also used in metagratings and metasurfaces. Fan's group designed metagratings using adjoint-based topology optimization combined with the rigorous coupled-wave analysis (RCWA) solver [47]. With their methods, the designed metagratings which have freeform patterns can deflect light to angles as large as 80 degree with high efficiency. Using their proposed optimization method, Fan's group also achieved a large-area, high-numerical-aperture silicon metasurface lenses with focusing efficiencies over 90% [48]. Further, they combined their adjoint-based topology optimization with generative neural network to achieve a new platform for global population optimization of metasurfaces [49,50]. Mansouree et al. applied adjoint optimization to 2.5D metastructure design, which are stacked layers of interacting metasurface layers [51]. They fabricated and measured their metastructures and found that the performance of their design has significantly higher efficiencies than metasurfaces designed by a conventional simplified approach. In addition, Mansouree et al. designed a large-scale parametrized metasurface using an adjoint-based method [52]. The metasurface is made of many parameterized meta-atoms and is easy to fabricate.

The adjoint method is highly efficient in derivative calculation and widely used in many optimization schemes. The adjoint-based algorithms are still developing and more and more works based on adjoint-based algorithms will be reported.

3. Heuristic algorithms

As introduced above, optimization algorithms are used to search for a "better" solution through an iterative process. The key principle to generate a new solution, is what we used to identify a certain algorithm and classify it. In contrast to gradient-based

algorithms, heuristic algorithms, or gradient-independent algorithms, generate new solutions following various strategies without mathematical analysis for gradient information. Containing randomness or not, most of the heuristic algorithms are embedded within the thoughts of Monte-Carlo method, with whose validity being promised by the law of large numbers.

In the following sections, we will first introduce individual-based algorithms along with their applications. Simple algorithms like hill-climbing algorithm and DBS algorithm search the neighborhood the current solution, and update it when finding a better one. Complicated algorithms like TS incorporate stochasticity and short-term memory to jump out of local minimum and avoid cycling back.

Population-based algorithms resemble the analogy of parallel individual-based algorithms, but information from the whole population also controls the searching direction. Here, genetic algorithm, differential evolution algorithm, particle swarm algorithm, and ant colony algorithm will be introduced in order.

3.1 Hill-climbing algorithm

3.1.1 Introduction

Hill-climbing algorithm is a locally optimized method that uses feedback information to help generate solutions. The algorithm simulates the process of climbing a mountain, randomly selects an initial location to climb the mountain, and moves to a higher direction each time until it reaches the top of the mountain. As shown in Fig. 7A [4], it starts from the current node and compares with the values of neighboring nodes. If the current node is the best, then the current node is taken as the maximum value (the highest point of the mountain); otherwise, the highest neighbor node is used to replace the current node in order to achieve the purpose of mountain climbing. This iteration is repeated until it reaches the highest point. The optimal solution in the adjacent space is selected as the current solution each time until an optimal solution with relatively good performance appears.

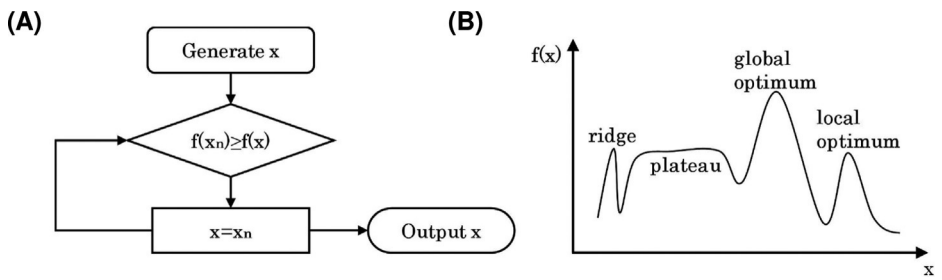


Fig. 7 (A) Flowchart of the hill-climbing algorithm and (B) an example of the target function in which the difficulties of hill climbing are shown.

The advantage of the hill-climbing algorithm is that it does not require a traversal process to reach the highest point of the solution space; it selects nodes with higher values through a heuristic method, which greatly improves efficiency. The process does not need to memorize the previous steps, which makes it memory-saving when searching for optimal solutions in large parameter space. However, the hill-climbing algorithm also has several disadvantages. As is shown in Fig. 7B [4], when a node is higher than any of its neighbors, but not the highest point of the whole function, this node will be regarded as the optimal result, which makes us sometimes unable to get satisfactory results in the calculation. In addition, this algorithm is easily limited by certain function shapes. When encountering a "plateau" (the function value remains unchanged in a certain range) during the searching process, the search direction cannot be determined, and will move randomly, which reduces the efficiency. When encountering a "ridge" (the function is steep near the maximum), it may oscillate repeatedly around the optimal solution, and the forward speed will slow down.

3.1.2 Applications

The hill-climbing algorithm is a relatively basic algorithm that is easy to get started with. Using a hill-climbing algorithm can further improve performance of an initial structure.

Lin et al. designed a one-dimensional photonic crystal split beam cavity based on the deterministic method and hill-climbing algorithm [53]. The initial structure is a one-dimensional photonic crystal, and the filling fractions are quadratically tapered. This structure has been proved to be effective in a previous work [54]. The radius of the first three holes in the center greatly affects the optical performance of the split beam cavity, so the hill-climbing algorithm is used to fine-tune the radius of the three holes in the central neighborhood to obtain a higher resonance mode quality factor. As a result, the quality factor of the second-order TE mode is improved to 1.99×10^4.

The optimization of nanophotonic devices is often a complex problem. It is necessary to find the optimal solution in the full parameter space, and the form of the objective function is often complicated. Therefore, hill climbing may not be an excellent method for designing nanophotonic devices with complicated functionalities. However, when the initial structure has been proved to possess an effective function, using hill climbing can further improve the performance of the device.

3.2 Direct binary search

3.2.1 Introduction

The following section describes the origin and main flow of DBS algorithm. Ref. [55] presents an iterative binary search algorithm called direct binary search (DBS) algorithm in the synthesis of digital holograms. The DBS algorithm in the synthesis of digital holograms could manipulate the hologram transmittance directly to produce the best reconstruction, which is illustrated in the flow chart of Fig. 8 [56].

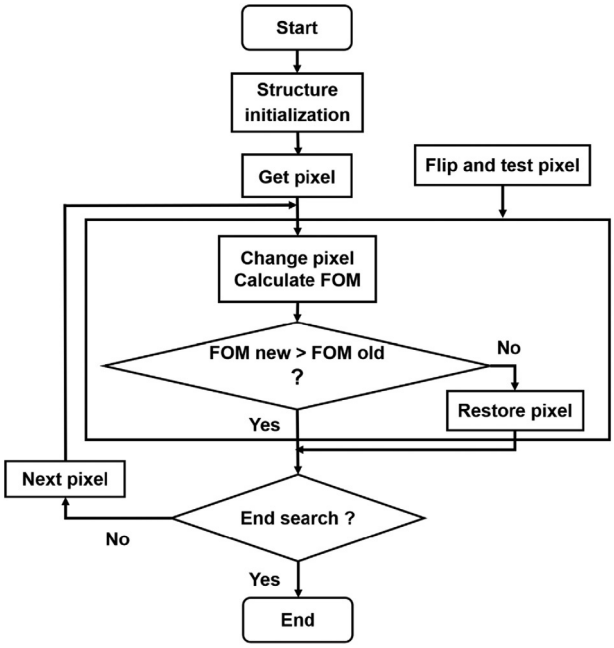


Fig. 8 Flow chart of direct binary search algorithm.

The main flow of DBS algorithm is as follows. The designed space of devices is discretized into grids called "pixels" first. Each pixel has two states: 0 or 1, corresponding to two materials occupying the pixel. For an initial structure, randomly select a pixel and switch its state, and then a figure-of-merit (FOM) is calculated. The FOM generally refers to objective function. If the FOM is promoted, the perturbation is retained for the pixel. If not, the perturbation is reversed. After the evaluation of FOM, algorithm proceeds to the next pixel. An iteration contains traversing all pixels in a certain order. The iteration continues until the FOM converges to a stable value. An upper bound on the total number of iterations and a minimum change of FOM are defined to enforce numerical converges [57–61]. For special samples, the DBS algorithm is slightly different, but the overall steps are described above basically.

Judging from the descriptions above, it is obvious that DBS algorithm happens to be a particular case of hill-climbing algorithm where the solution space is composed of binary variables. In both algorithms, improved solutions are accepted and worse solutions are refused. The neighbor nodes in a binary integer solution space can be defined like DBS. By simply flipping every pixel in order, one can make a traversal within all the neighbor nodes.

3.2.2 Applications

The DBS algorithm is a relatively simple iterative search algorithm. With the development of intelligent algorithms and nanophotonics, some scholars have introduced DBS

algorithm into the design of nanophotonic devices. The algorithm provides an effective approach to designing nanophotonic device, including the design of diffractive optics [62,63], nanophotonics for light trapping [58], couplers [64], computational microscopy [65], free-space polarizers [66], polarization beam splitter [57], the integrated cloak [67], mode router [68,69], sharp adiabatic bends [70], and power splitters [71]. Several typical devices designed by the DBS algorithm are introduced below.

Using DBS algorithm, Shen et al. have carried out rich works of nanophotonic devices. They designed a free-space-to-waveguide coupler [64], polarization beam splitter [57], and integrated cloak [67], which are shown in Fig. 9. Fig. 9A shows a free-space to multimode waveguide coupler and polarization splitter, where label a is the structure diagram and labels b and c are simulated time-averaged intensity distribution for light

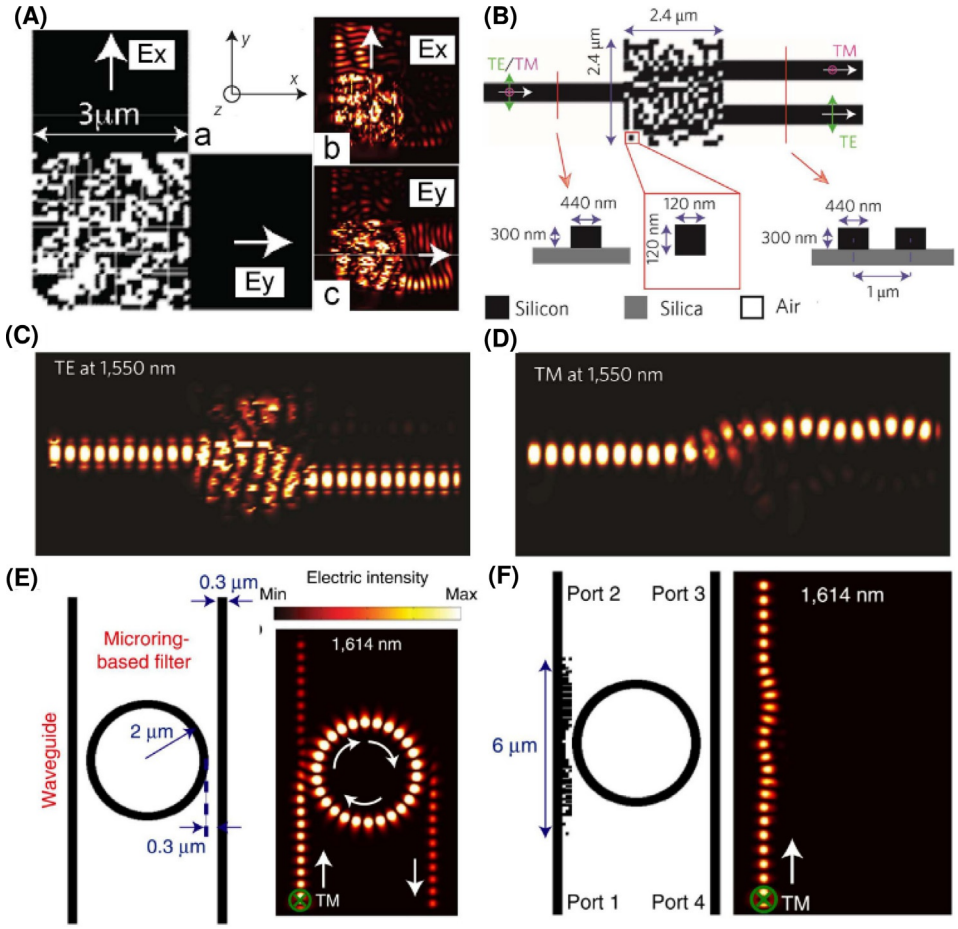


Fig. 9 Nanophotonic devices designed by direct binary search algorithm. (A) A free-space to multimode waveguide coupler and polarization splitter; (B) a designed polarization beam splitter; (C and D) the simulated steady-state intensity distributions for TE and TM polarized light; and (E and F) demonstration of a nanophotonic cloak for micro-ring resonator.

polarized along *X*-axis and that polarized along *Y*-axis, respectively [64]. For the design of polarization beam splitter, they used the concept of free-form metamaterials. By freely optimizing the geometry of the metamaterial, devices can be highly functional and occupy a small foot-print. Moreover, nanopatterning enables one to engineer the refractive index in space at a deep sub-wavelength scale. In this way, devices that achieve high-efficiency mode conversion in an extremely small area become feasible. They also designed a polarization beam splitter with a footprint of $2.4\,\mu m \times 2.4\,\mu m$ in the above way, which is shown in Fig. 9B, and the simulated steady-state intensity distributions for TE and TM polarized light at the design wavelength of 1550 nm are shown in Fig. 9C and D, respectively [57].

With the development of photonic integrated circuits, the integration density needs to be improved. In addition to directly reducing the device size, one of the options is to decrease the spacing between individual devices. The DBS algorithm is applied to design the integrated cloak with a footprint of just a few micrometers to decrease this spacing without considerably increasing crosstalk [67]. In most of the applications, light is coupled into the resonator via a waveguide that is placed in close vicinity to the ring. However, if another waveguide is placed close to the micro-ring, the two optical components would work as a coupled system with functionality that is different from that of either one working independently, as shown in Fig. 9E. Shen et al. designed a nanophotonic cloak that allows a wave guide to be placed at a gap of only 300 nm from the micro-ring and essentially renders the waveguide invisible to the micro-ring. The structure diagram of the device and the steady-state intensity distribution are shown in Fig. 9F.

Another option to increase integration density is to combine the function of multiple devices into a single compact device. Liu et al. designed a mode-division multiplexing circuit consisting of a multiplexer, a crossing, and a demultiplexer, where the DBS algorithm was used to optimize the structure of nanohole distribution [69]. The distributions of silicon and air pixels determined by DBS algorithms are suitable for fabrication. Moreover, Han et al. theoretically designed three 1×2 power splitters based on photonic crystal-like metamaterial structure with the DBS algorithm [71]. These nanophotonic devices can be effectively assembled into arbitrary direction, multichannel and ultracompact power splitters to satisfy the different requirements.

For the design of nanophotonic devices, DBS algorithm is suitable for relatively smaller parameter space and generates discrete structure, which is more favorable to the fabrication using traditional techniques like focused ion beam milling or electron beam lithography. However, as the search space increases, the possibility of falling into local optimal value will tend to increase.

3.3 Simulated annealing algorithm

3.3.1 Introduction

In this section, we introduce the simulated annealing algorithm (SAA) along with its application in designing nanophotonic devices. By allowing worse solutions according

to the Metropolis principle [72], this individual-based algorithm is armed with the ability to jump out of local optima. Like most heuristic algorithms, the incorporation of certain strategies also introduces parameters to be decided by experience, which we will discuss later.

Consider a crystal with perfect translational symmetry. All the atoms are periodically set in a certain spatial location, except for several defects and fluctuations. How does this beautiful periodic structure form in natural world? We are not going to discuss statistic physics here, but the following discussion would inevitably contain some basic physical agreements. Consider a pot of fully boiled atoms, all of which together form a high energy state of the system. When we cool down the temperature slowly enough, this pot of atom soup will find enough time to adjust itself to lower energy state. If the temperature no longer changes, the system should have the most stable state among all the possible states, which happens to be a solid crystal. Thus, the formation of beautifully periodic structure should be the result of a slow enough cooling process. More discussion can be found in thermodynamics.

The SAA [73] was invented by mimicking the above-introduced process. Getting back to the topic of heuristic algorithm,

$$\min \text{objective} = f(\boldsymbol{x}), \boldsymbol{x} = (x_1, x_2, x_3, \ldots)$$
$$\text{s.t.} g_i(\boldsymbol{x}) = 0, h_i(\boldsymbol{x}) \leq 0, i = 1, 2, \ldots$$

still the objective function and the constraints on the parameters to be optimized are predefined.

The SAA is identified by Metropolis principle [72], which guides the algorithm to converge and jump out of local minimum. In the frame of SAA, we introduce the temperature, T, as a controlling parameter over the Metropolis principle.

In a certain iterative process, when we have evaluated the current solution, let us say, $f(\boldsymbol{x_i})$, we can generate a new solution $\boldsymbol{x_{i+1}}$, by bringing stochastic disturbance into each component of the current solution. If the newly generated solution has a better function value, it should always be accepted, like the hill-climbing algorithm. If not, by Metropolis principle, a worse solution will still be accepted with the probability

$$P = e^{\frac{-f(\boldsymbol{x_{i+1}})+f(\boldsymbol{x_i})}{kT}} \tag{9}$$

where k is the Boltzmann constant in a narrow sense, which can be set to 1 for convenience. If not accepted, the new solution will be abandoned, and this iteration ends.

At the end of each iteration, the temperature shall be decreased by a certain ratio α,

$$T_{i+1} = \alpha T_i. \tag{10}$$

In the whole optimization process, the temperature T slowly changes from a high value at the beginning, to nearly zero in the end. From the Metropolis principle, we could see that a high temperature allows relatively bad solutions, compared with the

current one, while a low temperature means less tolerance over a worse solution. By slowly cooling down the temperature, the system would finally converge at a "low energy state," refusing all the worse solutions.

If the parameter k is set to be 1, there is only one parameter, the temperature, T to be decided by experience. If T is set too high at the beginning, any worsen disturbance of the current solution will be tolerated, resulting in invalid search and waste of time. On the contrary, if T is set too low at the beginning, the algorithm will degenerate into a local search, with only iteratively search for better solutions, losing its advantage from the Metropolis principle. Another parameter that we shall not miss is the cooling rate, which determines the whole running time of this optimization. To find a better solution, a small cooling rate is always recommended, but it also requires more computational cost. The flowchart of SAA is given in Fig. 10.

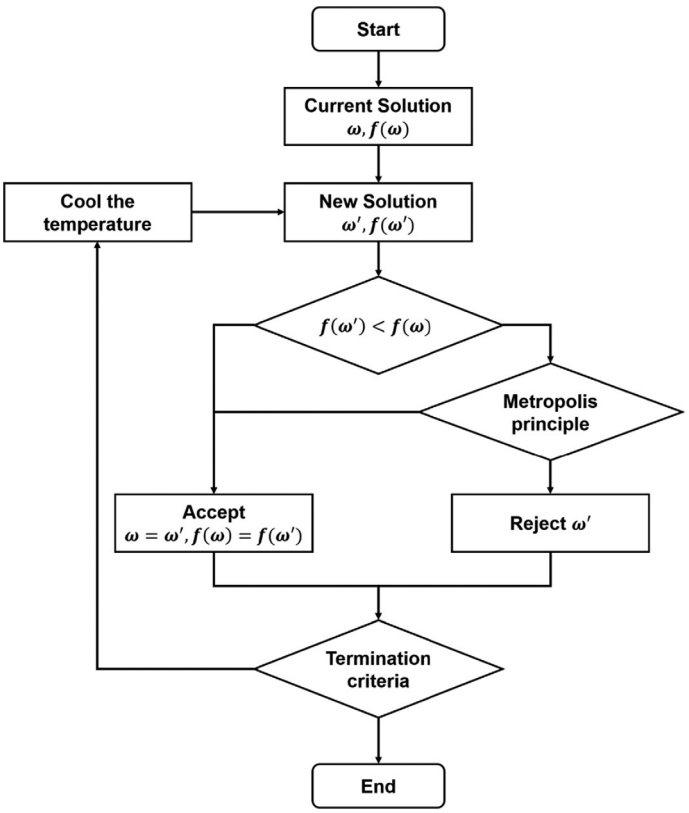


Fig. 10 Flow chart of simulated annealing algorithm.

3.3.2 Applications

Even when artificial intelligence has been utilized in the simulation and inverse design of nanophotonic devices [74,75], the SAA still works well by giving a general frame in which designing arbitrary devices can be incorporated. For the last decade and even earlier, the SAA has generated many works. We will go through a quick scan and give detailed discussion on one example, hoping that the readers can find it helpful to learn about this algorithm.

Photonic crystal with simple geometrical structure in primitive cell can be easily incorporated into the framework of simulated annealing optimization, because one can represent its structure with several variables, integral or continuous. Selective filters were designed based on one-dimensional photonic crystal by the SAA [76]. Here, the heuristic algorithm was employed together with mathematical analysis to generate a global optimization algorithm, which reached far beyond what common heuristic algorithms can achieve.

An all-dielectric compact magnetic resonator was designed based on the incorporation of disorder into an ordered photonic slab [77]. Natural materials rarely have strong response to the magnetic field component of light, but man-made meta-materials are more powerful to enhance this weak interaction. In this work, SAA was incorporated into artificial intelligence to help amplify this response. Due to the simple and clear structure of the element, we can take it as a good example for readers to understand the whole design procedure. The schematic of the element structure, optimization history, and relevant simulation results is depicted in Fig. 11. A plane wave excitation is given and the amplitude of normalized magnetic field H_z is strongly enhanced at a certain point, as depicted in Fig. 11D. A strong magnetic light-mater interaction is thus attained.

To employ an optimization algorithm in the design procedure, the author first declared an objective function,

$$F(\boldsymbol{x}) = \max\left[f_{\omega 1}(\boldsymbol{x}), f_{\omega 2}(\boldsymbol{x}), \ldots, f_{\omega n}(\boldsymbol{x}) \right] \tag{11}$$

Here, the components of vector $\boldsymbol{x}$ are the coordinates of the 25 cylinders in the rectangular array. $\omega 1$, $\omega 2$, …, ωn are discrete frequencies within the selected frequency range, and $f_{\omega i}(\boldsymbol{x})$ is the largest enhancement factor among all the spatial locations at the frequency ωi. Clearly, these factors are functions to the structure parameter $\boldsymbol{x}$. FEM analysis was incorporated in the optimization for solution estimations.

SAA was then employed to maximize the objective function in formula (11). The evolution history is depicted in Fig. 11B. The best value was found around the 50th iteration. To be clear, the ordered photonic slab was firstly optimized to find a better structure and make it the initial structure of the second optimization where disorder was introduced.

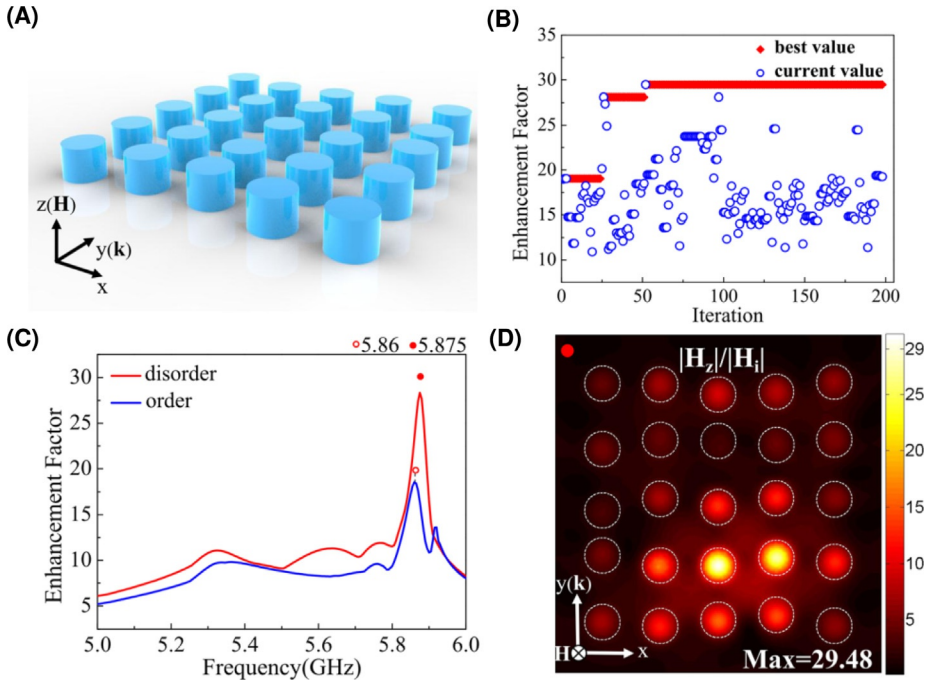


Fig. 11 (A) Schematic of the element structure; (B) optimization history of simulated annealing algorithm; (C) the spectra of the largest factor of the magnetic field profile; and (D) simulation result of the normalized magnetic field $|H_z|$.

Similar jobs including twisted light emitter [78] and microstrip patch antenna [79] have also been reported by the employment of SAA. Its generality allows devices in arbitrary forms, only asking the structure to be fully depicted by finite parameters. Except for directional utilization in inverse design, heuristic algorithms can produce training data for neural network [75]. Other group-dependent algorithms, like genetic algorithms, will come to a stage where most of the individuals share the similar structures, which contradicts with purpose of providing training data with good diversity.

It is also not difficult to find out its defects as a heuristic algorithm, which are high computation cost and experience-required hyperparameters, such as temperature and the cooling rate. Moreover, it fits only problems where the dimension of solution space is small, otherwise genetic algorithm shall be a better choice [80]. As introduced before, the validity of heuristic algorithm is promised by the law of large numbers. This is a natural defect for an algorithm because it requires a large search scope to provide a valid solution. Here for the SAA, one would find the output solution better if he/she sets a relatively higher initial temperature and smaller cooling rate, which would also inevitably bring more computation burden from more function evaluations.

The utilization of Metropolis principle enables SAA to jump out of local optimum and conducts a more global search in the solution space. By controlling the temperature and cooling rate, convergence velocity can be easily tuned, which gives more freedom in designing a complicated algorithm. The core idea of SAA, the Metropolis principle, can be easily incorporated into most of the algorithms. It is reasonable to believe that when it comes to the utilization of heuristic algorithm for an inverse design problem, the combination of various mechanisms shall bring us more possibilities on broadening the boundaries of nanophotonic devices.

3.4 Tabu search

3.4.1 Introduction

Tabu search (TS) is a kind of improved hill-climbing algorithm [81]. Hill-climbing algorithm, as introduced above, chooses the best solution from the neighborhood structure and terminates when running into a local minimum. TS enables tolerance over worse solutions and incorporates short-term memory recording previously visited solutions, to avoid cycling back.

In one iteration, TS evaluates all the solutions in the neighborhood structure and chooses the best one. If the move to this best-value solution has been recorded in the short-term memory, or tabu list, the algorithm turns to the second-best solution. After each iteration, the tabu list containing recently visited moves will be updated. If the best and second-best solutions are both recorded, the algorithm comes to a termination.

The definition of tabu list can be strict or not. For instance, if the variable v_1 changes from 0 to 1 in an iteration, a strict tabu list records (v_1), preventing any change of variable v_1. Less strict tabu lists can be (v_1, 1) or (v_1, 0, 1). One could see that too strict a tabu list forbids more solutions and faster termination, which means potential better solutions are abandoned simultaneously. To avoid this side effect, aspiration criteria is combined with the use of tabu list. If the best-value solution from the neighborhood is better than the best one ever found, the algorithm always accepts it, ignoring the tabu list. The flow chart of TS is given in Fig. 12 for a clear view. Even if the tabu list rejects the best option ω' from neighborhood structure, ω' can still be accepted by the aspiration criteria. Only if ω' is rejected by both criteria, the algorithm turns to the second-best solution ω'' and sees if it is accepted.

Notice that any small perturbation on the current solution brings a new solution into the neighborhood structure. In a solution space with too many design variables, the neighborhood structure can be too large to travel through. By introducing the strategy of randomness, the algorithm evaluates only part of the neighborhood structure. At the price of missing potential good results, this strategy greatly improves the efficiency.

In short, a simple TS is the combination of tabu list and hill-climbing algorithm. The strategy of abandoning recently visited solutions can be easily mixed up with other

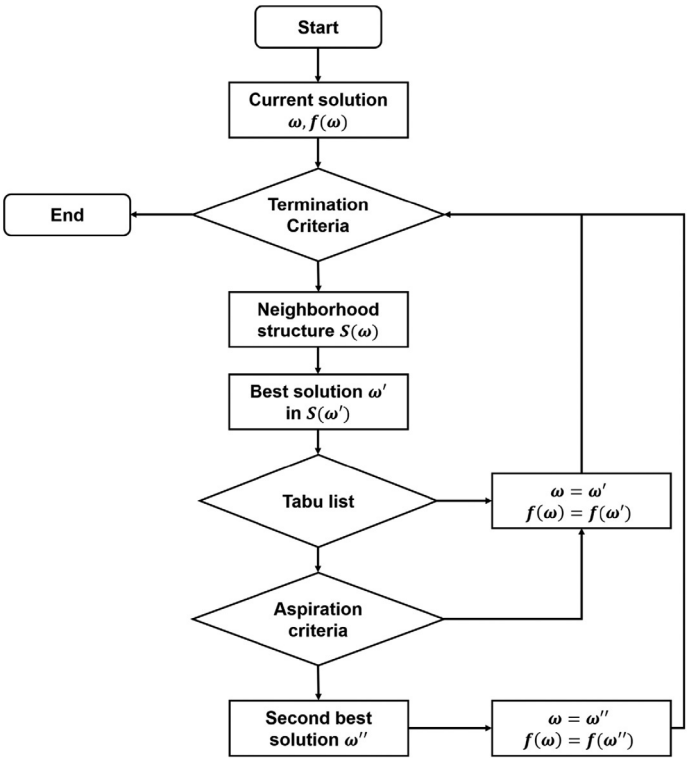


Fig. 12 Flow chart of tabu search.

optimizations and worked well in the previously proposed works. For a more detailed discussion, please refer to [81].

3.4.2 Applications

TS was used in the optimization of nanophotonic elements. Due to the preference on the solution space, optimization problems of material allotment like design of photonic crystals and meta-surface are better platforms for TS. We will introduce here the works done by employing TS.

An optical beam shaping element was designed by utilizing a parallel TS algorithm [82], which successfully generated comparable or better solutions compared to genetic algorithm, while requiring less computational time and fewer adjustable parameters. Here, the optimization problem greatly suits the TS because the structure of photonic crystal can be easily denoted as a binary vector of 0 and 1.

Metasurfaces, thin films with nanoscale patterning offer highly directional optical absorption and other charming functions. The combination of SAA and TS was applied

in the design of directional absorbing thin film [83]. Result of this technique proved efficient compared to the naive brute force alternative. This is a good example where two different strategies are incorporated together to work as a global random search optimization technique.

Gagnon et al. proposed a polarization filter based on integrated photonic element [84]. Like the optical beam shaping element, utilization of TS greatly suits the SOI structure and lithography technology, because the holes in thin film do not require extreme etching precision.

Ngo et al. developed a hybrid algorithm employing staged continuous tabu search (SCTS) and quasi-Newton method (QNM). A fiber Bragg grating (FBG) was designed by this algorithm [85]. In formal procedure, SCTS was supposed to find a good solution as close to the target as possible, while the QNM was later applied to carry on a local search to enhance the result from the first step.

Karim utilized direct tabu search (DTS) to design a triangular index-guiding photonic crystal fiber (PCF) [86]. In this device, hole-to-hole distance, circular air-hole diameter, solid-core diameter, ring number, and PCF length were all considered by DTS and all of them together constitute the whole solution space. Three synthesized PCFs with different parameters were provided with similar objective function. On this point, we could learn that heuristic algorithms can hardly give the global optimum, but only some local ones.

There are mainly two enhancements in TS compared to hill-climbing algorithm. First, TS is given the power to jump out of local minimum by allowing a worse solution to the current one. Second, the incorporation of short-term memory in TS helps avoid inefficient cycling back to previously visited solutions. Other strategies like aspiration criteria and probability-controlled searching direction also help improve this algorithm. To be clear, not all the strategies mentioned above are necessary for every optimization. Analysis and experiments should be taken before the final optimization.

3.5 Genetic algorithm

3.5.1 Introduction

In the following section, we will introduce the basic conceptions of genetic algorithm (GA), including chromosome, selection, mutation, and crossover. To give a panorama of GA, we will talk about some conclusions from the theory of survival of fittest, to meet the consensus which admits the validity of this algorithm. Assume there is population of certain kind of species, living in an environment where only the individuals with certain genes can survive. With time passing, individuals with nonadaptive genes will be weeded out, leaving the population with only adaptive individuals. However, the reproductive process brings more offspring into this group. After a long time, all the individuals in this group should share similar genes, with the same characters promising that they are not being weeded out by the environment.

The optimization scheme can be addressed in the following formula:

$$\min \text{objective} = f(\boldsymbol{\omega}), \boldsymbol{\omega} = (x_1, x_2, x_3, \ldots)$$
$$\text{s.t.} g_i(\boldsymbol{\omega}) = 0, h_i(\boldsymbol{\omega}) \leq 0, i = 1, 2, \ldots$$

where the objective function is the target of the designed device. When designing nanodevices, it can be written as the function measuring the device's performance. Take a two-channel wavelength router for an example, the objective can be written as transmissions at different channels. And the parameters $\boldsymbol{\omega} = (x_1, x_2, x_3 \ldots)$ is what determines the structure of a device, the so-called chromosome in GA.

As a population-based algorithm, every individual in this optimization plays a similar role to the individual-based algorithm like SA and TS. In each iteration, after function evaluations, individuals with worse performance should be abandoned with a certain probability. This step is called selection, where only adaptive individuals can produce offspring and spread their chromosomes.

The following step is a crossover, parents surviving the selection can exchange chromosomes and produce offspring to keep the population size invariant. To give the algorithm ability to jump out of local optimum, a step called mutation is incorporated. The chromosome of a new offspring is mutated at a mutation probability. Taking a binary integer chromosome for instance, the mutation operator swaps it from 0 to 1 or from 1 to 0. After all these steps, one iteration ends. The flow chart of GA is given in Fig. 13 to make it clear.

There are many hyperparameters to be decided by experience, the size of group, the number of abandoned individuals, crossover probability, mutation probability, and so on. Even so, the overall performance of GA is good enough to enable it to give near-global optimum in most cases. It has great convergence and produces fairly good results. However, because population-based heuristic algorithm mimics the lengthy evolution process, it is obviously time-consuming and requires more function evaluations, compared to individual-based heuristic algorithms, let alone the gradient-based ones.

During the past years, many variants of GA are presented, including real-coded GA, multi-objective GA, parallel GA, chaotic GA, and hybrid GA [87]. One word should be addressed, the refiner the method, the more specific the problem. For a new optimization problem at hand, analysis is necessary before application of various complex methods.

3.5.2 Applications

GA is very general and suitable to optimize different complex problems arising from different fields including economics [88], politics [89], management [90], and engineering [91]. In recent years, with the improvement of design methods for nanophotonic devices and the development of computer science, GA was also introduced into the design of

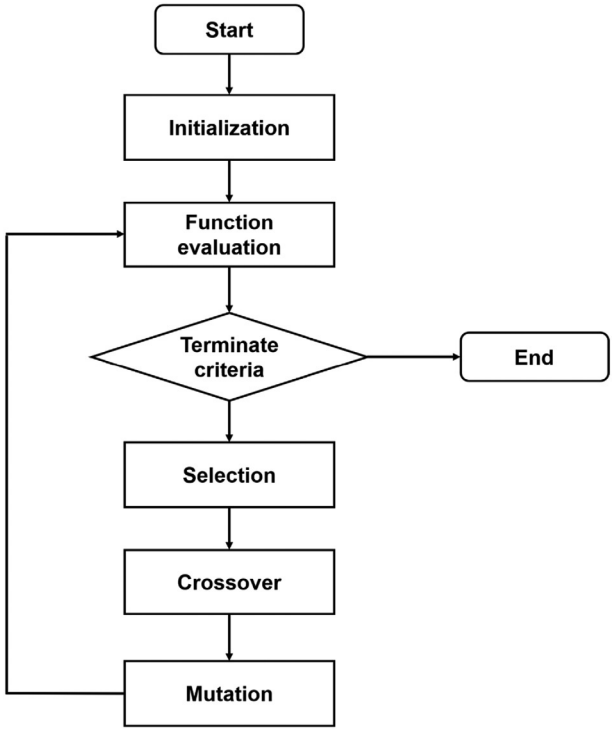


Fig. 13 Flow chart of genetic algorithm.

nanophotonics. Many works are reported, such as wavelength router [92], polarization rotator [93], polarization router [94], reflectors [95], special flat lens [96], light scattering [97], light absorption [98], high-resolution imaging [99], phase-controlled dielectric meta-surfaces [100], and nondeterministic quasi-random nanophotonic structures [101]. Below, some examples of GA's application in nanophotonics are introduced.

One typical work in this field is provided by Lu's group [92]. They have developed an intelligent algorithm by combining GA and finite element method (FEM) together to design an on-chip nanophotonic wavelength router, which is suitable for different structures, different materials, and different wavelength ranges. For example, one structure is shown in Fig. 14A, where the blue holes are silver and the white area is filled with air. The whole structure has a small footprint of $1.0\,\mu m \times 1.0\,\mu m$. Fig. 14B shows the simulated transmission spectrum covering from 480 to 950 nm. Fig. 14C and D is simulated magnitude of the electric field at two transmission peaks in Fig. 14B. In their design workflow, FEM is used to simulate wavelength router and to get transmission of each port. The components of design parameter $\boldsymbol{\omega}$ are set as position coordinates and side lengths of the inner rectangles, which are encoded into chromosomes of the population.

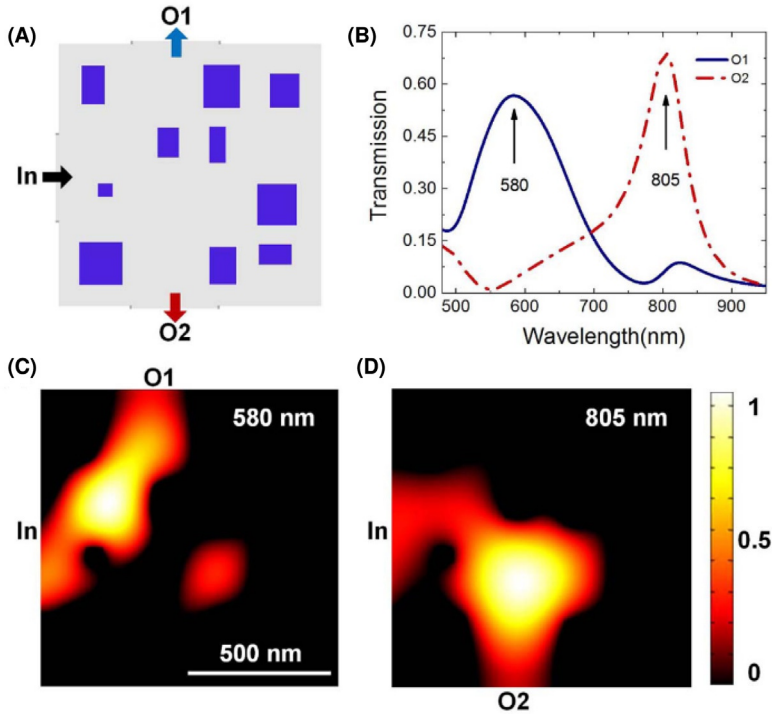


Fig. 14 (A) Sketch for the designed wavelength router; (B) simulated transmission spectrum; and (C and D) the simulated magnitude of the electric field at 580 and 805 nm, respectively.

Following the previous discussion, and the objective function should be a function of solution $\boldsymbol{\omega}$. But the concrete form of the objective function can be vital to the optimization result, and in most of the cases, is not easy to define.

In this work, the objective function is defined as harmonic mean of target transmission subtracting squared mean of unwanted transmission

$$\mathrm{F} = \left(\Sigma_i \frac{(N_i - N_0)^2}{i}\right)^{\frac{1}{2}} - J\left(\Sigma_j \frac{1}{S_j}\right)^{-1}, \tag{12}$$

where N_i represents the transmission in unwanted wavelengths and S_j represents the transmission in target wavelengths. By doing so, solutions need to promote all the signal transmission while restrain all the noise transmission to get a high function value.

In a later work, Lu et al. further developed the intelligent algorithm to successfully realize a polarization router as small as 970 nm × 1240 nm [94]. More detailed information is given in Fig. 15, from which we can also see how the structure evolves during the optimization. GA greatly decreased the size of the element while maintaining its performance. Judging from these two works, the validity and generality of GA are obvious.

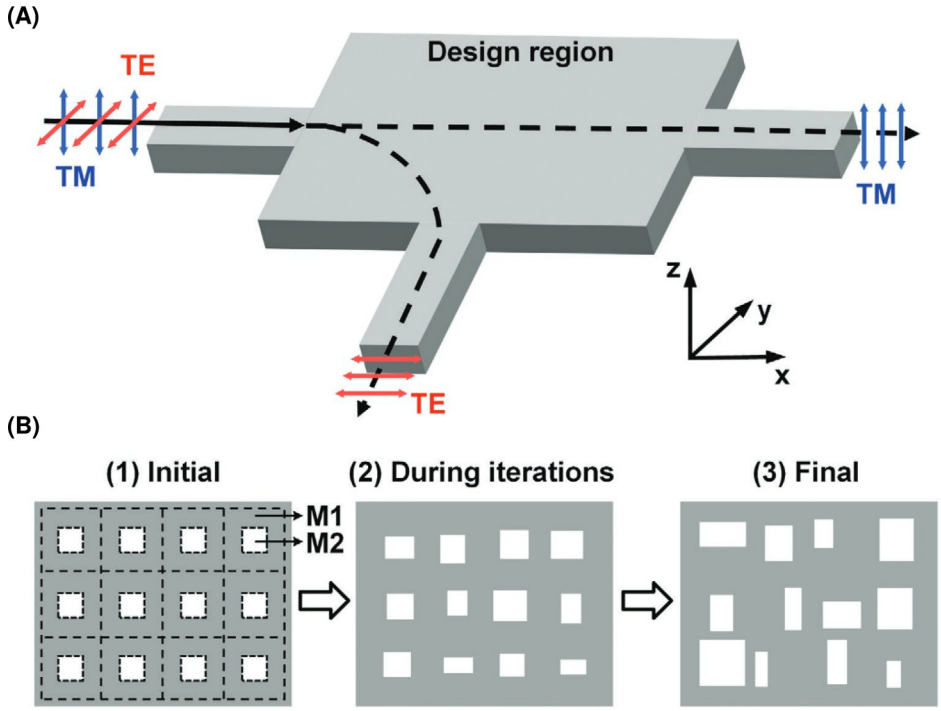


Fig. 15 (A) Sketch for the polarization router and (B) sketch for the optimization process.

As one of the most used heuristic algorithms, GA is good at global search, especially when given with certain parameters, like high mutation probability and a big number to abandon low-adaptive individuals. However, we shall notice that, a high computational cost must be the sacrifice of a global searching ability. When using heuristic algorithm, make sure the optimization has reached a balance to avoid this conflict.

GA is of high efficiency when dealing with problems with limited design variables. With the randomness in the generation of new individuals, GA can search solutions in a broad solution space, which can provide a better performance of nanophotonic devices in many cases. However, when the design variable number becomes extremely large, the convergence performance of GA will be weakened.

3.6 Differential evolution algorithm

3.6.1 Introduction

The differential evolution (DE) algorithm was proposed by Storn et al. in the 90s [102]. It is a stochastic model that simulates biological evolution. Through repeated iterations, those individuals that are adapted to the environment are preserved. However, compared

with other evolutionary algorithms, DE retains the global search strategy based on population, adopts real number coding, simple mutation operation based on difference and one-to-one competitive survival strategy, which reduces the complexity of a genetic operation.

Fig. 16 is the flow chart of the DE algorithm. The optimization process of the DE algorithm is very similar to that of the genetic algorithm (GA), and its main working steps include three operations: mutation, crossover, and selection. But the specific definitions of these operations are different from GA. The main idea of the DE algorithm is to start from a randomly generated initial group, use the difference vector of two individuals randomly selected from the population as the source of the random change of the third individual, and weight the difference vector according to certain rules. Summing with a third individual creates a new individual. This operation is called mutation. Then, the mutant individual is mixed with a predetermined target individual to generate a test individually, and this process is called crossover. If the fitness value of the test individual is better than the fitness value of the target individual, the test individual will replace the target individual in the next generation, otherwise, the target individual will still be preserved, and this operation is called selection. In the evolution process of each generation, each individual vector is used as the target individual once, and the algorithm keeps good

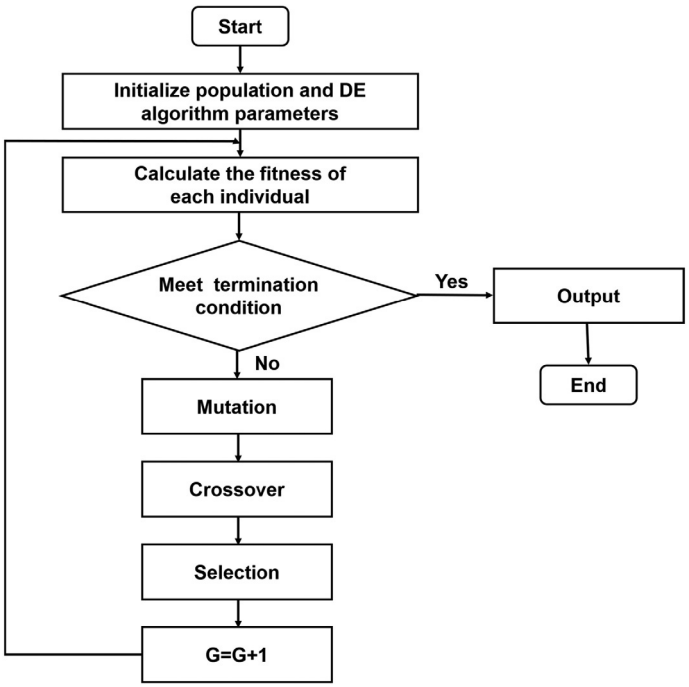


Fig. 16 Flow chart of differential evolution algorithm.

individuals and eliminates inferior individuals through continuous iterative calculation, and guides the search process to approach the global optimal solution.

The difference between GA and DE is that GA controls the probability of the selected offspring after the parent selection, crossover, and mutation according to the fitness value. In the maximization problem, the probability of an individual with a large fitness value being selected will be correspondingly larger. In the DE algorithm, the mutation vector is generated by the parental difference vector, and crosses with the parental individual vector to generate a new individual vector, which is directly selected with the parental individual. Obviously, the approximation effect of DE algorithm is more significant than that of GA.

3.6.2 Applications

Roy et al. proposed DE algorithm for efficient allocation of wavelength converters in wavelength division multiplexed optical networks [103]. They exploited the search capability of the DE algorithm to achieve near-ideal performance using a minimum number of wavelength converters at each node in wavelength division multiplexed optical network. As the time complexity is concerned, the proposed algorithm performs better than the previous approach for large-scale practical problems.

The DE algorithm is an efficient global optimization algorithm. The algorithm principle is simple and the control parameters are only a few. The main parameters are only the population size, the crossover probability, and the scaling factor. The algorithm has been successful in many fields due to its ease of use, robustness, and strong global optimization ability.

3.7 Particle swarm optimization algorithm

3.7.1 Introduction

Proposed by Kennedy and Eberhart [104], particle swarm optimization (PSO) is a simple social model that mimics the real rules of the bird flock's foraging. Unlike the previously introduced simulated annealing algorithm and tabu search, PSO is a swarm algorithm, with each particle in the swarm playing the same role of the individual algorithm. It should be proper to take it as an analogy of parallel individual algorithms. In this analogy, each particle in the swarm searches the feasible region based on its own memory and experience from the whole swarm. In the following section, the basic PSO template will be introduced, followed by several strategies to enhance the performance of PSO algorithm.

As a population optimization, every individual in the swarm represents a single solution. The individuals are initiated with randomized velocity, which guides them to update the solutions in each iteration, following formula (13):

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^n + \boldsymbol{v}_i^n \Delta t \tag{13}$$

The solution and velocity of particle i in the nth iteration are denoted as $\boldsymbol{x}_i^n$ *and* $\boldsymbol{v}_i^n$, respectively. Also, one should notice that the optimization procedure is discretized by unit time step Δt. The velocity of a particle is decided by its own memory and knowledge from the whole swarm. It should be the linear combination of its previous velocity, which comes from the effect of inertance, the direction pointing to the current best solution to the particle, and the current best solution to the whole swarm. To be concise, refer to formula (14)

$$\boldsymbol{v}_i^{n+1} = \omega \boldsymbol{v}_i^n + \frac{c_1 r_1 \left(\boldsymbol{p}_i - \boldsymbol{x}_i^n\right)}{\Delta t} + \frac{c_2 r_2 \left(\boldsymbol{p}_g^n - \boldsymbol{x}_i^n\right)}{\Delta t} \tag{14}$$

Here, the $\boldsymbol{p}_i$ and $\boldsymbol{p}_g^n$ are the current best solution found by the particle i so far and the best solution found by the whole swarm in nth iteration, respectively.

Refer to the flow chart in Fig. 17 for a brief review.

Naturally, the birds flock should spread as much as possible to search throughout the forest. Also, the distribution of the initial swarm may pose an effect on the result.

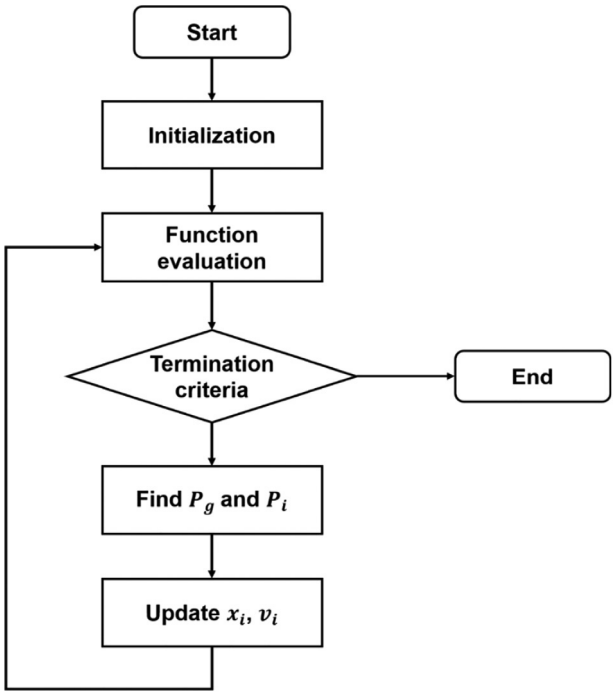


Fig. 17 Flow chart of particle swarm optimization algorithm.

Research was carried out to check if special initial distribution could generate better solutions. The conclusion shows that the initial distribution is not important to the overall performance of PSO algorithm. A uniform random distribution is favored for convenience, in accordance with intuition. Without external implementation, the algorithm should keep running even when all the particles gather at the same point. A scientific convergence criterion would help avoid unnecessary function evaluations when the algorithm has found a solution good enough. A simple strategy can be predefining a maximum change in objective function.

Like all the other heuristic algorithms, there are some hyperparameters, c_1, c_2, and ω, which are decided by experience in PSO algorithm. There are literatures proposing a combination of $c_1 = 2$, $c_2 = 2$ which means equal confidence in individual experience and collective experience. Other strategies containing a bias of either c_1 or c_2 are reasonable for specific problem, but extreme parameters ignoring either of them are not recommended. For the parameter ω, there used to be literature suggesting using $0.8 < \omega < 1.4$, starting from a larger value and ending with smaller value. Larger ω stands for more global searching ability and vice versa.

In real engineering problems like inverse design, the solution space is constrained in one way or another. For PSO algorithm, an exterior penalty function can be applied like GA in the following formula (15):

$$\widetilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \alpha \sum_{i=1}^{m} \max\left[0, g_i(x)\right]^2 \tag{15}$$

Here, the $f(\boldsymbol{x})$ is the original objective function, α is a penalty parameter and $g_i(\boldsymbol{x})$ is the violation of the ith constraint. By doing so, particles moving outside the feasible region have worse objective value, and never attract other particles.

3.7.2 Applications

The simple structure of PSO allowed a great number of applications based on this algorithm. We have discussed the trend of hybridizing different algorithms to form a comprehensive algorithm, and we can see this attempt to improve the performance of algorithms in many works concerning PSO.

By introducing a nanocavity into photonic waveguide, M. Djavid et al. optimized the radius of this nanocavity and thus proposed a notch filter based on PSO [105]. The combination of PSO and photonic waveguide generated lots of works in the past, including directional emission [106], slow light [107], and index sensor [108]. Due to its simple structure, the solution space is small and easy to search through by PSO. More charming works from this platform can be expected in the following years. Other creative structures are also adorable in bringing charming applications. Wang et al. proposed a 2×2 multimode interference coupler based on strongly guiding deep-etched InP waveguide

[109]. Akhlaghi et al. designed a photonic crystal fiber Raman amplifier [110]. Antiresonant fibers with extremely low confinement loss were also designed by PSO, illustrating its generality [111]. An all-optical NOT gate was proposed by Lu et al. using a QR-code-like structure [112]. The meta-surface supported work, a Fano resonance element with high Q factor was proposed recently, with also relatively simple structure in accordance with the requirement of PSO [113].

The validity of PSO has been examined by all these previous works. It does not require continuity in problem definition, and easily generates near-global solutions. However, as a heuristic algorithm, it requires more function evaluations and brings more computational budget. Even parallel computing can help alleviate this problem, huge computational cost is still not negligible.

3.8 Ant colony algorithm

3.8.1 Introduction

Ant colony algorithm (ACA) was first proposed by Italian scholars Dorigo et al. in the 1990s. In the process of studying ants foraging, they found that the behavior of a single ant is relatively simple, but the ant colony can reflect some intelligent behavior. For example, ant colonies can find the shortest path to food in different environments. Ants release pheromone on the path they pass through. Ants in the ant colony have the ability to perceive pheromone. They will walk along the path with higher concentration of pheromone, and each passing ant will leave pheromone on the path. After a period of time, the entire ant colony will follow the shortest path to reach the food.

Fig. 18 is the flow chart of the optimization process of ACA [114]. The basic ACA is expressed as follows: at the initial moment, m ants are randomly placed, and the initial amount of pheromone on each path is equal. At the moment t, the probability of the kth ant moving from node i to node j is

$$p_{ij} = \frac{\tau_{ij}\eta_{ij}}{\sum \tau_{ij}^{\alpha}\eta_{ij}^{\beta}}$$

where τ_{ij} is the intensity of pheromone on the line of ij at time t; η_{ij} is the heuristic factor indicating the expected degree of ant moving from node i to node i, usually taking the reciprocal of the distance between i and j. α and β represent the degree of relative importance of the pheromone and expectancy heuristic factors, respectively. After each ant traverses once, the pheromone updated on each path is

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij},$$

where ρ represents the loss level of the total amount of pheromone on the path and $1-\rho$ represents the residual factor of the pheromone. $\Delta\tau_{ij}$ represents the increment of the pheromone on path ij after the completion of this iteration, which can be represented as

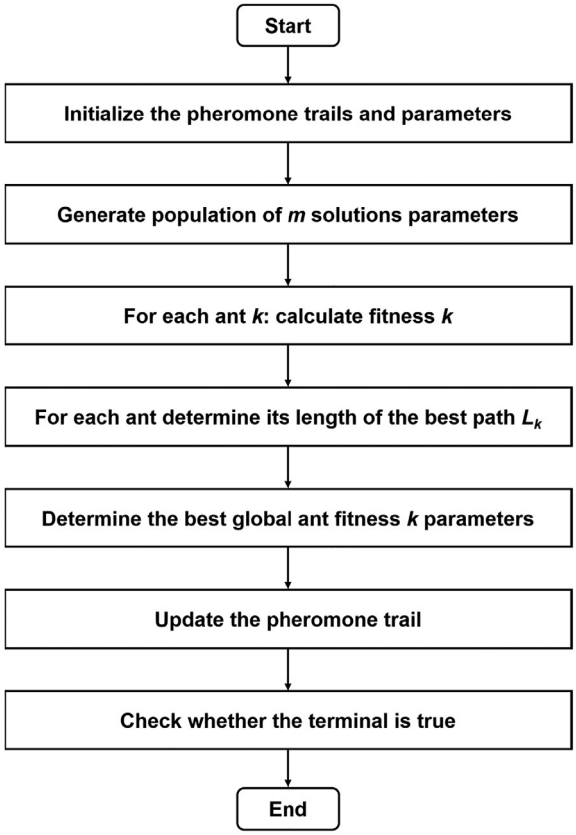


Fig. 18 Flow chart of ant colony algorithm.

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{L_k}, & \text{the } k\text{th ant gose by } ij, \\ 0, & \text{others}, \end{cases}$$

where L_k represents the length of the path traveled by the kth ant in this traversal.

3.8.2 Applications

Using ACA, Saouane et al. obtained the setting of the optimal inclination angle for the photovoltaic collector through simulation and improved the efficiency of the collector [114]. The optimum angle varies in 1 year, from a minimum of 0 degree in June to a maximum of 59 degree in December and January. Guo et al. used ACA method for optimizing the antireflection coating system [115]. They used ACA to optimize the antireflection coating system with broadband omnidirectional characteristics for silicon solar cells incorporated with the solar spectrum.

ACA adopts the mechanism of positive feedback, which makes the search process continue to converge and finally approach the optimal solution. During the search process, multiple individuals perform parallel computing at the same time, which greatly improves the computing power and operating efficiency of the algorithm, and makes it easier to find the global optimal solution.

4. Conclusion

In conclusion, the introduction of intelligent algorithms in nanophotonic devices has accelerated the prosperity of nanophotonics field. Different intelligent algorithms have different frameworks and workflows. These algorithms can be generally classified into two kinds stated above, gradient-based and heuristic (gradient-independent), according to whether they use gradient in optimization process.

Gradient-based algorithms calculate gradient information of the current solution to find an iterative direction. Topology optimization is a general category in gradient-based algorithms and includes variable density method, level set method, and BESO, which are used to design nanophotonic structures in many cases. Variable density method can extremely enlarge solution space. Level set method is convenient when dealing with changeable boundaries. BESO is successfully applied in the design of various photonic crystals. Different from topology optimization, objective-first algorithm seeks for a solution with high performance even in the expense of unfulfilling the governing equations. Objective-first algorithm has been proven successful in designing various on-chip integrated nanophotonic devices. To get the gradient information efficiently, adjoint method is applied in most gradient-based algorithms, including the other gradient-based intelligent algorithms. Adjoint method can calculate gradients with only two simulations no matter how large the solution space is.

Heuristic algorithms are free of gradient information and they generate new solutions based on various strategies, with whose validity being promised by Monte-Carlo method. Heuristic algorithms can be further divided into individual-based and population-based algorithms. Individual-based heuristic algorithms include hill-climbing algorithm, DBS algorithm, SAA, and TS algorithm. Population-based heuristic algorithms include GA, DE algorithm, PSO algorithm, and ACA. Except for hill-climbing algorithm and DBS algorithm, most heuristic algorithms are all armed with the ability to jump out of local minimum. They do not promise a global optimum, but they can provide a solution near global optimum within finite calculation steps. Heuristic algorithms are suitable to solve problems with limited variable number and may become inefficient when dealing with a problem with too many design variables. This chapter introduced intelligent algorithms applied in nanophotonic design and gave examples of design using these algorithms.

As for the utilization of intelligent algorithms, we believe during the designing process of nanophotonic devices, multiple algorithms can be adopted simultaneously to provide

efficient and optimal solutions, rather than just one algorithm. In addition, with the development of the algorithms and computer power force, more and more intelligent algorithms may be further developed and applied to various communities. When there are too many algorithms to choose, the more frequently used and reported method may be a better reference for similar problems. Sometimes, when facing complicated problems, we do not rely on one algorithm. The results obtained from one optimization algorithm can be regarded as the initial structure or guess for the second optimization algorithm, and a complicated problem can be divided into several optimization steps by combing serval intelligent algorithms instead of only one algorithm. Whatever the complexity of the problems, we believe, more intelligent algorithms can bring solutions. As Philip W. Anderson says: More is different.

References

[1] C. Lu, Y.-C. Liu, X. Hu, H. Yang, Q. Gong, Sci. Rep. 6 (2016) 27428.
[2] R.E. Caflisch, Acta Numer. 7 (1998) 1.
[3] K. Binder, A. Baumgärtner (Eds.), Applications of the Monte Carlo Method in Statistical Physics, second ed., Springer, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1987.
[4] L. Ma, J. Li, Z. Liu, Y. Zhang, N. Zhang, S. Zheng, C. Lu, Chin., Opt. Lett. 19 (2021) 011301.
[5] M.P. Bendsøe, N. Kikuchi, Comput. Methods Appl. Mech. Eng. 71 (1988) 197.
[6] M.P. Bendsøe, Struct. Optim. 1 (1989) 193.
[7] J.S. Jensen, O. Sigmund, Laser Photonics Rev. 5 (2011) 308.
[8] D.A. Tortorelli, P. Michaleris, Inverse Probl. Sci. Eng. 1 (1994) 71.
[9] J.S. Jensen, O. Sigmund, Appl. Phys. Lett. 84 (2004) 2022.
[10] L.H. Frandsen, A. Harpøth, P.I. Borel, M. Kristensen, J.S. Jensen, O. Sigmund, Opt. Express 12 (2004) 5916.
[11] P.I. Borel, A. Harpøth, L.H. Frandsen, M. Kristensen, P. Shi, J.S. Jensen, O. Sigmund, Opt. Express 12 (2004) 1996.
[12] F. Wang, J.S. Jensen, O. Sigmund, J. Opt. Soc. Am. B 28 (2011) 387.
[13] F. Wang, R.E. Christiansen, Y. Yu, J. Mørk, O. Sigmund, Appl. Phys. Lett. 113 (2018), 241101.
[14] L.H. Frandsen, Y. Elesin, L.F. Frellsen, M. Mitrovic, Y. Ding, O. Sigmund, K. Yvind, Opt. Express 22 (2014) 8525.
[15] L.F. Frellsen, Y. Ding, O. Sigmund, L.H. Frandsen, Opt. Express 24 (2016) 16866.
[16] P. Camayd-Muñoz, C. Ballew, G. Roberts, A. Faraon, Optica 7 (2020) 280.
[17] Y. Elesin, B.S. Lazarov, J.S. Jensen, O. Sigmund, Photonics Nanostruct. Fundam. Appl. 10 (2012) 153.
[18] R.E. Christiansen, F. Wang, O. Sigmund, S. Stobbe, Nanophotonics 8 (2019) 1363.
[19] H. Yuan, Z. Liu, M. Wei, H. Lin, X. Hu, C. Lu, Micromachines 12 (2021) 1506.
[20] P. Wei, Z. Li, X. Li, M.Y. Wang, Struct. Multidiscip. Optim. 58 (2018) 831.
[21] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, 2003.
[22] C.Y. Kao, S. Osher, E. Yablonovitch, Appl. Phys. B Lasers Opt. 81 (2005) 235.
[23] C.M. Lalau-Keraly, S. Bhargava, O.D. Miller, E. Yablonovitch, Opt. Express 21 (2013) 21693.
[24] J. Lu, J. Vučković, Opt. Express 21 (2013) 13351.
[25] A.Y. Piggott, Nat. Photonics 9 (2015) 5.
[26] A.Y. Piggott, J. Lu, T.M. Babinec, K.G. Lagoudakis, J. Petykiewicz, J. Vučković, Sci. Rep. 4 (2015) 7210.
[27] A.Y. Piggott, J. Petykiewicz, L. Su, J. Vučković, Sci. Rep. 7 (2017) 1786.
[28] N. Lebbe, C. Dapogny, E. Oudet, K. Hassan, A. Gliere, J. Comput. Phys. 395 (2019) 710.
[29] Y.M. Xie, G.P. Steven, Comput. Struct. 49 (1993) 885.

[30] X. Huang, Y.M. Xie, M.C. Burry, Adv. Struct. Eng. 10 (2007) 727.
[31] X. Huang, Y.M. Xie, M.C. Burry, JSME Int. J. Ser. C Mech. Syst. Mach. Elem. Manuf. 49 (2006) 1091.
[32] F. Meng, X. Huang, B. Jia, J. Comput. Phys. 302 (2015) 393.
[33] Y. Chen, F. Meng, G. Li, X. Huang, Smart Mater. Struct. 28 (2019), 015025.
[34] F. Meng, S. Li, H. Lin, B. Jia, X. Huang, Finite Elem. Anal. Des. 117–118 (2016) 46.
[35] Y. Chen, F. Meng, B. Jia, G. Li, X. Huang, Phys. Status Solidi (RRL) 13 (2019) 1900175.
[36] Y. Chen, F. Meng, Y. Kivshar, B. Jia, X. Huang, Phys. Rev. Res. 2 (2020), 023115.
[37] Y. Chen, F. Meng, J. Zhu, X. Huang, Appl. Math. Model. 102 (2022) 194.
[38] Y. Chen, F. Meng, Z. Lan, B. Jia, X. Huang, Phys. Rev. Appl. 15 (2021), 034053.
[39] J. Lu, J. Vučković, Opt. Express 20 (2012) 7221.
[40] W. Hadibrata, H. Noh, H. Wei, S. Krishnaswamy, K. Aydin, Laser Photonics Rev. 15 (2021), 2000556.
[41] F. Callewaert, S. Butun, Z. Li, K. Aydin, Sci. Rep. 6 (2016) 32577.
[42] W. Hadibrata, H. Wei, S. Krishnaswamy, K. Aydin, Nano Lett. 21 (2021) 2422.
[43] L. Su, A.Y. Piggott, N.V. Sapra, J. Petykiewicz, J. Vučković, ACS Photonics 5 (2018) 301.
[44] C. Sideris, E. Garza, O.P. Bruno, ACS Photonics 6 (2019) 3233.
[45] T.W. Hughes, M. Minkov, I.A.D. Williamson, S. Fan, ACS Photonics 5 (2018) 4781.
[46] J. Wang, Y. Shi, T. Hughes, Z. Zhao, S. Fan, Opt. Express 26 (2018) 3236.
[47] D. Sell, J. Yang, S. Doshay, R. Yang, J.A. Fan, Nano Lett. 17 (2017) 3752.
[48] T. Phan, D. Sell, E.W. Wang, S. Doshay, K. Edee, J. Yang, J.A. Fan, Light Sci. Appl. 8 (2019) 48.
[49] J. Jiang, J.A. Fan, Nano Lett. 19 (2019) 5366.
[50] F. Wen, J. Jiang, J.A. Fan, ACS Photonics 7 (2020) 2098.
[51] M. Mansouree, H. Kwon, E. Arbabi, A. McClung, A. Faraon, A. Arbabi, Optica 7 (2020) 77.
[52] M. Mansouree, A. McClung, S. Samudrala, A. Arbabi, ACS Photonics 8 (2021) 455.
[53] T. Lin, F. Tian, P. Shi, F.S. Chau, G. Zhou, X. Tang, J. Deng, Opt. Lett. 40 (2015) 3504.
[54] Q. Quan, P.B. Deotare, M. Loncar, Appl. Phys. Lett. 96 (2010), 203102.
[55] M.A. Seldowitz, J.P. Allebach, D.W. Sweeney, Appl. Opt. 26 (1987) 2788.
[56] B.B. Chhetri, S. Yang, T. Shimomura, Appl. Opt. 39 (2000) 5956.
[57] B. Shen, P. Wang, R. Polson, R. Menon, Nat. Photonics 9 (2015) 378.
[58] B. Shen, P. Wang, R. Menon, Opt. Express 22 (2014) A311.
[59] B. Shen, P. Wang, R. Polson, R. Menon, Optica 1 (2014) 356.
[60] P. Wang, R. Menon, Opt. Express 21 (2013) 6274.
[61] G. Kim, J.A. Domínguez-Caballero, R. Menon, Opt. Express 20 (2012) 2814.
[62] G. Kim, J.A. Dominguez-Caballero, H. Lee, D.J. Friedman, R. Menon, Phys. Rev. Lett. 110 (2013), 123901.
[63] T.R.M. Sales, D.H. Raguin, Appl. Opt. 38 (1999) 3012.
[64] B. Shen, P. Wang, R. Polson, R. Menon, Opt. Express 22 (2014) 27175.
[65] G. Kim, R. Menon, Appl. Phys. Lett. 105 (2014), 061114.
[66] Y. Liu, S. Wang, Y. Wang, W. Liu, H. Xie, Y. Yao, Q. Song, X. Zhang, Y. Yu, K. Xu, Opt. Lett. 44 (2019) 4495.
[67] B. Shen, R. Polson, R. Menon, Nat. Commun. 7 (2016) 13126.
[68] H. Xie, Y. Liu, S. Wang, Y. Wang, Y. Yao, Q. Song, J. Du, Z. He, K. Xu, IEEE Photon. Technol. Lett. 32 (2020) 166.
[69] Y. Liu, K. Xu, S. Wang, W. Shen, H. Xie, Y. Wang, S. Xiao, Y. Yao, J. Du, Z. He, Q. Song, Nat. Commun. 10 (2019) 3263.
[70] Y. Liu, W. Sun, H. Xie, N. Zhang, K. Xu, Y. Yao, S. Xiao, Q. Song, Opt. Lett. 43 (2018) 2482.
[71] H. Ma, J. Huang, K. Zhang, J. Yang, Opt. Commun. 462 (2020), 125329.
[72] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, J. Chem. Phys. 21 (1953) 1087.
[73] D. Delahaye, S. Chaimatanan, M. Mongeau, Handbook of Metaheuristics, Springer, 2019, pp. 1–35.
[74] M.H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, K. Parsons, Sci. Rep. 9 (2019) 1368.
[75] C. Qiu, X. Wu, Z. Luo, H. Yang, G. Wang, N. Liu, B. Huang, Opt. Commun. 483 (2021), 126641.
[76] H.A. Badaoui, M. Abri, Prog. Electromagn. Res. B 53 (2013) 107.

[77] Y. Xie, M. Liu, T. Feng, Y. Xu, Nanophotonics 9 (2020) 3629.
[78] Z. Xie, T. Lei, F. Li, H. Qiu, Z. Zhang, H. Wang, C. Min, L. Du, Z. Li, X. Yuan, Light Sci. Appl. 7 (2018) 18001.
[79] J.E. Richie, C. Ababei, J. Comput. Des. Eng. 4 (2017) 249.
[80] L.D. Khalaf, A.F. Peterson, Int. J. Microw. Millim. Wave Comput. Aided Eng. 7 (1997) 108.
[81] M. Gendreau, J.-Y. Potvin (Eds.), Handbook of Metaheuristics, Springer US, Boston, MA, 2010.
[82] D. Gagnon, J. Dumont, L.J. Dubé, Opt. Lett. 38 (2013) 2181.
[83] S. Hajimirza, G. El Hitti, A. Heltzel, J. Howell, Int. J. Therm. Sci. 62 (2012) 93.
[84] D. Gagnon, J. Dumont, J.-L. Déziel, L.J. Dubé, Opt. Lett. 39 (2014) 5768.
[85] N.Q. Ngo, R.T. Zheng, J.H. Ng, S.C. Tjin, L.N. Binh, J. Lightwave Technol. 25 (2007) 799.
[86] F. Karim, Opto-Electron. Rev. 25 (2017) 41.
[87] S. Katoch, S.S. Chauhan, V. Kumar, Multimed. Tools Appl. 80 (2021) 8091.
[88] A.E. Drake, R.E. Marks, in: S.-H. Chen (Ed.), Genetic Algorithms and Genetic Programming in Computational Finance, Springer US, Boston, MA, 2002, pp. 29–54.
[89] R. Shang, J. Bai, L. Jiao, C. Jin, Phys. A: Stat. Mech. Appl. 392 (2013) 1215.
[90] K.J. Oh, T.Y. Kim, S. Min, Expert Syst. Appl. 28 (2005) 371.
[91] H.S. Bernardino, H.J.C. Barbosa, and A.C.C. Lemonge, n.d. in 2007 IEEE Congress on Evolutionary Computation, pp. 646–653.
[92] Z. Liu, X. Liu, Z. Xiao, C. Lu, H.-Q. Wang, Y. Wu, X. Hu, Y.-C. Liu, H. Zhang, X. Zhang, Optica 6 (2019) 1367.
[93] Z. Yu, H. Cui, X. Sun, Opt. Lett. 42 (2017) 3093.
[94] C. Lu, Z. Liu, Y. Wu, Z. Xiao, D. Yu, H. Zhang, C. Wang, X. Hu, Y. Liu, X. Liu, X. Zhang, Adv. Opt. Mater. 8 (2020), 1902018.
[95] Z. Yu, H. Cui, X. Sun, Photon. Res. 5 (2017) B15.
[96] M.D. Huntington, L.J. Lauhon, T.W. Odom, Nano Lett. 14 (2014) 7195.
[97] M. Liu, Y. Xie, T. Feng, Y. Xu, Opt. Lett. 45 (2020) 968.
[98] C. Liu, S.A. Maier, G. Li, ACS Photonics 7 (2020) 1716.
[99] L. Wang, W. Yan, R. Li, X. Weng, J. Zhang, Z. Yang, L. Liu, T. Ye, J. Qu, Nanophotonics 7 (2018) 1971.
[100] Y. Fan, Y. Xu, M. Qiu, W. Jin, L. Zhang, E.Y. Lam, D.P. Tsai, D. Lei, Nanophotonics 9 (2020) 3931.
[101] S. Yu, C. Wang, Y. Zhang, B. Dong, Z. Jiang, X. Chen, W. Chen, C. Sun, Sci. Rep. 7 (2017) 3752.
[102] R. Storn, K. Price, J. Glob. Optim. 11 (1997) 341.
[103] K. Roy, M.K. Naskar, Photon Netw. Commun. 16 (2008) 31.
[104] J. Kennedy, R. Eberhart, Proceedings of ICNN'95—International Conference on Neural Networks, IEEE, Perth, WA, Australia, 1995, pp. 1942–1948.
[105] M. Djavid, S.A. Mirtaheri, M.S. Abrishamian, J. Opt. Soc. Am. B 26 (2009) 849.
[106] M.S. Kumar, S. Menabde, S. Yu, N. Park, J. Opt. Soc. Am. B 27 (2010) 343.
[107] K. Abedi, S.M. Mirjalili, Opt. Commun. 339 (2015) 7.
[108] F. Rahman-Zadeh, M. Danaie, H. Kaatuzian, Opto-Electron. Rev. 27 (2019) 369.
[109] Q. Wang, S.-T. Ho, J. Lightwave Technol. 28 (2010) 1298.
[110] M. Akhlaghi, F. Emami, J. Opt. Soc. Korea 17 (2013) 237.
[111] F. Meng, X. Zhao, J. Ding, Y. Niu, X. Zhang, L. Yang, X. Wang, S. Lou, X. Sheng, G. Tao, S. Liang, Opt. Express 29 (2021) 35544.
[112] Q. Lu, X. Yan, Y. Luo, B. Li, X. Zhang, X. Ren, Asia Communications and Photonics Conference (ACP) 2018, OSA Technical Digest, paper Su2A.110, Optica Publishing Group, 2018.
[113] Y. Chen, Z. Ding, M. Zhang, J. Zhou, M. Li, M. Zhao, J. Wang, Appl. Opt. 60 (2021) 9200.
[114] I. Saouane, A. Chaker, B. Zaidi, C. Shekhar, Eur. Phys. J. Plus 132 (2017) 106.
[115] X. Guo, H.Y. Zhou, S. Guo, X.X. Luan, W.K. Cui, Y.F. Ma, L. Shi, Opt. Express 22 (2014) A1137.

CHAPTER 5

Artificial intelligence (AI) enhanced nanomotors and active matter

Frank Cichos[a], **Santiago Muiños Landin**[b], **and Ravi Pradip**[c]
[a]Leipzig University, Molecular Nanophotonics Group, Leipzig, Germany
[b]AIMEN Technology Centre, Smart Systems and Smart Manufacturing–Artificial Intelligence and Data Analytics Laboratory, Pontevedra, Spain
[c]Universitat Politècnica De Catalunya, Barcelona, Spain

1. Introduction

Research on active matter and its constituents—active particles—is concerned with understanding complex self-organization processes that occur in living systems at all length scales from individual bacteria to large organisms, robots, and societies. In most of these systems, collective behavior with dynamic patterns emerges as a consequence of energy uptake at the smallest scales. Active particles are the driving element of these organizational processes and are connected by physical interactions with a large variety of other different active and passive components. Already at the level of a single bacterium or cell, living systems have developed the ability to sense environmental signals that include, for example, chemical, mechanical, or thermal cues.

A single bacterium such as *E. coli* contains sensory arrays that couple chemoattractant binding with the machinery of flagellar motion to enable a process known as chemotaxis. Chemotaxis is the ability to swim up or down a chemical gradient without necessarily being able to directly measure the gradient of a chemical species at each point [1–3]. The signals involved in the resulting run-and-tumble motion known for *E. coli* and the signaling pathways explored pave the idea that mechanisms are already at work at these length scales which follow the concepts of computational information storage and processing. On larger length scales, such as for a human, such concepts are common and help us to perform actions based on the different experiences we have made. The variety of decisions we make is based on what we have learned. This adds a new quality to a living system, which is the ability to learn and correlate signals with actions without a physical change of the system itself. Living organisms as complex systems of active and passive matter are, therefore, the blueprint for adaptive materials that can perceive environmental influences, process this information, and respond with a dedicated reaction. As such, it has inspired artificial intelligence, machine learning, and robotics, and many modern approaches have their origins in our coarse-grained view of information processing in living systems. Machine learning is also stimulating biological research, as modern

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00005-6
Copyright © 2023 Elsevier Inc.
All rights reserved.

artificial intelligence tools are helping to unravel complex biological processes, including predictions about the conformation of small components such as proteins [4] or the analysis of medical images [5], to name just two examples.

Machine learning (Fig. 1) has already become an important tool in the field of active matter and soft matter physics within the last years [6,7]. Approaches with different machine learning techniques, either supervised, semi-supervised, or unsupervised, have been applied to study a variety of different aspects. Reservoir computing, for example, has been employed to predict chaotic dynamics of fluids [8]. Statistical learning frameworks have been applied to obtain the governing hydrodynamic equations for active particles from single particle trajectories [9,10]. Similarly, stochastic neighborhood embedding (t-SNE) and k-means clustering as methods of unsupervised learning were used to study how collective motion of bacteria in biofilms emerge from single bacterial

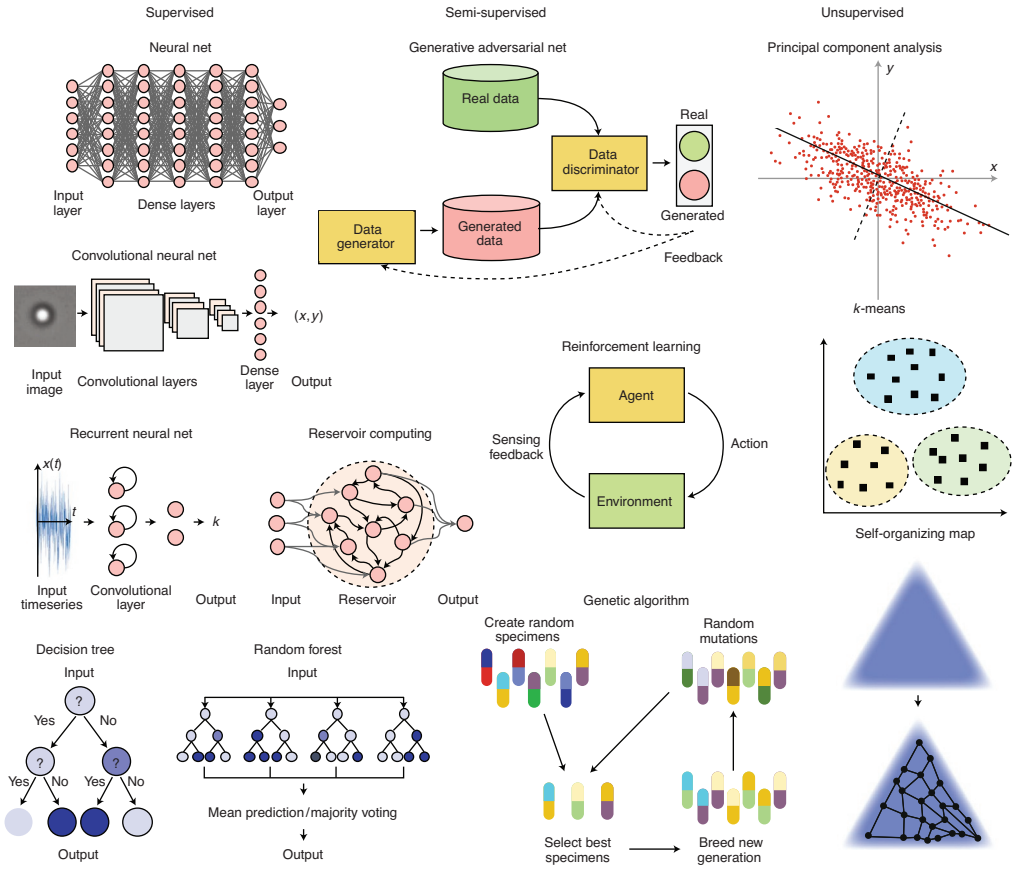


Fig. 1 Overview of different machine learning methods. *Reprinted by permission from Springer Nature: Nature Machine Intelligence, F. Cichos, K. Gustavsson, B. Mehlig, G. Volpe, Machine learning for active matter, Nat. Mach. Intell. 2 (2020) 94–103, Copyright 2020.*

properties [11]. Time series analysis for anomalous diffusion characterization has been performed and compared in the AnDi challenge with various machine learning techniques [11]. Image analysis is very important in the study of active matter, especially in the case of complex shapes and inhomogeneous signal-to-noise ratio in imaging techniques. Here, supervised convolutional neural networks are used not only for segmentation, tracking, and classification of objects [12–15] but also for the generation of artificial training data using actor-critic generative adversarial networks (GANs).

A special area of machine learning directly related to the human learning process is reinforcement learning (RL), which is also the subject of the remainders of this chapter. Reinforcement learning is neither supervised nor unsupervised but rather an exploratory technique where an agent learns by interacting with the environment. In the field of active matter and active particles, numerical and experimental researches have been carried out either with a focus on learning a strategy or finding a path to navigate toward a fixed goal. Reddy and coworkers used reinforcement learning to find a strategy to efficiently use turbulent convective thermal winds by gliders [16]. The glider, e.g., a bird, optimized its flight angles according to sensorimotoric cues such as the wind velocity, torque on the wings, or temperature. Rewarding specific combinations of vertical velocity and acceleration were found to greatly enhance the gain of height of such a glider, which paves the way for a direct transfer of these findings to aircraft technology [17]. Colabrese et al. have applied RL to the study of optimal navigation of gravitactic swimmers in turbulent flows. Those swimmers attempt to rise in height without getting trapped in the turbulent flow [18]. Durve et al. have employed multi-agent reinforcement learning (MARL) to explore how agents learn to flock [19]. Here, two scenarios have been considered. The first is learning from a teacher, while the second scenario considered agents without a teacher. In both cases, Vicsek like polar alignment of the active particles was observed [20]. Tsang et al. have demonstrated how an object of N bodies connected by extensible rods can learn to swim with the help of a Q-learning algorithm [21]. RL was also used to study the efficient swimming of fish in the wake turbulence of other fish in swarms [22].

Studies to navigate to specific fixed target location involve the work of Colabrese et al. [23]. They employed a Q-learning algorithm to train smart inertial particles to get control over their size (and inertia) to maneuver in a flow according to its local flow vorticity toward a target region with negative vorticity. Such smart inertial particles were found to outperform simple passive particles. Biferale et al. have studied the point-to-point navigation in a snapshot of a turbulent flow with an actor-critic reinforcement learning algorithm [24]. And also, more general computational studies have appeared that explore the navigation of active particles, for example, in a Mexican hat potential energy landscape [25]. Here, applying Q-learning algorithms the agents find the path taking the least time, which is a path surrounding the potential energy barrier for large barrier heights. In the case of smaller barriers, active particles may take a path straight across the potential energy barrier, which is unstable under noise in a conventional theoretical description but

becomes stable in the RL approach. Those studies may relate to the emergence of specific search strategies, e.g., in the foraging of animals, which often show specific statistics, e.g., Levy walks, though those experimental results are still under debate. Hartl et al. have combined genetic algorithms with reinforcement learning to find out about the evolution of chemotactic behaviors that are frequently observed for bacteria [26]. Bevan and coworkers have equipped artificial swimmers (mainly Janus particle swimmers) with an intelligent navigation scheme to help them conquer unknown territories [27]. In computational studies, they provided artificial microswimmers with a series of sensory inputs and actions to find strategies for swimming in an unfamiliar environment with obstacles using a reinforced learning algorithm.

While most of the applications of machine learning to the study of active particles are still computational in nature and have different goals, they share a vision that microscale synthetic active particles with sensing and information processing capabilities based on simple physical and chemical processes will be one of the next steps in artificial active matter research.

2. Synthetic active particles

Biological systems are not only endowed with a complex shape and a huge variety of different active components that enable the conversion of local energy into mechanical forces, but they also have complex chemical networks for processing environmental signals [28]. Synthetic active particles, which are of particular interest to this chapter, are very primitive as compared to their biological counterparts [29]. They break the time symmetry of the hydrodynamic equations at small length scales (small Reynolds numbers) [30] by an asymmetric structure [31] or energy release [32] and a simple propulsion mechanism. In recent years, several strategies for self-propulsion of active particles have been developed, but only a few a highlighted below. They are based on the conversion of different types of energy into a directed motion and rely on a balance of forces between the active particle and the surrounding liquid. For example, active particles have been developed that move forward due to the change in a magnetic field that causes a (chiral) microscopic structures to undulate [33] (Fig. 2A) or to rotate [34] (Fig. 2B). Chemically driven microswimmers convert chemical energy into mechanical motion.

Most of them use the catalytic splitting of hydrogen peroxide to create an imbalance in the concentration of the resulting chemical products on the surface, which drives osmotic (or electro-osmotic) fluxes that propel the particle [31,39,42] (Fig. 2E and F) or even nucleates gas bubbles that are ejected by a slightly conical tube [43]. Others use electrokinetic effects being the result of a catalytic reaction at different ends of bimetallic rods [44,45] or even more complex structures for photoelectrochemical splitting [35] (Fig. 2C). The Quincke polarization [46] provides another way to easily convert passive particles into active ones, the so-called Quincke rollers, in which a torque is created on the particle by an external electric field [47–49]. Other mechanisms involve temperature-induced changes in the composition of critical mixtures surrounding a particle.

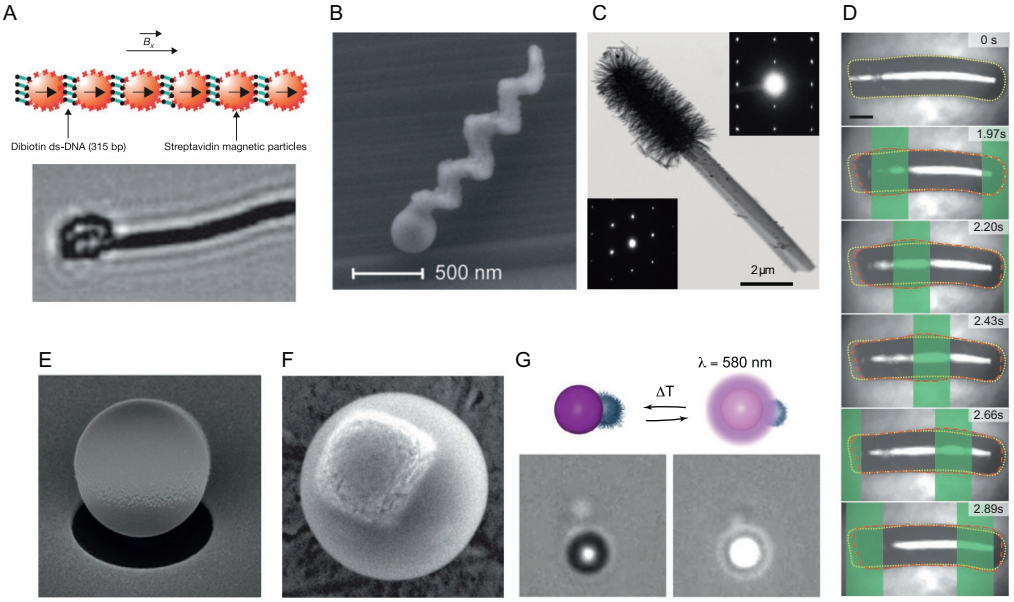


Fig. 2 Active particles. Examples of different types of active particles. (A) Chain of magnetic beads coupled by DNA strands to form a chain. The chain is connected to a *red blood cell* (gray in the print version) (image below) and actuated by an external magnetic field [33]. (B) Chiral swimmer coupled to a magnetic bead that is actuated by external magnetic fields to propel [34]. (C) TEM image of brush-like TiO_2 nanowires grown on a silicon nanowire with platinum nanoparticles at the silicon nanowire [35]. (D) Shape changing elastomeric swimmer that is illuminated by structured light to locally expand [36]. (E) Janus particle made of a polystyrene colloid (1 μm diameter) covered with a gold cap of 50 nm thickness to propel, e.g., by self-thermophoresis [37] or diffusiophoresis [38]. Gold can be replaced by a platinum capping to provide catalytic splitting of hydrogen peroxide [31,39]. (F) Hematite cube embedded in a polymer particle. The hematite cube enables photocatalytic splitting of hydrogen peroxide [40]. (G) Dye-loaded polymer particle coupled to a thermo-responsive PNIPAM-*co*-MAA microgel particle that undergoes a volume phase transition upon heating, modulating the propulsion [41]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *(A) Reprinted by permission from Springer, Nature, R. Dreyfus, Microscopic artificial swimmers, Nature 437 (2005) 862–865, Copyright 2005. (B) Reprinted with permission from Nano Letters, A. Ghosh, P. Fischer, Controlled propulsion of artificial magnetic nanostructured propellers, Nano Lett. 9 (2009) 2243–2245, Copyright 2009 American Chemical Society. (C) Reprinted by permission from Springer Nature, B. Dai, Programmable artificial phototactic microswimmer. Nat. Nanotechnol. 11 (2016) 1087–1092, Copyright 2016. (D) Reprinted by permission from Springer Nature, S. Palagi, Structured light enables biomimetic swimming and versatile locomotion of photoresponsive soft microrobots. Nat. Mater. 15 (2016) 647–653, Copyright 2016. (F) Used with permission of Royal Society, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences from J. Palacci, Light-activated self-propelled colloids, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 372 (2014) 20130372, Copyright 2014, permission conveyed through Copyright Clearance Center, Inc.*

Bringing such systems with its temperature very close to a critical point requires only a tiny amount of additional asymmetric heat release to cause a diffusiophoretic swimming [38,50]. Similarly, the interfacial tension at liquid–liquid boundaries can be perturbed by heat to yield strong Marangoni forces to drive swimmers [51]. A very general way of propelling active particles is by self-thermophoresis [37].

Here, temperature gradients generated by optical heating of a metal cap on a polymer or silica particle surface yield stresses in the liquid at the particle–liquid interface that drives the particles with interfacial thermo-osmotic flows (Fig. 2E) [52,53]. While most of the studies are carried out in water, thermophoresis provides a generic approach, since it is observed in all kinds of liquid systems. Even responsive self-thermophoretic swimmers have been realized by attaching a thermo-responsive PNIPAM microgel particle to a dye-loaded polystyrene colloid (Fig. 2G) [41]. All of those asymmetric microparticles

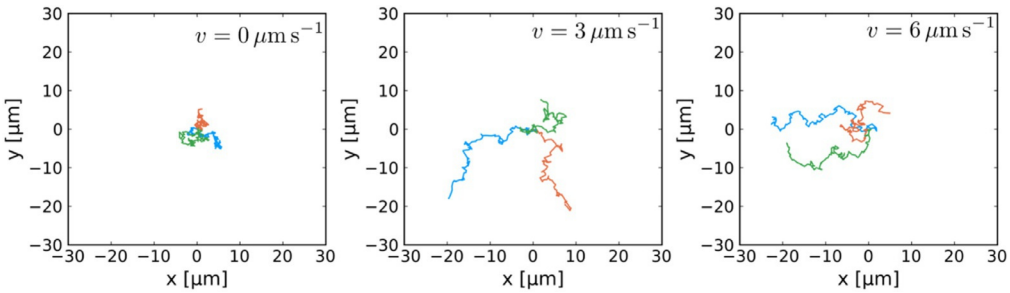


Fig. 3 Active particle trajectories. Trajectories of active particles (Janus particles) with a radius of 0.5 µm. Each graph displays three trajectories, all starting at the origin in random directions with increasing velocity from left to right.

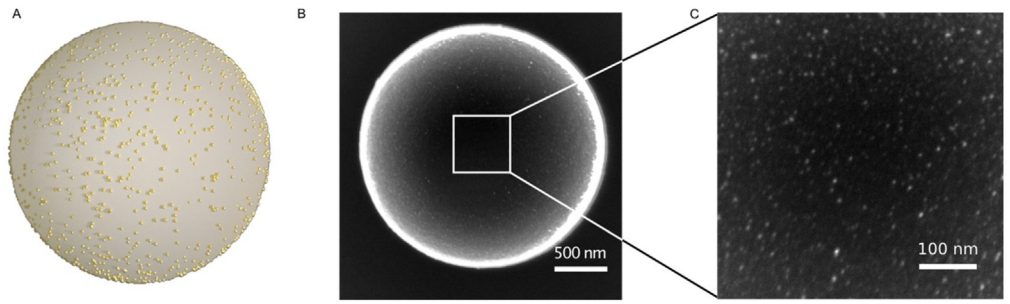


Fig. 4 Symmetric synthetic microswimmers. (A) Schematics of a symmetric microswimmer that consists of a polymer particle that is covered with 10-nm *gold* (gray in the print version) nanoparticles [32]. Panel B shows an electron microscopy image of the swimmer with a close up in (C) showing the individual gold nanoparticle (AuNP). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *(B and C) Reprinted with permission from ACS Nano, M. Fränzl, S. Muiños-Landin, V. Holubec, F. Cichos, Fully steerable symmetric thermoplasmonic microswimmers, ACS Nano 15 (2021) 3434–3440, Copyright 2021 American Chemical Society.*

carry out a directed motion that reveals on long timescales an enhanced Brownian motion (Fig. 3) as the particles still undergo an orientational diffusion or their propulsion direction [29].

Symmetric active particles can be prepared by covering a polymer particle with tiny gold nanoparticles homogeneously on its surface (Fig. 4) [32]. While the structure is now symmetric, the energy release can be made asymmetric by illuminating only the edge of the particle on one side. In this way, the particle propulsion direction becomes decoupled from the geometry of the particle and the orientational diffusion is not playing a role for the dynamics of the active particle anymore. This considerably enhances the ability to control active particles.

Besides these purely synthetic material systems, also biological building blocks are used to exploit the catalytic action of enzymes [54,55] embedded in lipid vesicles [56]. Even bacteria themselves can be modified to propel under light illumination [57–59].

3. Control of active particles

Synthetic active particles move due to local energy absorption. They can respond to simple environmental changes with changes in their propulsion speed or direction based on the physical mechanisms that drive the particles. For example, concentration gradients or intensity variations can steer the particle, modulate its velocity, produce polarization, or even cause the particle to rotate. Such behaviors can be considered a rudimentary form of chemotaxis [60–62] or even a form to communicate [63]. However, the more complex decoupling of the physical mechanisms of perception and the physical mechanisms of locomotion found in many living systems is not yet possible. Such separation in response requires more complex processing of local information, which is not yet realized in simple artificial microswimmers today. One approach to achieve such information processing and decision making nevertheless is to couple the active particle with an external feedback control mechanism [15,53,64–70]. This requires, on the one hand, an active particle that can be externally influenced at the individual level and, on the other hand, a feedback control mechanism at the interface between the microscopic observation and the response.

3.1 Light-controllable active particles

A control of active particles is on one side possible on a global scale, with each individual swimmer reacting on an external field, which is the same for all particles. In this way, the particle propulsion speed may be controlled globally by switching it on/off or adjusting its value or making it spatially inhomogeneous with the help of speckle patterns [71]. Also, external magnetic fields have been used to control the particle orientation and steer it in different directions. Although the fields used could be inhomogeneous even on length scales smaller than the microswimmers, individual control in this way is not possible and requires other approaches.

This more advanced control can be gained by illuminating individual particles separately to facilitate their propulsion. The illumination with light of a specific wavelength can enable photochemical processes [40], control proton pumps in bacteria [57], induce local phase separations by temperature [38], or just induce phoretic flows by temperature

gradients [37,53,64]. All these light-controlled systems allow for a local control of swimmers by focusing light onto small regions or even individual swimmers. This light control introduces a new concept into the study of microswimmers, namely the element of information and information exchange [66,68]. The following sections deal with the experimental implementation of the control of active particles in two dimensions.

3.2 Feedback control of active particles

Feedback control of microswimmers refers to the fact that one observes the state of the swimmer (its position X, orientation Θ, and velocity V) at a time t in a microscopy setup and responds to that by modifying its state, e.g., its velocity or orientation. In this way, the swimmers' velocity becomes a function of all observables $v_i = f(X, \Theta, V, \ldots)$. There is a growing number of experimental realizations, where feedback control of active particles has been realized. In their pioneering work, Qian et al. have shown [64] how a feedback control termed "Photon Nudging" can be used to steer a single Janus particle simply by using an intermittent propulsion based on the relative orientation of the Janus particle toward a target region. With photon nudging orientational diffusion can be harnessed for the steering process and, as it turns out, smaller particles can be more efficiently steered due to their faster orientational diffusion. Bregulla et al. have further shown that active particles can be localized with this method around a target location [53] and all details of the navigation and localization of active particles can be understood based on a statistical theory [72,73]. The technique gets in fact very valuable when studying the interactions of swimmers with heat sources or even their fundamental physics in activity landscapes [65].

Feedback control requires, in the simplest case three parts, an input signal, processing of that input signal to determine a kind of error signal, and a coupling of the input signal back to the original system. In current state-of-the-art experiments on microswimmers, the active particles are imaged in a microscopy setup with the help of a camera that is interfaced to a computer (see Fig. 5). The images are analyzed by some image processing procedures to yield the position and orientation of the particles, which then also allow to determine the dynamic quantities like the velocity. A desired response is calculated by an algorithm and executed by controlling the supply of energy to the individual particles at certain positions. The latter is carried out by beam steering optics that involves either spatial light modulators, digital mirror arrays, or acousto-optic modulators, where we detail some information on these devices below.

3.2.1 Experimental realization of feedback control

The sections below describe some technical details for the individual beam steering solutions.

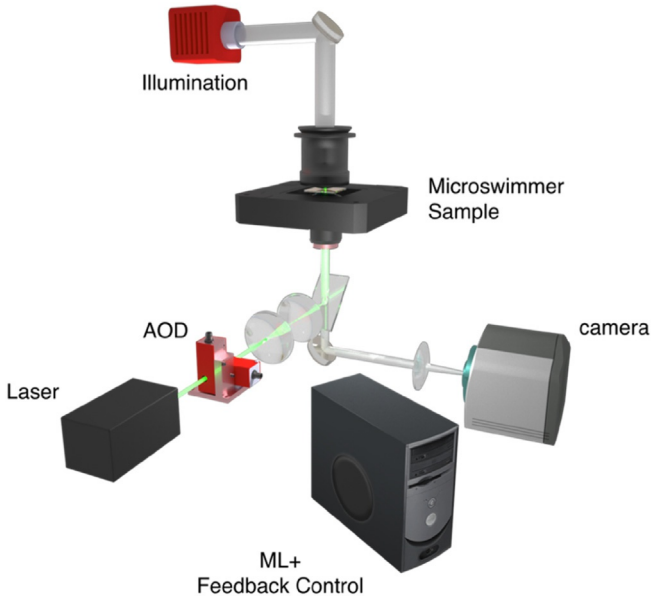


Fig. 5 Experimental setup. Typical setup to control the dynamics of single and multiple active particles by feedback control. The sketch depicts a dark-field microscopy setup illuminating the sample with swimmers from the top with a darkfield condenser. The scattered light from the sample is imaged to a CCD camera. The images are analyzed at video rate with the help of an image processing software in a computer. The computer calculates a response to control the deflection of a laser beam in the sample plane with an acousto-optic deflector (AOD).

a) Beam Steering

Beam steering can be carried out with different devices. Frequently, this comprises either an acousto-optic deflector, a spatial light modulator, or a digital mirror device, which all have their specific advantages.

- Acousto-optic deflector: The AOD is a device driven with acoustic waves in crystals, where the frequency of the acoustic wave (around 100 MHz) creates an optical grating deflecting the beam to different diffraction orders. By tuning the acoustic frequency, one tunes the grating constant and thus the deflection. For 2D beam steering, one needs two orthogonal deflectors, where deflection angles of several degrees are achieved. Their advantage is a very fast response (microseconds). The disadvantage is that for multiparticle control, the particles must be multiplexed meaning that only one particle is active at a time.
- Spatial light modulator (SLM): An SLM is essentially a liquid crystal "display" allowing to introduce controlled local phase shifts to the light wave to modify the intensity pattern that is generated in an image plane. The phase pattern to be induced is commonly generated by an iterative procedure (Gerchberg–Saxton [74]) or meanwhile even by

machine learning approaches [75]. This could be in complex cases time-consuming and require a GPU if fast-pattern generation is required. It can easily create hundreds of foci to control active particles individually. Using a GPU pattern generation within several 10 ms is possible. All foci are, contrary to the AOD, active at the same time, so all particles will be propelled at the same time. The speed of the SLM is comparable to typical frame rates (100 Hz) of commonly used camera systems.

- Digital mirror device: A DMD contains an array of multistable mirrors. The light power falling on the mirror surface is, therefore, split into power contributions of each individual pixel that are spread across the image area in the focal plane of the objective lens. As compared to an SLM, where all incident light power is mainly distributed across the focal spots, the light of the pixels of the DMD in the off state is just lost. Speed-wise, the DMD is comparable to an SLM.

b) Real-time image processing

In the simplest case, image processing requires determining the position and/or orientation of all active particles in the sample. This can be done in various ways. Image processing can be performed on special image processing cards, on graphics cards, or in the simplest case, only by the CPU of the computer. Since hardware control is often carried out with LabVIEW, NI Vision provides a powerful module for image processing using centroid-based particle localization methods or more advanced techniques like Hough transforms. Simple centroid-based particle detection can be carried out on timescales below 10 ms for an image size of 512 px × 512 px, even for larger amounts of particles. For example, thresholding an image of a single Janus particle (see Fig. 6) at least two different thresholds provides the location of the particle for the lowest threshold and the location of the metal cap of the Janus particle for the high threshold. Using both positions,

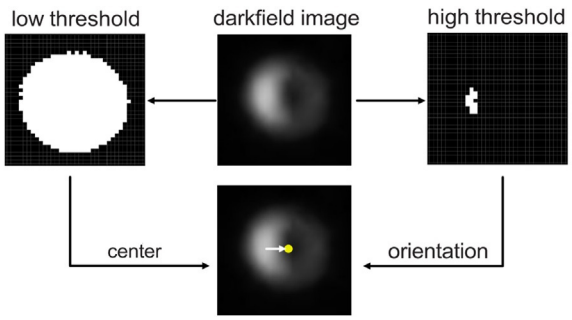


Fig. 6 Image processing for Janus particles. Simple position and orientation determination for single Janus particles. The darkfield microscopy image (*middle*) binarized by two different thresholds to obtain the position (low-threshold intensity, *left*) and the in-plane orientation (high-threshold intensity, *right*). First, the position is calculated. The high threshold image provides the position of the metal cap and yields with the difference to the *center*, the orientation of the Janus particle.

the in-plane orientation of the particle can be determined very quickly. Using in addition the intensity of the particle, even the three-dimensional orientation of the particle can be determined.

When images contain more complex objects and are heterogeneous in the object's intensity, more advanced techniques, e.g., involving deep learning [12,15] approaches, are advantageous but may need optimization and the application of GPUs to match the video rate of the experiment to carry out real-time feedback control. Fig. 7 shows a solution reported in reference [15]. It uses a single-shot convolutional neural network based on the YOLO architecture [76] to localize and classify objects in microscopy images at video rate (100 Hz). Therefore, individual objects may be addressed in different ways if required.

Currently, systems in use at the authors lab are capable of taking images at a framerate of 100 Hz. The YOLO-based tracking and classification solution is used with a custom-made interface to a GPU for LabVIEW to comply with these available framerates (Fig. 8).

Methods like deep reinforcement learning with real-time experience replay can be rather demanding in data storage, especially if the original images are used for experience replay. Current applications as demonstrated in the sections below use the detour of first analyzing images in real time. The processing of the image information provides the freedom to introduce in a less costly way any type of correlation of the input signal with the output signal. For example, a correlation of the propulsion of the particle with the orientation of the particle in the lab frame can be easily achieved. It is also possible to correlate the propulsion velocity with the position in the lab frame. As the brain of the

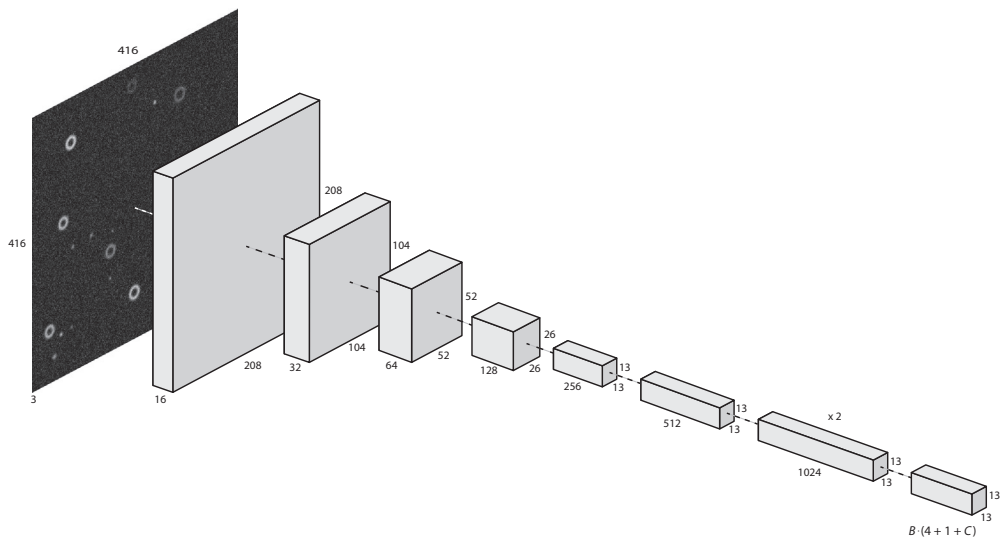


Fig. 7 Single-shot CNN for real-time image classification. Structure of the convolutional neural network used for the localization and classification of objects in microscopy images in real time [15]. The structure is based on the YOLO2 structure [77].

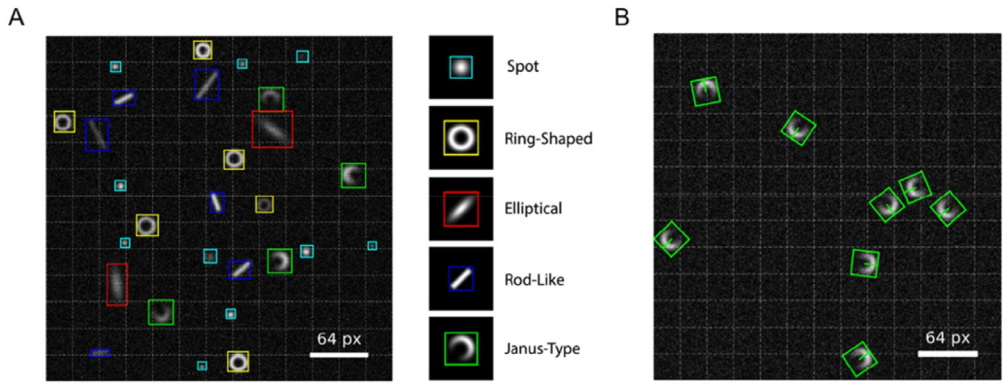


Fig. 8 Localization and classification by a CNN. (A) Localization and classification of different shaped particles in a synthetic microscopy image. The objects have different intensities. The grid indicates the image segmentation into 13×13 grid cells as done by the YOLO2 network. Each grid cell delivers a set of identified objects and its positions. (B) Extension of the YOLO2 CNN to detect the orientation of objects in microscopy images. The picture shows the orientation detected for synthetic images of Janus particles in darkfield microscopy [15].

particle is the computer, one may also give each particle a memory to carry out actions based on past observations or future extrapolations. Therefore, one has to keep the identity of the particle over the experiment. Using a trajectory linking, one is able to provide a long memory to each particle to introduce delays or forecasts [78]. There is immense freedom in this choice, which makes active particle control a very powerful method to study various processes.

4. Reinforcement learning

Learning is the way active systems develop models of causality about the elements they find in their environment [79]. Such models can vary in complexity depending on how active systems manage to obtain information about such environments. However, once these models are stored, they can be used by active agents to trigger the desired behavior in a given situation. The way active agents perceive and process such information is important for a formal description and classification of the learning process performed by a system. While it is possible to develop models of a system only by using information that can be acquired directly through interactions with the environment, it is also possible to develop knowledge only by processing data that have been acquired elsewhere and represented in an optimal way for its transmission and processing. The first case is probably the most intuitive way to develop knowledge about how to approach a problem and is the core idea of reinforcement learning.

Reinforcement learning (RL) is a topic of machine learning based on a specific mathematical approach to process control, the Markov decision process (MDP). It has been proposed for modeling decisions in various biological and artificial systems and is applied to problems for different purposes such as robot manipulation, game control [80], natural language processing, or energy management [81]. Inverse RL based on hidden Markov models is also a widely studied topic. It is used to gain information about the rules it follows that trigger observable behavioral sequences. Applications of inverse RL are found, e.g., in genomics and protein dynamics in biology as well as in speech or gesture recognition and even music structure analysis.

To formalize a MDP and consequently define an RL framework, there are two essential elements: an agent and an environment including a target. These elements can each be associated with observables that capture the information that needs to be correlated with specific actions through an interaction with the environment. These are a set of actions a, a set of states s, and a reward function $r(s)$, respectively. The first describes what an agent can do to gain experience from the environment in each state, while the second describes how the environment is organized to quantify the variations that occur after the agent performs an action (Fig. 9). The third is the signal, which essentially gives the agent an intuition about how good or bad it was to perform an action in each state. This allows the definition of a function called policy $\pi(s)$ that provides the best action for each state.

A key feature of MDPs and RL is that they are designed to provide solutions even under stochastic influences in the relationship state–action, typically quantified by a transition function T. This function models a family of probability distributions that characterizes the outcome of any action in any state prior to the knowledge of the agent of future events. If a system knows the transition function, it can estimate how good or

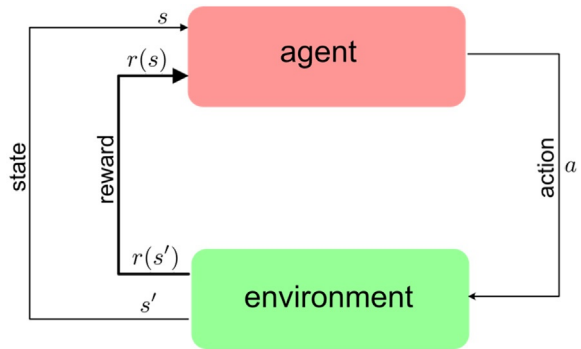


Fig. 9 Reinforcement learning loop. An agent being in a state s with a reward $r(s)$ takes and action a to explore the environment and a new state s'. The transition to the new state results in a reward $r(s')$. The agent starts over with this reward and state taking the next action. This reinforcement learning loop is repeated until an episode ends when arriving at a goal state or any other episode ending event occurs.

bad the application of an action is in a state by analyzing the consequences of all possible transitions and their probabilities. To formalize this, a value function U is defined as

$$U^*(s) = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_i\right).$$

Here, * denotes the optimal value function, which maximizes the expected discounted rewards as indicated by the $\max_{\pi}$ E. The maximum is selected from all possible policies π. The factor γ in this equation is the so-called discount factor. It modulates the importance of future events in the process. In the above equation, the discount factor appears in such a way that the rewards obtained in future steps are discounted geometrically, i.e., the best reward is the immediate reward. Nevertheless, the way future rewards are discounted can depend on the model of optimality considered. Three models of optimality can be used: the finite horizon, infinite horizon, and average reward model. Here, we only consider the infinite horizon situation where the geometric expression is used for the expected value of the reward r. One can extend the expression of the optimal value by writing the expected value of the reward using the transition function $T(s, a, s')$ to yield

$$U^*(s) = \max_{a} \left[r(s, a) + \gamma \sum_{s \in S} T(s, a, s') U^*(s')\right],$$

which is the Bellman equation, the foundation of dynamic programming. Thus, each state s has an action a with a maximum value depending on the possible next states s', the probability of transitioning to this state $T(s, a, s')$, and the next states maximum value $U^*(s')$. The $\max_{a}$ in the equation reveals the nonlinearity of the problem whose solution is the policy function π. Once an RL algorithm has converged, it yields the optimal policy

$$\pi^*(s) = \operatorname*{argmax}_{a} \left[r(s, a) + \gamma \sum_{s \in S} T(s, a, s') U^*(s')\right]$$

for which the agent maximizes its total reward. Note that the transition function provides the information on the transition from state s to s' with the action a. In a real-world application like the one described below for a single microswimmers, this function contains the "physics" of the problem, i.e., it describes the transition probability given a certain action.

The value function within a MDP can be also expressed as a matrix that stores the value associated with an action a in each state s. This matrix is usually called Q-matrix and is connected to the optimal value function by

$$U^*(s) = \max_{a} Q^*(s, a).$$

The Bellman equation, therefore, results in

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s \in S} T(s,a,s') \max_a Q^*(s',a'),$$

which is the basis for the so-called Q-learning. Whenever the transition function $T(s,a,s')$ is known, the Q-learning learning is *model-based* and the optimal policy can be obtained, for example, with the help of a value iteration procedure [79]. When the model or the detailed physics for the transitions is unknown, the temporal difference approach has been revealed as a powerful tool for solving strategy search problems in real systems as *model-free* Q-learning. Here the agent does not need to know beforehand what the model of the system is, and the information is propagated from the goal backwards with each transition to a new state. There is also no need to wait until the end of a learning episode, as the Q-matrix is updated during each transition, which makes Q-learning a more tractable implementation of temporal difference learning (TD-learning) for a real robot.

The optimal value function, Q-matrix and correspondingly also the optimal policy is commonly obtained in an iterative procedure, where the tabular Q-matrix for a given state and action $Q(s,a)$ is updated based on the value of the best action in the state s' after an action a has been taken. For Q-learning, this update rule is

$$Q(sa) := Q(sa) + \alpha \left[r + \gamma \max_{a'} Q(s'a') - Q(sa) \right].$$

The parameter α hereby denotes the learning rate, which defines the speed at which updates are incorporated into the Q-matrix. Note that the Q-learning algorithm is considering all actions a' of the following state s' by using $\max_{a'} Q(s',a')$ and is, therefore, called an *off-policy* method.

Another frequently used algorithm to learn the value function, which is called SARSA for *state–action–reward–state–action*. The update scheme of the *state–action* function $Q(s',a')$ is given by

$$Q(s,a) := Q(s,a) + \alpha(r + \gamma\, Q(s',a') - Q(s,a)).$$

The SARSA algorithm is, therefore, an *on-policy* method. *On-policy* methods estimate the value of a policy while using it for a control of the actions. In *off-policy* methods, the policy is effectively split into a behavior policy to generate behavior that may be unrelated to the policy that is estimated. In all the updated procedures, an element of randomness is commonly introduced to favor exploration of the environment over exploitation. In this element, called ϵ-greedy method, a random action is selected with a probability ϵ rather than the action defined by the Q-matrix value.

The first experimental demonstration of a microswimmer learning to navigate using strategies of RL was shown by Muiños-Landin et al. using a Q-learning update algorithm. The control of the experimental system and the detailed understanding of the underlying physics in this study allowed a direct connection of the model-free learning approach to a model-based numerical evaluation as briefly described below.

4.1 Reinforcement learning with a real microswimmer

Feedback control, as described above allows to correlate the dynamics of a microswimmer with all kinds of variables, such as position or orientation. While these rules are static, i.e., predefined in the control program, machine learning as RL provides the ability to have dynamic rules, which shape while exploring the environment with the swimmer. The goal of incorporating a machine learning algorithm could be to learn a specific strategy for navigating in an unknown environment [16-18,23,82–85] or explore what is required to mimic a certain biological behavior like swarming [19]. The experiments described below (and more extensively in Ref. [86]) refer to the simplest navigation problem for reinforcement learning.

The swimmer explores a board of square cells (see Fig. 10), the gridworld, to reach a goal state (green cell). While exploring this gridworld, the swimmer collects rewards. It learns to find a spatially fixed target rather than a strategy how to navigate. Each cell

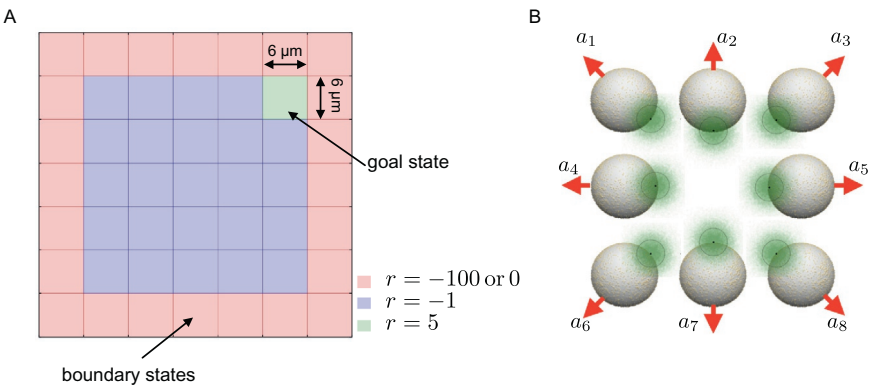


Fig. 10 Setup for reinforcement learning with single microswimmers. (A) The gridworld defined as virtual overlay to the sample region consists of 25 central states (*blue* (gray in the print version), reward $r = -1$) of size of $6\,\mu m \times 6\,\mu m$ with a goal state (*green* (dark gray in the print version), reward $r = 5$) and boundary states (*red* (dark gray in the print version), reward $r = -100$). (B) The symmetric microswimmers can carry out eight actions ($a_1 \ldots a_8$) by placing the laser at the indicated position and heating the gold particles are the surface of the polymer particle (see symmetric swimmer above). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.) *From Science, Robotics, S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021) eabd9285. Reprinted with permission from AAAS.*

represents a state s the swimmer can occupy and in each state the swimmer is able to choose between eight actions ($a_1 \ldots a_8$) as shown in Fig. 10B. The value of each transition between a state s and s' is denoted by the rewards that are taken to be just a function of the state $r(s)$. As the goal is to reach the green target cell, the reward for transitioning to this goal state is positive, while all transitions between the blue cells are giving a slight negative penalty. To reinforce in addition a confinement inside a certain region, additional boundary states have been implemented, which carry a strongly negative penalty (red cells in Fig. 10). Using this very general setup, Muiños-Landin et al. have used experimental microswimmers in a real sample arena to learn to navigate a virtual gridworld [86]. The main difference to the standard gridworld problem is the fact that microswimmers undergo Brownian motion; thus, their directed motion is noisy in position and orientation. This is especially important as the particles move within each state for some period until they reach the state boundary to transition to the next state. This contrasts with the common gridworld problem, where the execution of one action is causing a direct transition to a new state. A second difference is the discretization of the motion processes. The motion and its consequences are observed at discrete instances of time, e.g., with a camera snapshot.

The reinforcement learning loop (Fig. 11) in the experiment, thus, proceeds in the following way:

1. Determine the state: Snapshot of the sample area is taken in the microscopy system, e.g., a dark-field microscopy image is taken. The location of the swimmer is analyzed by one of the above-mentioned methods and the state in which the swimmer resides is determined.
2. Determine the action: The action is then determined from the Q-matrix by finding the maximum value of the Q-matrix for the current state.
3. Make a transition: The action a is applied until the swimmer reaches a new state. This is required as the microswimmer only moves a distance on the order of $\Delta r \approx v\Delta t$, where Δt is the inverse frame rate. The action is, therefore, not instantaneous but requires a certain amount of time.
4. Update the Q-matrix: Once a new state s' is reached, the Q-matrix is updated with the help of the reward according to the Q-learning update rule specified earlier.

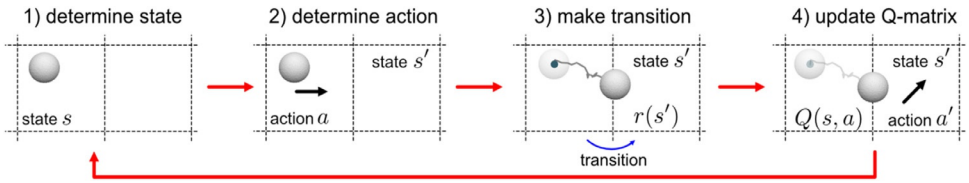


Fig. 11 Experimental reinforcement learning loop. *From Science Robotics, S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021) eabd9285. Reprinted with permission from AAAS.*

After these four steps, the RL loop starts again until the goal state or a boundary state is reached. In this case, we end an episode of the RL and start with the swimmer at a new random position in the gridworld. With the help of the Q-learning update algorithm, the Q-matrix of the system develops from the completely randomly chosen entries and corresponding policy as displayed graphically in Fig. 12A to a well-structured matrix with a converged policy in Fig. 12B. The convergence of the algorithm can be verified by the sum of the Q-matrix entries over time, which converges. Also, the number of transitions between states required to reach the goal state decreases with increasing learning progress as revealed by the right graph in Fig. 12C.

This convergence is the consequence of the interaction of the swimmer with the local environment (the rewards) that yields with each learning episode a better approximation of the optimal policy to reach the target from any point in the gridworld. This evolution of the policy is visible in the trajectories of the swimmer as shown in Fig. 13.

The experimental demonstration of RL with an active particle shows that learning in a direct physical environment under the noisy conditions of Brownian particles and with all the imperfections of the real sample is feasible, even if the learning processes is lasting over several hours. The results do also indicate that new effects appear in real-world scenarios as compared to simulations, which emphasizes the so-called "reality gap" between strategies obtained purely in a computer as compared to those in the experiment. The microswimmer is subject to a specific noise that appears as a result of an unavoidable time delay between sensing and action and Brownian motion. Such a delay refers to the finite bandwidth of feedback loops and results in the considered experiment in an uncertainty of its propulsion direction, even if the action is well defined. The consequence of this uncertainty in the propulsion direction is an optimal speed of the particle that leads to an optimal survival when swimming toward a goal area [86]. The noise in the system thereby alters the strategy. Understanding this unexpected influence of noise and measuring the transition function $T(s, a, s')$ finally also allows a modeling of the policy by the model-based approaches discussed before.

The study presented by Muiños-Landin et al. also highlights the examples where swimmers navigate around virtual obstacles (Fig. 14A) or share their information in a collective learning process. Here a single Q-matrix is updated by two particles at the same time, i.e., the two swimmers are sharing their information to yield an expected faster learning (Fig. 14B).

4.2 Deep reinforcement learning with a single active particle

In the real world, state and action spaces are commonly continuous and high dimensional. Q-learning is implemented by finding a low-dimensional representation (e.g., a set of discrete regions as in the gird world) of complex environments thereby bypassing this *curse of dimensionality*. Although Q-learning is a robust strategy to navigate in this

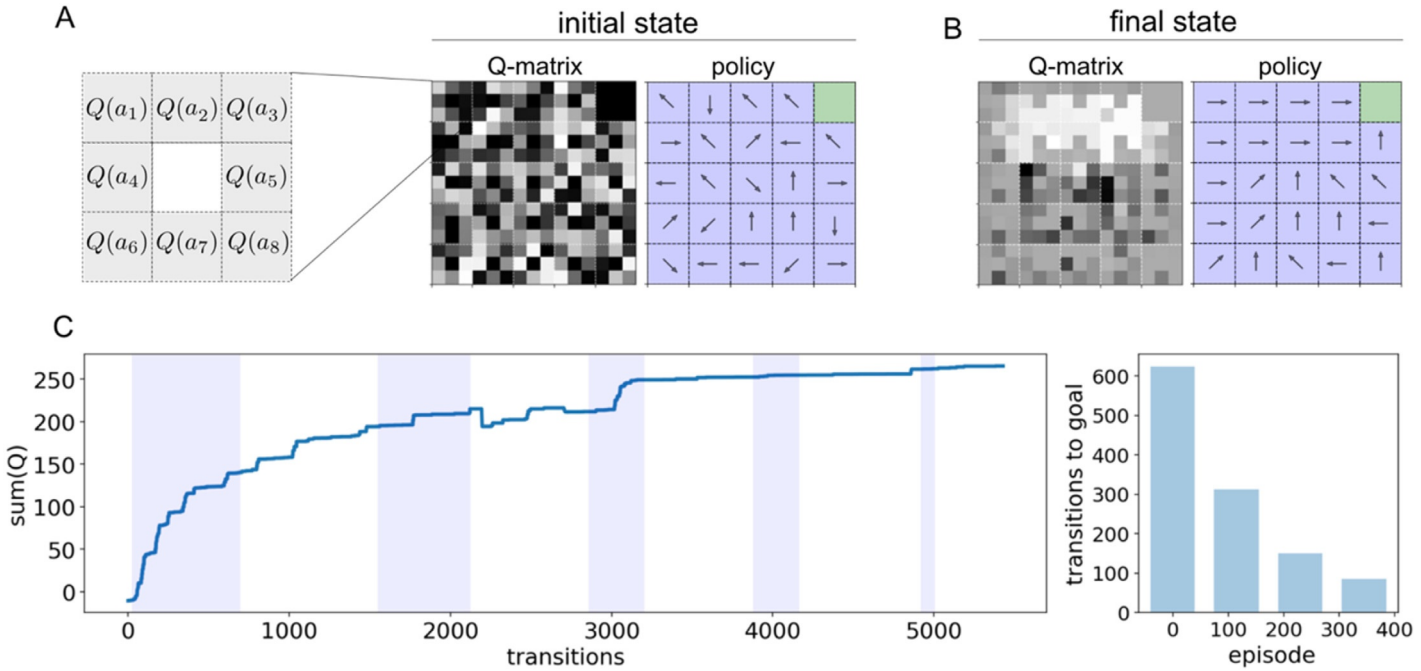


Fig. 12 Learning process. (A) Graphical representation of the Q-matrix and policy at the beginning of the learning process. The Q-matrix values in each state are represented by the gray values as indicated in the sketch on the left. Brighter values correspond to higher Q-values. (B) Graphical representation of the Q-matrix and the policy after the learning process. (C) Sum of all Q-matrix values during the learning process indicating that the algorithm converges. The *blue-shaded* (dark gray in the print version) time period corresponds to 25 episodes. The graph on the right demonstrates that the number of transitions required to reach the goal decreases with progressing learning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *From Science Robotics, S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021) eabd9285. Reprinted with permission from AAAS.*

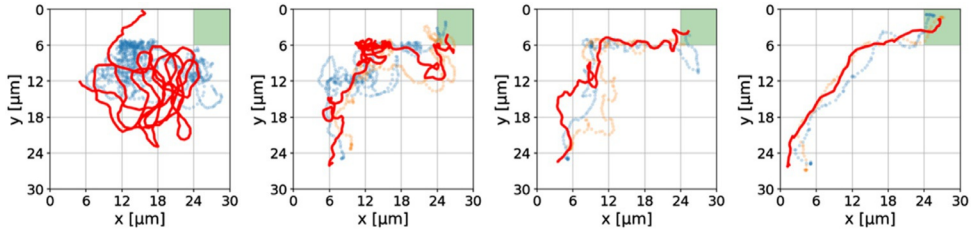


Fig. 13 Trajectory evolution. Evolution of the active particle trajectories during the learning process. At the beginning of the process (*left*), the active particle motion is still determined by the random actions. With progressing learning, the trajectories get straighter and finally only determined by the Brownian motion of the microswimmers. The trajectories are taken after 1 episode, 100 episodes, 300 episodes, and 600 episodes of learning. The dotted trajectories are additional trajectories to indicate the variation of the trajectories at a specific times. *From Science Robotics, S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021) eabd9285. Reprinted with permission from AAAS.*

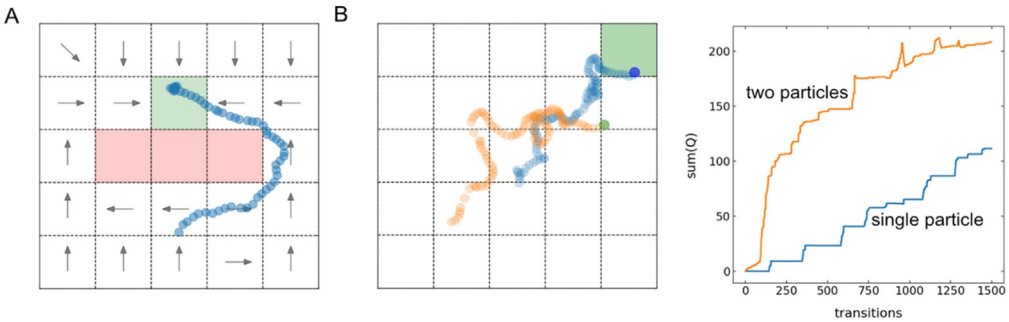


Fig. 14 Reinforcement learning with virtual obstacles and multiple particles. (A) Policy learned by a single active particle in a virtual gridworld overlayed to a real-world sample. The gridworld contains a virtual obstacle (*red* (gray in the print version)) that is implemented with a high negative reward ($r=-100$) and the goal state (*green*) next to the obstacle with ($r=5$). (B) Reinforcement learning with two active particles using the same Q-matrix. This "knowledge sharing" leads naturally to a faster convergence of the learning process as displayed in the right graph. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) *From Science S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021) eabd9285. Reprinted with permission from AAAS.*

region of square cells, the algorithm gets more resource hungry and fails to scale when presented with a closer to the real-world representation of the environment. Simply a denser grid or a finer set of actions restricts the agent from learning effectively, let alone the possibility to handle continuous domains or the real-time pixel data. To deal with such limitations, one then typically eludes from a table to represent the value function and instead approximates the same with an artificial neural network (ANN). ANNs are popular for their implicit similarity to the human brain and are known to outperform

all existing algorithms at finding patterns in abstract multidimensional data. The paradigm of deep reinforcement learning combines this ability of ANNs with reinforcement learning to produce exceptional results at solving complex decision-making tasks [80,87]. An ANN of the type referred here can be thought of as a collection of artificial neurons, each holding a number called their "activation." Generally, they are arranged in a layer-by-layer fashion, interconnected to their likes in the next layer by a series of weights ($\mathbf{W}^{ii}$) and biases ($\mathbf{C}^{i}$). The activation of a neuron in the successive layer is from the linear combination of activations of the neurons in the previous layer followed by a non-linear activation function (Fig. 15). Once the weights and biases are optimized, the ANN can potentially approximate abstract multidimensional mappings including the optimal Q-value function for the navigation task at hand. This approach, where an ANN is used to predict the Q-value of a state–action pair, is called deep Q-learning (DQL) [87].

Instead of discretizing the state space, which effectively adds noise and causes delayed reinforcements within the learning process, DQL can handle representations of the state of the microswimmer as a continuous vector. The action set, however, was kept discrete as in the Q-learning approach but reduced to fewer movements: up, left, down, and right. Since the authors' experimental setup could image and act on the microswimmer at a fast enough rate, actions can be considered to be quasi-continuous. The task at hand

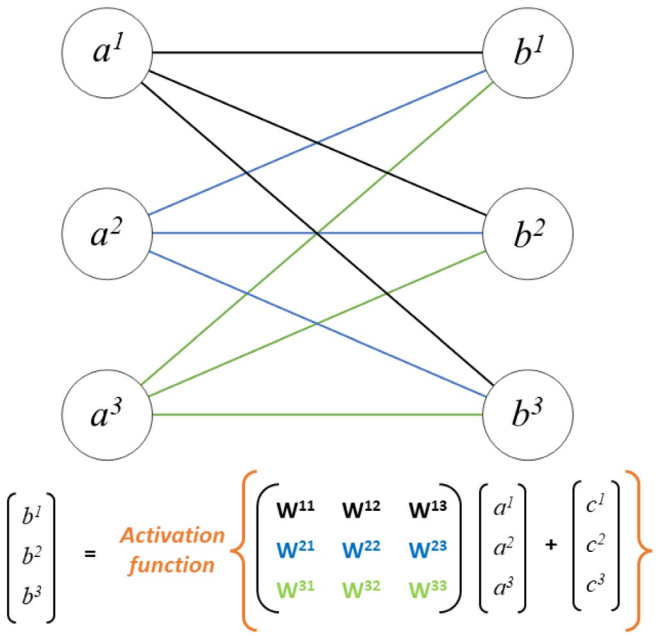


Fig. 15 General scheme for the activation propagation in the ANN. The formulation of activations in the successive layer are from the linear combinations of activations from the previous layer combined with a non-linear activation function.

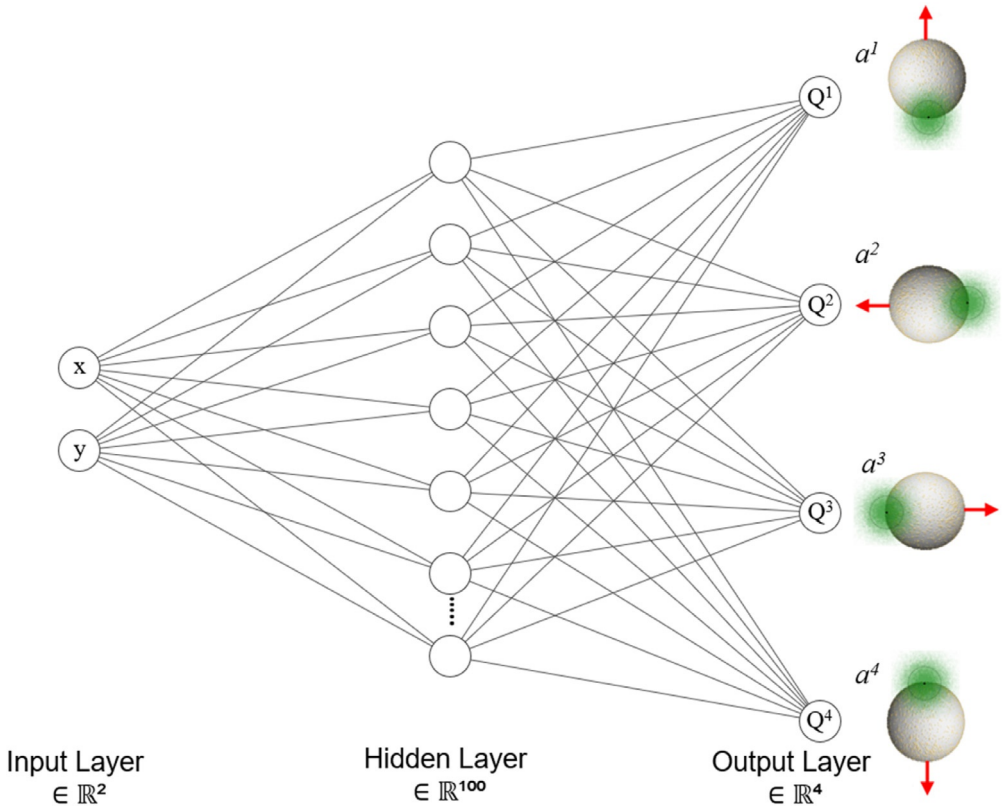


Fig. 16 ANN architecture used in the DQL implementation. The ANN takes in the two-dimensional position of the particle (*x*, *y*, state) and outputs Q-values corresponding to each possible movement (action) of the microswimmer. In each iteration, the loss is calculated as the difference between the current output values and the same predicted by the *Bellman equation*. The weights and biases are then adjusted to minimize this loss by running a gradient descent step.

for the microswimmer is to navigate to a small region at the center, starting from any random point in the playing field. To approximate the optimal Q-value function, a modest 3-layer ANN was chosen (Fig. 16). The network consisted of two nodes at its input layer accepting the spatial coordinates of the microswimmer followed by a hidden layer of 100 nodes and finally an output layer with four nodes corresponding to each action, all separated by a hyperbolic tangent activation function. The reward scheme was also redesigned. Each state was assigned a reward value that scaled with its distance to the goal region located at the center of the confining boundary.

In the barebones DQL algorithm, the training begins with initializing the ANN with random weights. At each time step, the agent, the microswimmer, undergoes a state transition by performing one of the allowed actions, receiving a reward as a consequence. The preceding state is then fed into the network to predict Q-values corresponding to each possible action. These Q-values are compared with target Q-values defined

by the *Bellman equation* where a loss is calculated [87]. The network parameters are then adjusted to minimize this loss by running a gradient descent step. Calculating gradients for all the parameters of the network is indeed possible and the process is called backpropagation. These steps are repeated until the microswimmer reaches a terminal state, either the goal region or outside the confining boundary, both of which mark the ending of an "episode." Ideally, over several episodes of training, the ANN converges toward an optimal value function that maps the current state of the microswimmer to a set of four Q-values that correspond to each of the allowed actions. The optimal policy is then derived by choosing the action with the greatest Q-value at any given state. However, like in most cases, a few more additions from the state of the art were required to ensure stable training and convergence of our ANN. Since DQL estimates a Q-value function instead of an explicit policy, the sequence of the inputs is irrelevant. This allows the agent to learn not only from the consequences of its immediate action but also from previous ones, just like in the case of biological counterparts. This technique termed *experience replay* improves the training performance by feeding the ANN with inputs sampled from a batch of past experiences instead of just the immediate one, randomizing and reducing correlations between the same [80]. On another note, using a less frequently updated copy of the ANN to calculate the loss was also helpful in stabilizing the training [88].

In the authors' experiment, the algorithm was implemented with Python which ran within a custom LabVIEW program. The PyTorch library was used to create and train the ANN, while LabVIEW took care of the hardware side of things. Optimal performance was achieved at 300 episodes equaling about 3 h of training. The microswimmer was able to navigate to the target irrespective of where it started within the confining boundary (Fig. 17). A closer look at the learned policy revealed reproducible differences

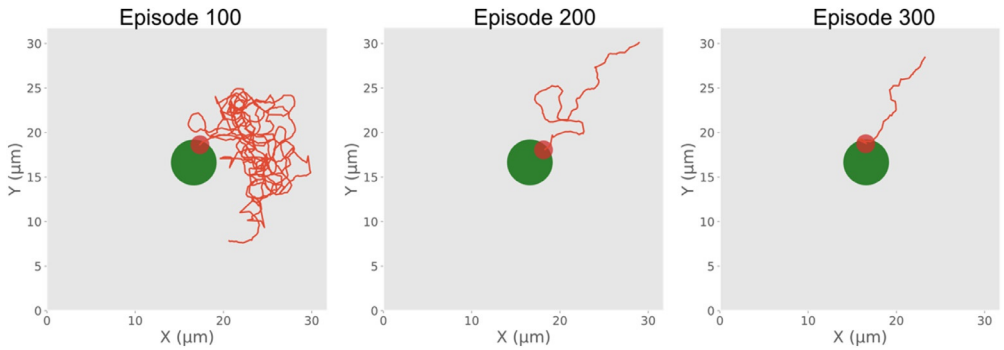


Fig. 17 Evolution of the trajectories during the DQL of a single microswimmer. The trajectories taken by the microswimmer to reach the goal (*green* (dark gray in the print version)) region during subsequent phases of the training. By episode 300, the microswimmer was able to navigate itself directly to the goal region regardless of the location it began. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

compared to their initial simulations hinting toward true adaptive behavior. For example, minor imperfections in the calibration of the steering mechanism produced an asymmetry in the propulsion of the particle, for which the algorithm aptly compensated. Moreover, defects arising from the handling or storage of the microswimmers, irregularities in their geometry, or hydrodynamic interactions of the particles with glass surfaces, all of which collectively constitute a "reality gap," can also create uncertainties in propulsion. Although undesired, these inevitable imperfections served as an excellent showcase for the potential of deep reinforcement learning in steering the active particles which equipped them to adapt and overcome such challenges.

4.3 Multi-agent systems

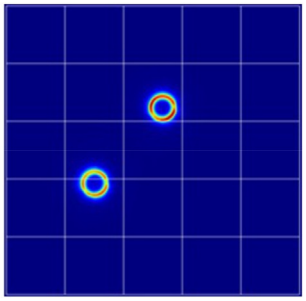
In robotics, planning is important at a single-agent level but especially critical when it comes to multiple agents. A single-agent approach is more focused on the modeling of actions and transitions in order to enable the selection of an optimal sequence to complete a task. But the complexity at the multi-agent level comes on the one hand from the impact of one agent on another and its temporal dependency. On the other hand, the potential simultaneous execution of actions by multiple agents makes it difficult to model any transition of the system. These problems, and the fact that in many real cases there is a remarkable influence of stochasticity or that the processes are only partially observable, make it clear that planning at the multi-agent level is a problem that cannot be solved unambiguously. However, through various simplifications or assumptions, a solution becomes possible in real systems, and agents cooperate to solve problems. This section gives some indication of how this happens in microscopic artificial multi-agent systems. Since RL was presented as a success story for the single-agent case, we present here the basic transfer from a single-agent situation to a multi-agent system (MAS). This will highlight the difficulties in implementing a MAS based on RL.

4.3.1 Multi-agent reinforcement learning (MARL)

All concepts presented so far were introduced for reinforcement learning of a single agent. When more than one agent is present in the environment and tries to maximize the expected reward, the problem has to be redefined [19,89].

This redefinition can be done by considering stochastic game theory, which is a generalization of MDP. Consider the simplest case of a pair of symmetric swimmers navigating in a grid world (see Fig. 18), with the set of actions as previously defined for a single float. In this context, a so-called payoff matrix is defined for each of the particles, which plays the role of a reward in a first step. The values of this matrix assign values to each action performed by each particle, which depend on the actions that the other particles can perform. We analyze two main situations. The first is the so-called zero-sum game. Here the interest of each particle is opposite, i.e., if the payoff matrix of each particle is (M_1, M_2) and the first particle's reward is the other particle's loss a single matrix M is

A



B

M_1 and M_2 payoff matrix for agent 1 and 2

M_1

	a_1^2	a_2^2
a_1^1	1500	50
a_1^1	2000	60

M_2

	a_1^2	a_2^2
a_1^1	2000	4000
a_1^1	100	200

Fig. 18 Multi-agent reinforcement learning. (A) Two active particles in the gridworld in an experiment. (B) If agent 1 decides on action 1, the decision taken by agent 2 can lead to a minimum of its payoff. While if it decides 2 whatever agent 2 makes it does not get the minimum. The same happens for agent 2.

sufficient to describe the two agents as we have $M_1 = M$ and $M_2 = -M$. This situation is also called the single matrix game. The second situation will be the general-sum game where the sum of M_1 and M_2 is not zero. This situation is also called the bimatrix (or multimatrix) game and will be considered in the following.

For the multimatrix game as shown in Fig. 18, a payoff matrix is defined by M_1, M_2 for each particle. Each particle's decision is then taken into account when considering what might happen with each of the other particle's possible decisions. This way both particles will find out that there is an action a^i that causes the payoff to become maximum whatever the action taken by the other particle is. This is the so-called Nash equilibrium. In a Nash equilibrium, each agent agrees with his strategy choice, even in retrospect he would make it the same way again. This Nash equilibrium is formalized as follows:

$$r^1\left(a^1, a^2\right) \geq r^1\left(a^1, a^2\right) \quad \forall a^1 \in A^1$$
$$r^2\left(a^1, a^2\right) \geq r^2\left(a_\star^1, a^2\right) \quad \forall a^2 \in A^2,$$

where A^1 and A^2 are the set of actions of each particle and $r^k(a^1, a^2)$ is the corresponding payoff value. However, such a definition for the Nash equilibrium is only valid for a pure or deterministic system. For a system where stochasticity plays a role, the definition of the Nash equilibrium is as follows:

$$\rho_\star^1 M^1 \rho_\star^2 \geq \rho^1 M^1 \rho_\star^2 \quad \forall \rho^1 \in \sigma\left(A^1\right)$$
$$\rho_\star^1 M^2 \rho_\star^2 \geq \rho_\star^1 M^1 \rho^2 \quad \forall \rho^1 \in \sigma\left(A^2\right),$$

where $\sigma(A^i)$ is a set of probabilities over an action space A^k. For any $\rho^k \in \sigma(A^k)$, the sum

$$\sum_{a \in A^k} \rho^k = 1$$

is the expected value of the payoff of the particle 1 under the situation where both particles adopt the mixed strategy ρ^1, ρ^2. This solution for the problem, the Nash equilibrium, defines the so-called the optimal strategy in the multi-agent problem.

$$\rho^1 M^1 \rho^2 = \sum_{a^1} \sum_{a^2} \rho^1\left(a^1\right)\, r^1\left(a^1, a^2\right)\rho^2\left(a^2\right).$$

Now with this concept of Nash equilibrium and strategy in mind, we can define our system as a 6-tuple.

$\langle S, A^1, A^2, r^1, r^2, p\rangle$, where we find the set of states S, the set of actions A^k for the particle k, the payoff p^k for each case, and p the transition probability map. Note that from now on, the concept of state is no longer the fact that a particle is placed in a cell. Instead, the state is now the joint state of one particle being in a specific cell and the other particle in any other possible cell. This will substantially increase the difficulty of the calculations.

For simplicity, we can consider a situation where the active particles do not physically interact. This is helpful because otherwise the transition function of the system is no longer stationary and the MDP cannot be applied. In our experimental system, this is satisfied as long as we have a relatively small number of particles in the grid world. This means that the probabilities $p(s, a1, a2, s')$ are constant in time. To formalize the set of equations needed to derive the Bellman equation in analogy with what has been shown for the single-agent case, the optimal value v^k can be rewritten as

$$v^1\left(s, \pi^1, \pi^2\right) = \sum\nolimits_{t=0}^{\infty} \gamma^t E\left(s, \pi^1, \pi^2, r_t^1\right)$$

$$v^2\left(s, \pi^1, \pi^2\right) = \sum\nolimits_{t=0}^{\infty} \gamma^t E\left(s, \pi^1, \pi^2, r_t^2\right)$$

and from here the Q-learning algorithm can be reapplied. When implementing Q-learning in an experimental MAS, the amount of information that needs to be managed by the system must be carefully evaluated and the actual goals of the experiments must be defined. How big is the environment and how big are the agents? How many joint actions should be explored to build a model? What is the target of the system? These are questions that might look naïve, but in many cases, the MARL can be simplified reevaluating the idea behind the necessity of a stationary transition function.

Depending on the task that one assigns to a MAS, it might be better to establish some rules in their behavior which constrains the system. In this way, the problem of modeling becomes much more tractable given that many transitions do not need to be explored but the transitions of the whole system try to achieve some goal. This is done by coordinating the agents. In this way, specific planning is performed at a higher level to reduce the complexity of the emergent solutions provided by the interacting agents. This is called a multi-agent plan coordination problem. Such a procedure consists in partitioning the global problem with its corresponding fronts in terms of constraints and differentiation.

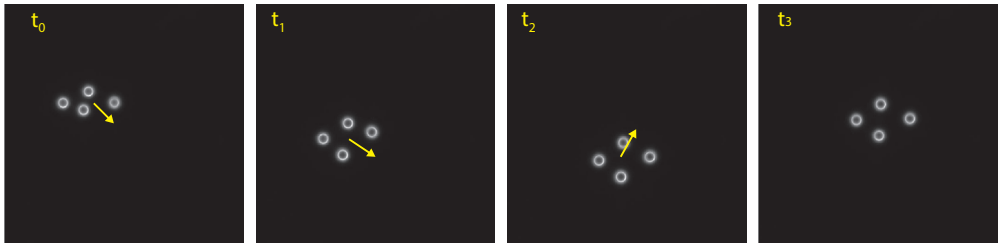


Fig. 19 Example of the navigation of a self-thermophoretic swimmer structure through a distributed algorithm (four swimmers forming a tetramer). In a first step, virtual interactions drive the formation of the swimmer structure. A second step produces the swarm navigation considering the same heating point for each swimmer and the propulsion direction taken with the center of mass of the structure as the reference point of the swarm.

An example of such an approach is shown in Fig. 19 with four microswimmers. On the one hand, each individual swimmer exhibits an active motion, and on the other hand, the swimmers form well-defined structures by defining virtual interactions [66]. Considering this, a distributed algorithm can be defined to now perform an active motion with the "swimmer molecule" or swarm as shown in Fig. 19. In a first step of the algorithm, the swimmers try to satisfy an "interaction condition" that creates the emergent structures, and in a second step, it takes care of the coordinated propulsion of each swimmer (component of the swarm) that results in the motion of the swarm with a given direction.

In such a scenario, the actions and the state of the swarm can be defined by the action taken by each swimmer (same action for all swarm components) and the position of its center of mass, respectively, and a classic reinforcement learning approach can be easily deployed. This represents a significant reduction in the complexity of the simple navigation problem for a swarm of swimmers, compared to defining a pure multi-agent concept with a single strategy function instead of an optimal joint strategy. Such a framework was also modified by replacing the target state of a classical gridworld by the position of a passive particle in the environment. This results in an exploratory behavior of the swarm, which tries to catch such a passive particle in the center of mass of the swarm ("haunting mode" in Fig. 20). To this end, the distributed algorithm is divided into an additional learning task. First, the swarm navigates through artificial interactions and active motion to its position that traps the passive particle at its center of mass. Second, the distance between the particles of the swarm is actively changed by the algorithm to trap the passive particle in the center.

5. Future directions for MARL

An important factor that determines the complexity of the problem to solve is the level of uncertainty not only in the transitions of the system after applying an action but also in the

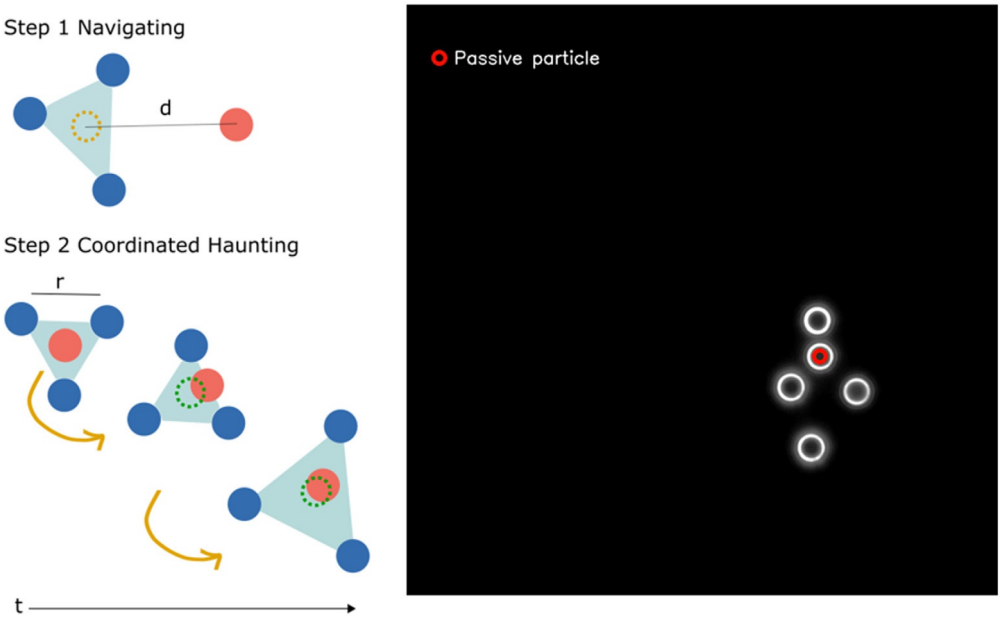


Fig. 20 Example of the coordinated haunting of a passive particle by a swarm of swimmers. In a first step, the swarm navigates toward the passive particle (*red circle* (dark gray in the print version)) that acts as a target. This mode is kept until a certain distance between target and the center of mass of the swarm is achieved. Under such a condition, a second mode is activated where the distance between the particles of the swarm is varied considering the dynamics of the passive particle in a way that the target remains always in the surroundings of the center of mass of the swarm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

observations. A framework to tackle this problem is known as the partially observable Markov Decision Process (POMDP). This is a generalization of the MPD approach but with the agent getting just partial information about the state of the world that includes also other agents. For that, the concept of belief is introduced as a probability distribution extended to all the states in the world. This enables the possibility to solve problems in a decentralized way. Meaning that the agents perform their own observations but the whole system develops a global solution. Such an approach is widely developed nowadays, and the systems based on it are known as decentralized partially observable Markov decision processes (DEC-POMDP). This probabilistic approach could be materialized in real systems using Bayesian reinforcement learning. Such a development might be particularly interesting for multi-agent systems especially when the collectivity shows heterogeneities. For the micro-robotic systems based on swimmers, the uncertainty regarding the robot position (states) is relatively low. However, if combined with another type of agents, such as biological matter, the state of these new agents, in a heterogeneous collectivity, would have certainly a significant uncertainty. Using MARL, the behavior

of the MAS can be studied and information regarding all the agents can be extracted by inverse learning strategies.

Another open field is the on-demand design of micro-robotic systems based on information extracted from robot interaction with the environment. Depending on the application, some requirements in terms of compatibility with the environment might determine the optimality of the robot. For that RL strategies can be combined with generative approaches in an objective-reinforced generative adversarial way (ORGAN). Here geometries and materials would play a key role in the design of micro-robotic systems to optimize their performance driving new research possibilities.

References

[1] D.J. Webre, P.M. Wolanin, J.B. Stock, Bacterial chemotaxis, Curr. Biol. 13 (2003) R47–R49.
[2] R. Stocker, J.R. Seymour, Ecology and physics of bacterial chemotaxis in the ocean, Microbiol. Mol. Biol. Rev. 76 (2012) 792–812.
[3] P. Romanczuk, G. Salbreux, Optimal chemotaxis in intermittent migration of animal cells, Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 91 (2015), 042720.
[4] J. Jumper, D. Hassabis, Protein structure predictions to atomic accuracy with AlphaFold, Nat. Methods 19 (2022) 11–12.
[5] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (2017) 221–248.
[6] F. Cichos, K. Gustavsson, B. Mehlig, G. Volpe, Machine learning for active matter, Nat. Mach. Intell. 2 (2020) 94–103.
[7] A.R. Dulaney, J.F. Brady, Machine learning for phase behavior in active matter systems, Soft Matter. 17 (2021) 6808.
[8] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach, Phys. Rev. Lett. 120 (2018) 24102.
[9] S. Maddu, Q. Vagne, I.F. Sbalzarini, Learning deterministic hydrodynamic equations from stochastic active particle dynamics, arXiv (2022).
[10] S. Maddu, B.L. Cheeseman, C.L. Müller, I.F. Sbalzarini, Learning physically consistent differential equation models from data using group sparsity, Phys. Rev. E 103 (2021), 042310.
[11] H. Jeckel, et al., Learning the space-time phase diagram of bacterial swarm expansion, Proc. Natl. Acad. Sci. U. S. A. 116 (2019) 1489–1494.
[12] S. Helgadottir, A. Argun, G. Volpe, Digital video microscopy enhanced by deep learning, Optica 6 (2019) 506.
[13] S. Helgadottir, et al., Extracting quantitative biological information from bright-field cell images using deep learning, Biophys. Rev. 2 (2021), 031401.
[14] B. Midtvedt, et al., Quantitative digital microscopy with deep learning, Appl. Phys. Rev. 8 (2021), 011310.
[15] M. Fränzl, F. Cichos, Active particle feedback control with a single-shot detection convolutional neural network, Sci. Rep. 10 (2020) 12571.
[16] G. Reddy, A. Celani, T.J. Sejnowski, M. Vergassola, Learning to soar in turbulent environments, Proc. Natl. Acad. Sci. U. S. A. 113 (2016) E4877–E4884.
[17] G. Reddy, J. Wong-Ng, A. Celani, T.J. Sejnowski, M. Vergassola, Glider soaring via reinforcement learning in the field, Nature 562 (2018) 236–239.
[18] S. Colabrese, K. Gustavsson, A. Celani, L. Biferale, Flow navigation by smart microswimmers via reinforcement learning, Phys. Rev. Lett. 118 (2017), 158004.
[19] M. Durve, F. Peruani, A. Celani, Learning to flock through reinforcement, Phys. Rev. E 102 (2020), 012601.

[20] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, Phys. Rev. Lett. 75 (1995) 1226–1229.
[21] A.C.H. Tsang, P.W. Tong, S. Nallan, O.S. Pak, Self-learning how to swim at low Reynolds number, Phys. Rev. Fluids 5 (2020), 074101.
[22] S. Verma, G. Novati, P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, Proc. Natl. Acad. Sci. U. S. A. 115 (2018) 5849–5854.
[23] S. Colabrese, K. Gustavsson, A. Celani, L. Biferale, Smart inertial particles, Phys. Rev. Fluids 3 (2018), 084301.
[24] L. Biferale, F. Bonaccorso, M. Buzzicotti, P.C.D. Leoni, K. Gustavsson, Zermelo's problem: optimal point-to-point navigation in 2D turbulent flows using reinforcement learning, Chaos 29 (2019), 103138.
[25] E. Schneider, H. Stark, Optimal steering of a smart active particle, EPL Europhys. Lett. 127 (2019) 64003.
[26] B. Hartl, M. Hübl, G. Kahl, A. Zöttl, Microswimmers learning chemotaxis with genetic algorithms, Proc. Natl. Acad. Sci. U. S. A. 118 (2021), e2019683118.
[27] Y. Yang, M.A. Bevan, B. Li, Efficient navigation of colloidal robots in an unknown environment via deep reinforcement learning, Adv. Intell. Syst. 2 (2020) 1900106.
[28] G. Gompper, et al., The 2020 motile active matter roadmap, J. Phys. Condens. Matter 32 (2020), 193001.
[29] C. Bechinger, et al., Active particles in complex and crowded environments, Rev. Mod. Phys. 88 (2016), 045006.
[30] E.M. Purcell, Life at low Reynolds number, Am. J. Phys. 45 (1977) 3–11.
[31] J.R. Howse, et al., Self-motile colloidal particles: from directed propulsion to random walk, Phys. Rev. Lett. 99 (2007), 048102.
[32] M. Fränzl, S. Muiños-Landin, V. Holubec, F. Cichos, Fully steerable symmetric thermoplasmonic microswimmers, ACS Nano 15 (2021) 3434–3440.
[33] R. Dreyfus, et al., Microscopic artificial swimmers, Nature 437 (2005) 862–865.
[34] A. Ghosh, P. Fischer, Controlled propulsion of artificial magnetic nanostructured propellers, Nano Lett. 9 (2009) 2243–2245.
[35] B. Dai, et al., Programmable artificial phototactic microswimmer, Nat. Nanotechnol. 11 (2016) 1087–1092.
[36] S. Palagi, et al., Structured light enables biomimetic swimming and versatile locomotion of photoresponsive soft microrobots, Nat. Mater. 15 (2016) 647–653.
[37] H.R. Jiang, N. Yoshinaga, M. Sano, Active motion of a Janus particle by self-thermophoresis in a defocused laser beam, Phys. Rev. Lett. 105 (2010), 268302.
[38] I. Buttinoni, G. Volpe, F. Kümmel, G. Volpe, C. Bechinger, Active Brownian motion tunable by light, J. Phys. Condens. Matter 24 (2012), 284129.
[39] A. Brown, W. Poon, Ionic effects in self-propelled Pt-coated Janus swimmers, Soft Matter 10 (2014) 4016–4027.
[40] J. Palacci, et al., Light-activated self-propelled colloids, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 372 (2014) 20130372.
[41] L. Alvarez, et al., Reconfigurable artificial microswimmers with internal feedback, Nat. Commun. 12 (2021) 4762.
[42] T.C. Lee, et al., Self-propelling nanomotors in the presence of strong Brownian forces, Nano Lett. 14 (2014) 2407–2412.
[43] S. Sanchez, A.N. Ananth, V.M. Fomin, M. Viehrig, O.G. Schmidt, Superfast motion of catalytic microjet engines at physiological temperature, J. Am. Chem. Soc. 133 (2011) 14860–14863.
[44] W.F. Paxton, et al., Catalytic nanomotors: autonomous movement of striped nanorods, J. Am. Chem. Soc. 126 (2004) 13424–13431.
[45] W.F. Paxton, et al., Catalytically induced electrokinetics for motors and micropumps, J. Am. Chem. Soc. 128 (2006) 14881–14888.
[46] G. Quincke, Ueber Rotationen im constanten electrischen Felde, Ann. Phys. Chem. 295 (1896) 417–486.

[47] E. Han, L. Zhu, J.W. Shaevitz, H.A. Stone, Low-Reynolds-number, biflagellated Quincke swimmers with multiple forms of motion, Proc. Natl. Acad. Sci. U. S. A. 118 (2021), e2022000118.
[48] A. Morin, N. Desreumaux, J.-B. Caussin, D. Bartolo, Distortion and destruction of colloidal flocks in disordered environments, Nat. Phys. 13 (2017) 63–67.
[49] A. Bricard, J.B. Caussin, N. Desreumaux, O. Dauchot, D. Bartolo, Emergence of macroscopic directed motion in populations of motile colloids, Nature 503 (2013) 95–98.
[50] F. Schmidt, et al., Microscopic engine powered by critical demixing, Phys. Rev. Lett. 120 (2018), 068004.
[51] K. Dietrich, N. Jaensson, I. Buttinoni, G. Volpe, L. Isa, Microscale Marangoni surfers, Phys. Rev. Lett. 125 (2020), 098001.
[52] A.P. Bregulla, A. Würger, K. Günther, M. Mertig, F. Cichos, Thermo-osmotic flow in thin films, Phys. Rev. Lett. 116 (2016), 188303.
[53] A.P. Bregulla, H. Yang, F. Cichos, Stochastic localization of microswimmers by photon nudging, ACS Nano 8 (2014) 6542–6550.
[54] X. Ma, et al., Enzyme-powered hollow mesoporous Janus nanomotors, Nano Lett. 15 (2015) 7043–7050.
[55] A. Joseph, C. Contini, D. Cecchin, S. Nyberg, L. Ruiz-Perez, J. Gaitzsch, G. Fullstone, X. Tian, J. Azizi, J. Preston, G. Volpe, G. Battaglia, Chemotactic synthetic vesicles: design and applications in blood-brain barrier crossing, Sci. Adv. 3 (8) (2017) e1700362, https://doi.org/10.1126/sciadv.1700362.
[56] S. Song, et al., Engineering transient dynamics of artificial cells by stochastic distribution of enzymes, Nat. Commun. 12 (2021) 6897.
[57] J.M. Walter, D. Greenfield, C. Bustamante, J. Liphardt, Light-powering *Escherichia coli* with proteorhodopsin, Proc. Natl. Acad. Sci. 104 (2007) 2408–2412.
[58] G. Frangipane, et al., Dynamic density shaping of photokinetic E. coli, eLife 7 (2018), e36608.
[59] J. Arlt, V.A. Martinez, A. Dawson, T. Pilizota, W.C.K. Poon, Painting with light-powered bacteria, Nat. Commun. 9 (2018) 768.
[60] M.N. Popescu, W.E. Uspal, C. Bechinger, P. Fischer, Chemotaxis of active Janus nanoparticles, Nano Lett. 18 (2018) 5345–5349.
[61] L. Baraban, S.M. Harazim, S. Sanchez, O.G. Schmidt, Chemotactic behavior of catalytic motors in microfluidic channels, Angew. Chem. Int. Ed. Eng. 52 (2013) 5552–5556.
[62] C. Zhou, et al., Torque-driven orientation motion of chemotactic colloidal motors, Angew. Chem. Int. Ed. 61 (2022), e202116013.
[63] D.P. Singh, et al., Interface-mediated spontaneous symmetry breaking and mutual communication between drops containing chemically active particles, Nat. Commun. 11 (2020) 2210.
[64] B. Qian, D. Montiel, A. Bregulla, F. Cichos, H. Yang, Harnessing thermal fluctuations for purposeful activities: the manipulation of single micro-swimmers by adaptive photon nudging, Chem. Sci. 4 (2013) 1420.
[65] N.A. Söker, S. Auschra, V. Holubec, K. Kroy, F. Cichos, Active-particle polarization without alignment forces, arXiv (2020).
[66] U. Khadka, V. Holubec, H. Yang, F. Cichos, Active particles bound by information flows, Nat. Commun. 9 (2018) 3864.
[67] T. Bäuerle, R.C. Löffler, C. Bechinger, Formation of stable and responsive collective states in suspensions of active colloids, Nat. Commun. 11 (2020) 2547.
[68] T. Bäuerle, A. Fischer, T. Speck, C. Bechinger, Self-organization of active particles by quorum sensing rules, Nat. Commun. 9 (2018) 3232.
[69] M.A. Fernandez-Rodriguez, et al., Feedback-controlled active brownian colloids with space-dependent rotational dynamics, Nat. Commun. 11 (2020) 4223.
[70] T. Mano, J.-B. Delfau, J. Iwasawa, M. Sano, Optimal run-and-tumble-based transportation of a Janus particle with active steering, Proc. Natl. Acad. Sci. 114 (2017) E2580–E2589.
[71] G. Volpe, S. Gigan, Brownian motion in a speckle light field: tunable anomalous diffusion and selective optical manipulation, Sci. Rep. 4 (2014) 3936.

[72] M. Selmke, U. Khadka, A.P. Bregulla, F. Cichos, H. Yang, Theory for controlling individual self-propelled micro-swimmers by photon nudging I: directed transport, Phys. Chem. Chem. Phys. 20 (2018) 10502–10520.
[73] M. Selmke, U. Khadka, A.P. Bregulla, F. Cichos, H. Yang, Theory for controlling individual self-propelled micro-swimmers by photon nudging II: confinement, Phys. Chem. Chem. Phys. 20 (2018) 10521–10532.
[74] R., W. G., A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik 35 (1972) 237–246.
[75] M. Hossein Eybposh, N.W. Caira, M. Atisa, P. Chakravarthula, N.C. Pégard, DeepCGH: 3D computer-generated holography using deep learning, Opt. Express 28 (2020) 26636–26650.
[76] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[77] R. Joseph, F. Ali, YOLO9000: Better, Faster, Stronger, arXiv, 2016.
[78] M. Mijalkov, A. McDaniel, J. Wehr, G. Volpe, Engineering sensorial delay to control phototaxis and emergent collective behaviors, Phys. Rev. X 6 (2016) 1–16.
[79] R. Sutton, Introduction to Reinforcement Learning, second ed., 2018.
[80] V. Mnih, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.
[81] J. Degrave, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (2022) 414–419.
[82] K. Gustavsson, L. Biferale, A. Celani, S. Colabrese, Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning, Eur. Phys. J. E 40 (2017) 110.
[83] G. Beintema, A. Corbetta, L. Biferale, F. Toschi, Controlling Rayleigh–Bénard convection via reinforcement learning, J. Turbul. 21 (2020) 585–605.
[84] M. Buzzicotti, L. Biferale, F. Bonaccorso, P.C.D. Leoni, K. Gustavsson, Optimal control of point-to-point navigation in turbulent time-dependent flows using reinforcement learning, arXiv (2021).
[85] F. Borra, L. Biferale, M. Cencini, A. Celani, Reinforcement learning for pursuit and evasion of microswimmers at low Reynolds number, arXiv (2021).
[86] S. Muiños-Landin, A. Fischer, V. Holubec, F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. 6 (2021), eabd9285.
[87] V. Mnih, et al., Playing atari with deep reinforcement learning, arXiv (2013).
[88] Z. Wang, et al., Dueling network architectures for deep reinforcement learning, arXiv (2015).
[89] L. Panait, S. Luke, Cooperative multi-agent learning: the state of the art, Auton. Agent. Multi-Agent Syst. 11 (2005) 387–434.

CHAPTER 6

Applications of convolutional neural networks for spectral analysis

Yu Li[a,b], Zhengchang Liu[a], and Zheyu Fang[a]
[a]School of Physics, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China
[b]Silicon Photonics R&D Center, CUMEC, Chongqing, China

1. Introduction

Subwavelength optical metamaterials exhibit unique optical properties compared with a conventional macroscopic optical medium, including a plasmonic structure that can support free electron oscillations on the metal surface [1], and dielectrics with a high refractive index that can realize complex Mie scattering and Bragg diffraction [2]. Among them, metallic nanostructures can confine the electromagnetic field in the subwavelength scale, which can realize high manipulation efficiency for both near-field distributions and far-field spectra [3]. Low-loss dielectric photonic crystal materials show higher efficiency in optical far-field manipulations and are widely used in metalens [4], optical holograms [5], and metasurfaces [6]. With the development of nanofabrication technology and increasing demand for integrated optoelectronic devices, efficient and rapid design flows to assist researchers in studying and developing diversified light-field manipulation are especially in need.

As a rapidly developing discipline in recent years, neural networks (NNs) have been promoted and applied in various research and application fields, including the detection of the geological data models [7], prediction of earthquake risks [8], derivation of element periodic law through compound structure formula [9], and reading of nuclear magnetic resonance imaging for malignant tumor assessment [10]. Those algorithms can recognize the hidden rules and patterns in huge amounts of data, analyze and predict mapping from feature variables to the corresponding objective functions, and optimize functions.

According to the calculation flow of the NNs, there are various architectures that are designed for specific machine learning tasks, including fully connected neural networks (FCNNs), recurrent neural networks (RNNs), graph neural networks (GNNs), and convolutional neural networks (CNNs) [11]. Among them, CNNs only involve adjacent elements in each layer calculation, so the trainable parameters of this method are far less than those of FCNNs with the same number of elements [12]; this can avoid overfitting

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00006-8
Copyright © 2023 Elsevier Inc.
All rights reserved.

and increase training efficiency. In addition, CNN input is mostly in a matrix format and the geometric features of the input patterns are recognized by kernels, which makes it more suitable for edge detection and clustering information between pixels.

In research on optical metamaterials, the matrix-type input makes CNN parameterize the structural design from a larger parameter space, which increases the scalability of the model. In addition, for the spectra analysis target, CNN can output both the vector and matrix, which makes it possible to integrate more complex optical responses such as angular distribution, time response, and near-field distribution. From the data-driven perspective, CNNs can help researchers find the potential relationship between a large number of samples as well as construct the mapping features or clustering rules between complex material designs and spectral responses. Moreover, CNNs can also be utilized as generative models to design metamaterials in vast parameter space [13], realize the spectra control of various dimensions and multiple targets, and then directly contribute to the development of high-performance optical materials and devices.

2. Fundamentals of CNNs for photonics

There are five steps for CNN-assisted metamaterial analysis: parameterizing nanostructures into network inputs, building an appropriate network architecture, providing effective training datasets, training the network, and quantifying the prediction accuracy of the model.

2.1 Parameterization strategies

For FCNNs and other conventional prediction models, the structural design is parameterized into a vector (1,n) according to the specific pattern type, and each number in the vector can represent the structural or material properties such as length, width, height, azimuth, refractive index, etc. However, due to the need to define the design type in advance, the free degree of structural design is limited. Compared with FCNNs, the design diversity can be greatly expanded by CNNs because the CNN input is a matrix with the shape of (C,n,n), where C is the channel and (n,n) is the size of the input pixels.

Before feeding designs into models, the structure should be parameterized into a specific matrix, where the row and column position of each element in the matrix corresponds to the spatial position in the structural design. For example, for a $1000\,\text{nm} \times 1000\,\text{nm}$ designing space, represented by a (100,100) sized matrix, the corresponding spatial range of each pixel is $10\,\text{nm} \times 10\,\text{nm}$. The size of the structure matrix determines the accuracy of the parameterized model. Higher accuracy can achieve a more precise design of nanostructures, but it also increases the complexity of the CNN model. In addition, it is necessary to match the parametric accuracy, numerical simulation accuracy, and fabrication accuracy. At present, a large number of training datasets are provided by numerical simulations. When the simulation accuracy is lower

than the parametric accuracy, the discrete element of simulation cannot distinguish the tiny difference of input structure. Under this condition, different structural parameters may be mapped to the same spectrum, which can lead to a large amount of unnecessary resource consumption for network training. On the contrary, if the simulation accuracy is much higher than the parameterization accuracy, generating training data can be time-consuming, and the risk of overfitting may be increased.

In the parameterization process, the physical meaning of each pixel can be specifically designed according to the characteristics of the considered system. A well-applied method is to describe the single layer of structures with a binary matrix (0/1), where "1" represents locations existing in the structure and "0" represents locations with air, shown as Fig. 1A. This parametric method shows advantages for its simplicity. No additional normalization processing should be applied for network training, and a one-step lithography or electron beam exposure is sufficient for structure fabrication.

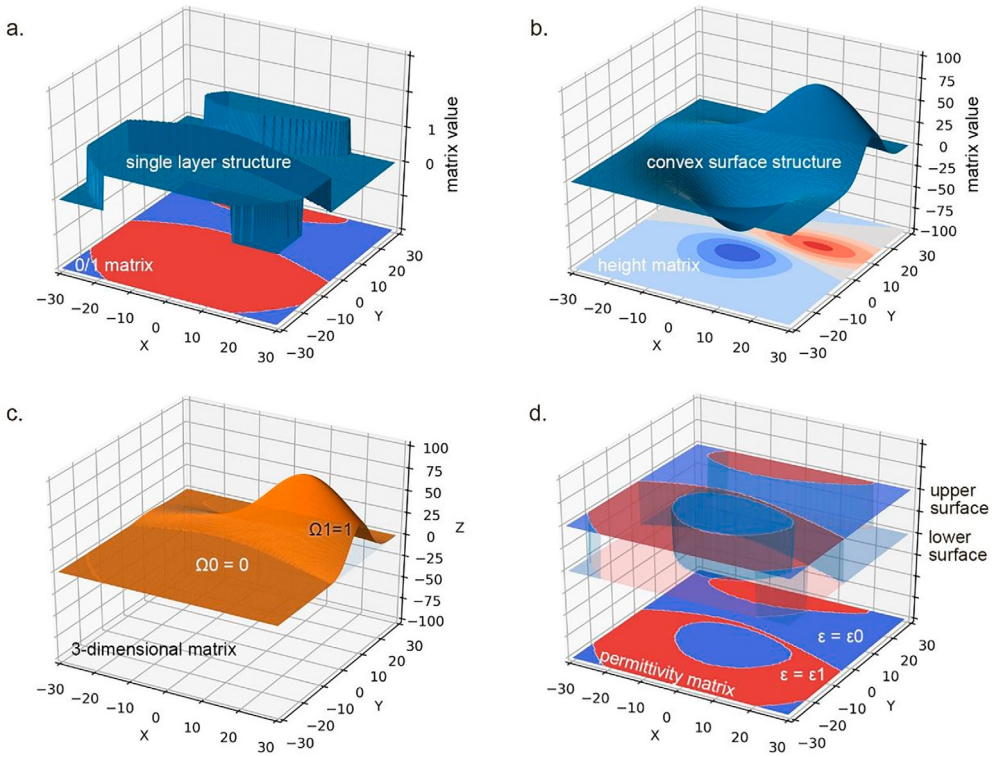


Fig. 1 Several parameterization methods for metamaterial design. (A) Binary matrix for a single-layer structure. (B) Matrix with continuous values for a convex surface; the value of each pixel represents the height of the surface at the corresponding location. (C) Three-dimensional matrix for an arbitrary structure; the value of voxel Ω is binary. (D) Parameterization with the matrix of which the value is permittivity.

The disadvantages are also obvious, as this method has less information and gives up the structural features that can be carried by pixel values between 0 and 1. Also, because the intermediate value is not defined, NN needs a long training epoch to make the output value be distributed at both ends of 0/1 as much as possible rather than an undefined gray region.

If continuous pixel intensity is introduced and the pixel value is defined as the height information, the design of any convex surface can be described (Fig. 1B). When applying this method, the continuity between adjacent pixels should be predefined to avoid the rough surface that cannot be experimentally fabricated.

At the same time, this method can only define the height information once at a coordinate position, so the defined structure is limited as a convex surface. To define nonconvex structures, the channel of the input matrix can be used to parameterize the structure by voxel, the channel index corresponds to the height in the z-direction of the structure, and the definition of each lattice point in the three-dimensional (3D) space can be realized. Shown in Fig. 1C, in each spatial (XYZ) location of the voxel, the value of $\Omega = 1$ or 0 corresponds to the volume with or without the structure.

When the pixel value is defined as the material property itself rather than the structural design, such as the refractive index or permittivity of the material in different spatial positions, a higher degree of design freedom can be generated in the single-layer material design (Fig. 1D). But at the same time, the defined refractive index must exist in the real world, and the pixel value should be stable in a certain region to meet the requirements of fabrication.

2.2 Mathematical operations

A CNN can predict the mapping between the metamaterial design to the spectral response or electromagnetic field distribution in nanophotonics. Its core mathematical operations include convolution, pooling, and nonlinear activation [14]. Generally speaking, the output of spectral response is a vector while the output of electromagnetic field distribution is a matrix. Therefore, based on the core mathematical operation, specific network structures should be built to meet the needs of different optical response analyses.

In the convolution process of CNN, the parameters that determine the range of pixels involved in each layer of computation are called the kernel. The simplest kernel is a matrix, and the value contained in the matrix is the trainable weight of the CNN. The process of fitting is realized by adjusting the weight information. In each calculation, the kernel multiplies points corresponding to coordinates with the part with the same number of rows and columns in the input matrix and outputs several matrices with the same number of rows and columns. During the calculation, the number of weights

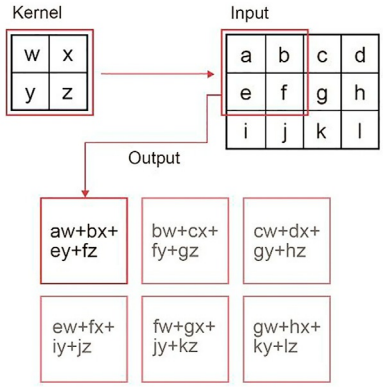


Fig. 2 Demonstration of the kernel method. A kernel with the size of (2,2) is applied in the input with the size of (4,3).

formed between pixels is determined by the kernel size, so the total parameters are effectively reduced, which can reduce the complexity of the model and prevent overfitting of the model. The mathematical expression of the convolutional kernel is as follows:

$$V(x, y) = (I \cdot K)(x, y) = \sum_m \sum_n I(x + m, y + n)K(m, n) \tag{1}$$

While the corresponding manipulation methods are as demonstrated below.

Shown in Fig. 2, the calculation range of the convolutional kernel is (2,2). In each time, four pixel-wise products are performed between the kernel matrix and the influenced region of the input matrix, and the sum of the products is output. For example, applying $\begin{pmatrix} w & x \\ y & z \end{pmatrix}$ on the part of the input $\begin{pmatrix} a & b \\ w & f \end{pmatrix}$, the output $= aw + bx + ey + fz$. For the other parts of the input matrix, the convolutional kernel shares the same weight parameters, and the whole input matrix is calculated by sliding the kernel position. Another important characteristic of a CNN is the attribute of weight sharing. For the same kernel, similar features at any location of the whole input matrix can be recognized, and the variables in the input matrix will no longer be regarded as independent numbers. Instead, trainable parameters are combined by the kernel size and the definition of sliding mode; this method can effectively increase the recognition ability of the algorithm for multiscale features in the matrix.

Max-pooling is a method to simplify the operation of input data by using the maximum (most sensitive region) feature points in the input region (an n × m matrix). Pooling itself does not store any weight information, and one of the core objectives is to provide spatial variation. By this method, a CNN can identify an object even if its

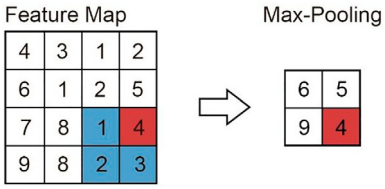


Fig. 3 Max-pooling manipulation. Demonstration of max-pooling manipulation with the size of (2,2) applied on a (4,4) feature map.

appearance changes in some way. Moreover, pooling can extract data input with a large amount of information into less data output, which realizes a dimension reduction. The demonstration of the max-pooling calculation is as follows:

Applying max-pooling with the size of (2,2) on a (4,4) feature map, an output with the size of (2,2) can be acquired while the number of channels is unchanged (Fig. 3). When running spectra analysis with a CNN, the structural matrix is mapped into an increasing number of channels to load geometric features. Meanwhile, max-pooling manipulation is applied to decrease the complexity of the output while maintaining the spectra information. Because of the existence of max-pooling, the perceptual field of the convolutional kernel can be varied according to the depth of the layer, which enables feature recognition for different scales of nanostructures.

The nonlinear activation function is the core to realizing complex regression and classification for an NN. When feeding data into a network unit, a linear transformation is first achieved by the production between the weight matrix and the input vector. Then, a nonlinear activation is applied before feeding into the next layer. The most common activation functions include rectified linear unit (ReLU), sigmoid, and tanh. ReLU is comprised of two segments of linear activation, which gives it high computational efficiency and fast training speed and makes it suitable for the hidden layers in most of the spectra analysis tasks. Furthermore, the kinds of activation functions used often need to be carefully selected according to specific computing tasks, especially in the output layer of the neural network. Because the data calculated by nonlinear functions directly obtains spectral data, it is necessary to consider that the specific output domain of the activation function needs to match the range of spectral response.

According to the above response curve in Fig. 4, the value range of ReLU is $(0,\infty)$, the value range of sigmoid is (0,1), and the value range of tanh is $(-1, +1)$. There is no saturation in the ReLU function range, which may make the output exceed the upper limit of the spectral value, so it is not suitable for the network to predict the transmittance or reflectivity. In contrast, sigmoid activation and tanh can be saturated to (0,1), and (−1.1), respectively, which makes them more suitable to represent normalized data and the direction of energy flow through a certain observation surface.

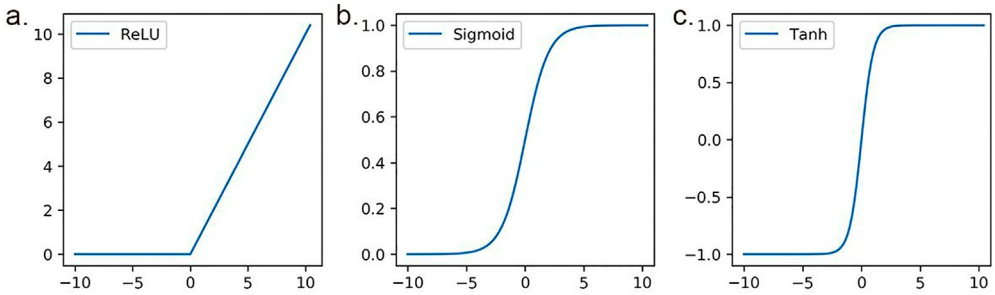


Fig. 4 Activation functions. Responses of ReLU, sigmoid, and tanh activation functions.

2.3 Sampling and training

The training data for CNN is comprised of a structure matrix and corresponding spectra responses, which can be either vector or matrix. Corresponding spectra responses can be acquired by experimental measurements or numerical simulations such as the finite difference-time domain (FDTD) or the finite element analysis (FEA) method.

The designing structures are generated by random algorithms and the samples should meet the requirements of diversity and effectiveness. Diversity means that different kinds of designs can be provided to simultaneously predict a rectangle, circle, polygon, triangle, or other complex structure. The above structures need to be described by different parameter vectors in FCNNs, but they can all be pixelized by structure matrix, so different structure types for CNNs can belong to the same training dataset simultaneously. At the same time, the inductive learning of boundary and pattern information can support multistep sampling. Combined with different types of structural characteristics, the model can automatically generate training data that do not exist in the initial dataset to realize automatic sampling. The training data can also be obtained by image processing or a combination of multiple types of parametric structures, such as erosion, dilation, tailoring, intersection, union, etc.

By importing the patterns in Fig. 5A into the structural arrays with the period of 200 nm × 200 nm, where the black area is defined as a single layer of 60 nm gold and the source is defined as the normal incidence of vertically polarized light, the corresponding spectra responses can be calculated by FDTD; the obtained transmission spectra are shown in Fig. 5B.

For the same original structure, the transmittance of after erosion increases because of smaller structural size, on the contrary, dilation leads to the decreases of transmission, but the transmission valley near the 600 nm wavelength does not shift significantly due to the small change of the length in the *Y* direction. While the change of tailoring and intersection to the structural design results in the decrease of both duty ratio and geometric length in the *Y* direction, so the transmission valley blue-shifts and the transmission

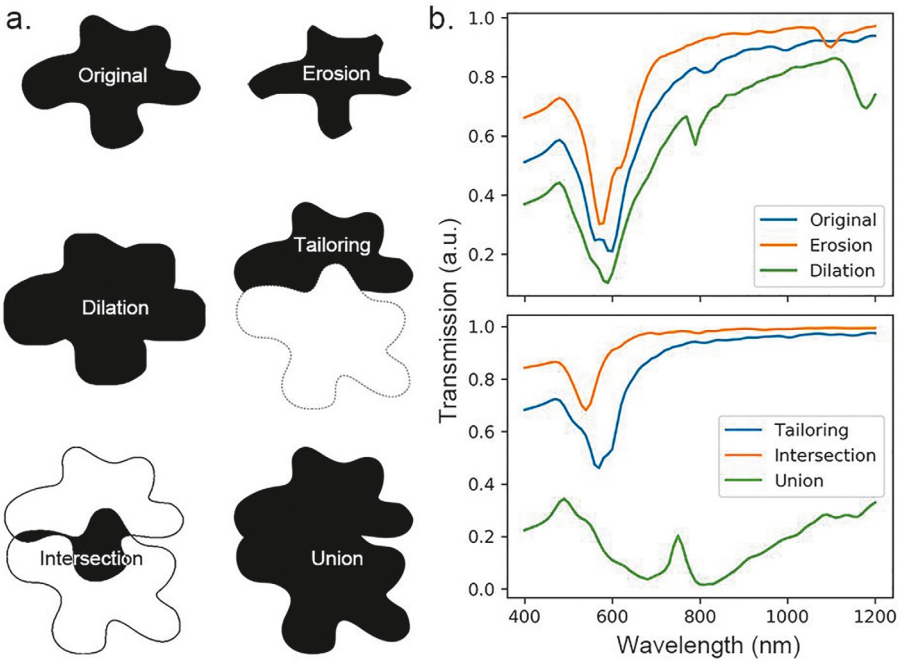


Fig. 5 Demonstration of image processing methods for sample generation. (A) Generating new training samples with erosion, dilation, tailoring, intersection, and union manipulations. (B) Corresponding transmission spectra.

increases greatly. Because the union pattern covers almost the whole structure period, it presents two transmission valleys at 680 and 800 nm, and the average transmittance is less than 0.4. When fine-tuning the spectrum within a limited range of parameters is needed, erosion and diffusion can be applied to find the local optimal region. However, if the training set should be expanded to increase variation, the other methods can be utilized to jump out of the local optimum.

The effectiveness of training data is determined by the data characteristics of the training set and test set. The goal of NN training is to use a gradient descent algorithm and backpropagation to adjust the weight and minimize the training loss. Therefore, the trained NN calculation results of training data are usually better than those of validation data and testing data. The closer the data distribution of testing data is to training data, the higher the prediction accuracy. In NN sampling and training, it is necessary to consider the consistency of distribution of the training set and the test set to ensure the effectiveness of samples. If the samples and spectral characteristics are so rare that the corresponding proportion in the training data is less than 5%, it can be difficult for the NN to learn the optical properties of the target in the nonequilibrium dataset. In a worst-case scenario, the minority data with less than 5% of the samples could be recognized as noise, resulting in poor performance in the prediction of the optical response of the target.

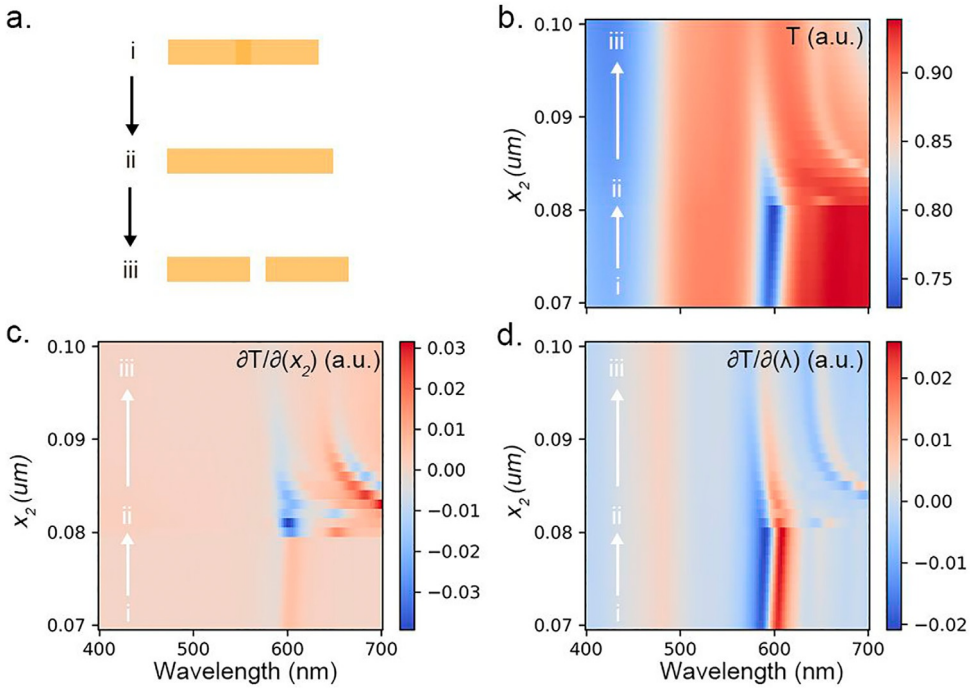


Fig. 6 Example of spectral property sensitivity changing with parameters. (A) Demonstration of two metal rods with changing *x*-position of the second bar. (B) Mapping of the transmission spectra (*T*) as the function of *x*. (C, D) The mapping of $\partial T/\partial x$ and $\partial T/\partial \lambda$ as the function of *x* and λ.

In some nanostructured systems, the tendency of spectral response changes sharply, which interferes with the fitting procedure. An example is the composite structure composed of two Au bars shown in Fig. 6A, of which the corresponding *x*-position is defined as x_1 and x_2. Changing x_2 will cause the continuous change of the transmission spectrum (Fig. 6B and C). When x_2 is less than 0.08 μm, the composite structure is a single structure with the long axis increasing gradually, and a stable dipole oscillation mode is formed on the metal surface. With the increase of x_2, the resonance wavelength increases gradually and has good continuity. However, when $x > 0.08$ μm, the long rod will be divided into two rods with a length of 0.2 μm. At this time, the original single dipole mode disappears and a strong gap mode is formed between the two rods, thus the spectra can be strongly changed. From the physical point of view, these discontinuous spectral changes can be explained, but for the input of NN, the instability of spectral output in different independent variable parameter spaces increases the difficulty of model training.

Additionally, the sharpness of spectral peaks ($\partial T/\partial(\lambda)$) at different wavelengths can also be sensitive, as shown in Fig. 6D. $\partial T/\partial(\lambda)$ at the wavelength of 600 nm is much higher than that at other wavelengths, so the activation of different neuron needs should

be more flexible, which may lead to unnecessary overfitting. All the above situations need to be fully considered before NN training to build customized CNN models according to the spectral characteristics of specific models and to achieve a balance between overfitting and underfitting.

To further improve the prediction accuracy of the test set, the direct solution is to sample near the designs with poor prediction performance and provide more structures for NN training, which also increases the calculation cost of training data. Thus, an optional method called data enhancement can be used. For the array structure, changing the structure position in the period does not influence the corresponding optical responses, so several structural matrices can be mapped to the same spectrum as the training set without additional simulations. Under the incident of unpolarized light, rotating the structure itself does not produce the change of spectral response, either. Moreover, when the incident light is circularly polarized, the optical response of left-handed circularly polarized light to the nanostructure is consistent with that of right-handed circularly polarized light to its mirror isomer. Because these data enhancement methods make use of the spatial symmetry of specific models, even the parameters of multiple input matrices are different, but the corresponding spectra can be inferred without additional numerical simulation.

3. Predictive models for spectra calculation

The input data of a CNN is matrices, which makes it widely used to process image information. In recent years, CNNs have been used to predict the optical response, which may be the Q-factor of photonic crystal [15]; the topological edge state of photonic crystal [16]; or the spectral shape design of nanostructures, such as reflection [17], transmission, or absorption spectrum [18]. This kind of CNN module that can predict the spectrum from the matrix plays an important role in the realization of generative adversarial networks (GAN) and other algorithmic networks. In the field of nanophotonics, CNN models have been widely used to map the images of nanostructures with their optical responses.

3.1 Fundamental spectra analysis

The steps in using CNNs to predict spectra generally include building an algorithm, calculating the dataset, network training, testing the prediction results, and experimental verification. As a data-driven algorithm, it needs enough data to achieve the best prediction level through training. The construction of the dataset is often completed by numerical simulation. However, the construction of numerical simulation often requires a lot of computing time and resources, so once the CNN used to predict the spectrum is well trained, it can greatly save the time and computing resources needed to obtain the spectrum of the new structure.

An algorithm network is proposed to predict the absorption curve of the plasmon structure, and it is composed of a CNN and a recurrent neural network (RNN) in series [19]. As shown in Fig. 7A, the CNN extracts the hidden geometric features from the nanostructure when its image passes through the convolution layer, pooling layer, and fully connected layer. Input to the RNN, the optical absorption curve of the plasmon nanostructure is obtained from geometric information through a gated recurrent unit of the RNN. The CNN module consists of 11 convolution layers, a maximum pooling layer, a leaky ReLU as the activation function, and also a residual network from which better results are available after a short cut where the gradient can directly enter deeper layers. Because the RNN needs a special data input format, a time-distributed layer is used to connect the CNN and RNN. This layer, which is a coding technology, can transform the output data of the CNN module into the input data of the RNN. After that, the data are input to a gated recurrent unit layer and then go through a flattening layer that converts the RNN output into the fully connected layer input. After passing through the flattening layer, the data are input into a fully connected layer with 3000 nodes, followed by a fully connected layer with 1000 nodes, which corresponds to a thousand frequency points of the spectrum.

A CNN algorithm based on a generative adversarial model is proposed to predict the metasurface transmission spectrum [20]. As shown in Fig. 7B, the algorithm framework consists of three modules: the generator, critical, and simulator. The function of the simulator module is to predict the transmission spectrum from the structure picture. In the whole training process of GAN, the simulator is a pretrained overfitted CNN module. The simulator consists of four convolution layers followed by a fully connected layer. The input data are the binary 64×64 single-channel matrix of the structure image. After four convolution layers, the data structure becomes 64×64 of 64 channels, 32×32 of 64 channels, 16×16 of 128 channels, and 8×8 of 256 channels successively. After a transition layer, the data structure becomes a vector with 4608 components, and then after two fully connected layers, the vector component decreases to 256 and 128 in turn. Finally, the vector with 128 components is the discretized transmission spectrum.

The critical module calculates the distance between the distribution of the structure image designed by the generator according to the input spectrum as well as the noise and the distribution of the prepared geometry-image dataset. In addition to the output of the simulator module, the distance calculated by the critical module is also introduced into the loss function to change the weight of the generator module. By reducing this distance, the critical module is used to guide the picture generation of the generator module. The images generated by the generator will be similar to the predefined geometry dataset.

A computational platform based on Bayesian optimization for obtaining the circular dichroism (CD) of the reflection spectra from gold nanostructures using CNN is proposed [21]. The metal structure is transformed into a 40×40 matrix, and the array of matrix elements represents the array of the pixel gold nanostructure on the substrate.

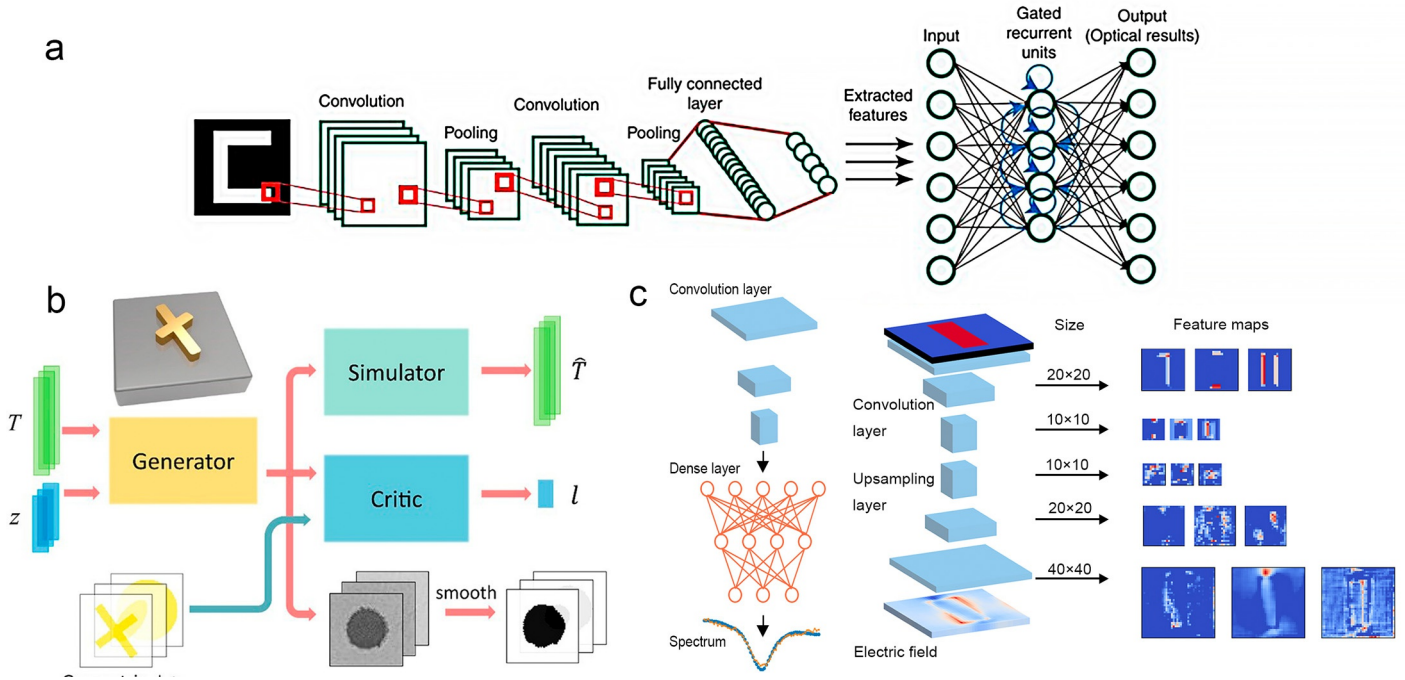


Fig. 7 CNN network architecture for predicting nanostructure spectra. (A) The neural network architecture for predicting the absorption spectrum of nanostructures based on CNN and RNN. (B) Deep CNN architecture for predicting the transmission spectrum of nanostructures based on GAN. (C) The neural network architecture for mapping the structure matrix with the optical response. The optical response of the left part is the far-field reflection spectrum, and the optical response of the right part is the near-field electric field distribution. *(From 2018 American Chemical Society, 2019 American Physical Society.)*

The two neural networks transform the matrix into the far-field reflection spectrum (left side of Fig. 7C) and the near-field electric field distribution (right side of Fig. 7C).

The CNN architecture for predicting the far-field reflectance spectra is shown on the left side of Fig. 7C. The hidden features of matrices are extracted through six convolution layers and three max-pooling layers. The structure of the single data will be changed from 40×40 of the single channel to 5×5 of the 512 channel. Every two convolution layers are connected with a max-pooling layer and they constitute a group. The numbers of convolution layer output channels of the three groups are 128, 256, and 512, respectively. The convolution layer output matrix sizes are 40, 20, and 10. The maximum pooling layer channel numbers are the same as the convolution layer output channel numbers. For these three groups, the output matrix sizes are 20, 10, and 5 successively and the parameter volumes are 526,720, 2,622,208, and 10,487,296. After the flattening layer that connects the CNN with a fully connected layer, the data become a 12,800-component vector. After three fully connected layers, the data will become a 60-component vector to establish the relationship between the features and the spectrum. Finally, the image is transformed into the far-field reflection spectrum. The right side of Fig. 7C shows a CNN architecture for predicting the near-field electric field distribution. The deeper the CNN layer, the smaller the size of the matrix and the greater number of data channels. In the downsampling process of CNN prediction, different convolution layers extract different structural information. The first convolution layer extracts the structural information of lines and angles, and the later convolution layer extracts the structural information for designing the electromagnetic mode distribution. In the upsampling of predicting the near-field electric field distribution, the deeper convolution layer extracts more information, including the distribution of electromagnetic modes and the details of reconstructing the near-field distribution.

In addition to building the neural network algorithm, we also need to obtain the dataset to train, validate, and test the algorithm. In nanophotonics, datasets usually consist of device geometry and its corresponding optical response. For a CNN, the geometric structure of the device is generally a single-channel or multichannel matrix, which is usually a picture carried by pixels. The optical response of a large number of structures is often obtained by solving Maxwell's equations, given the light source, geometry, material properties, and boundary conditions. For nanostructure arrays, the unit structure is often simulated numerically.

As shown in Fig. 8A, in the work of combining a Bayesian optimized CNN to obtain a high CD structure, the nanostructure is gold with a period of 400 nm and a height of 40 nm on an SiO_2 substrate. Under the silica layer is the silicon layer, and the gold structure is composed of cubic pixels with a side length of 10 nm. The 40×40-pixel matrix is a discrete unit matrix. If the value of the matrix element is 1, there is the structure in the corresponding position. If the value of the matrix element is 0, there is no structure. FDTD is used to construct the dataset and 104 samples constitute the database.

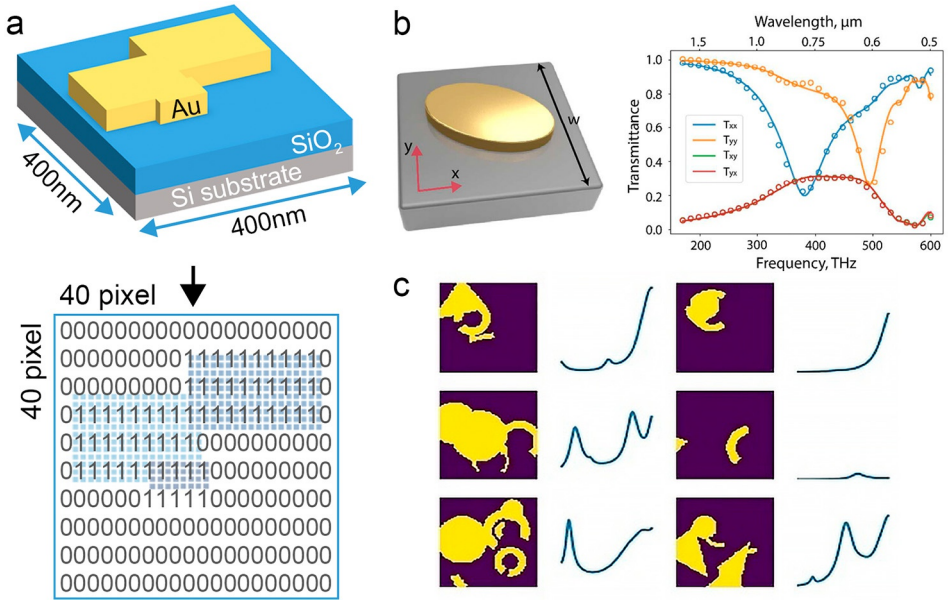


Fig. 8 Demonstration of the parameterization method and the corresponding dataset. (A) The Au nanostructures on the SiO_2 substrate are transformed into a 40×40 binary matrix as the structural data. (B) The schematic of metal nanostructures on the substrate is on the left, *w* is the period of the structure array, and the definition of the *XY* direction is shown in the figure. The transmission spectrum of the structure is on the right. The *solid line* is the data obtained by FEM electromagnetic simulation, the *dot* is the structure obtained by the simulator module, and the two subscripts are the linear polarization direction of the incident light and the probe light, correspondingly. (C) Six structures in the randomly generated dataset and the simulated absorption curves. *(From 2018 American Chemical Society, 2019 American Physical Society.)*

The optical response includes two parts, near-field electric field distribution and far-field reflection spectrum. In predicting the CD value, fix a working frequency to optimize the CD value. The fixed working frequencies are 650 nm, 700 nm, 750 nm, and 800 nm. Bayesian optimization is used to train the neural network, which selects samples from the dataset and inputs them into the neural network for training according to the output results of the current neural network. After many epochs, it completes the training of the neural network. Bayesian optimization in this work is an optimization method to complete neural network training by guiding training samples. This method makes the training easier, reduces the possibility of overfitting, and makes it easier to extract features from the structure.

As shown in Fig. 8B, in the work with a generative neural network to predict the transmission spectrum, many gold structures with a period of 340 nm and a height of 50 nm on the glass substrate form a metasurface. The operation frequency region is

170–600 THz, and 32 frequency points are sampled. According to the polarization direction of the incident light and the probe light, four transmission curves can be obtained. The first subscript of the transmission curve represents the polarization direction of the incident light, and the second represents the probe light. The solid line in the figure is the result of the FEM electromagnetic solver, and the dot is the result of a simulation neural network.

In the work of combining CNN and RNN to predict the absorption spectrum (Fig. 8C), the structure is silver with a height of 50 nm on the glass substrate. The period of the structural array is 500 nm, and the geometric structure of silver is constant along the direction perpendicular to the substrate. Therefore, a (100,100) pixel array can be used as a picture to characterize the unit structure. The operation wavelength region ranges from 800 to 1700 nm, which is sampled by 1000 frequency points. Compared with the visible light band, gold in the infrared band has a higher refractive index and a lower loss. FDTD is used to calculate the absorption curve of the structure. After retaining certain geometric features, such as triangles, split-ring resonators, circles, or triangles, pixel arrays are randomly generated to form the structure, and 105 simulations constitute the dataset.

When training the neural network with a dataset, several things should be noticed. The network requires that the data distribution of the training set, validation set, and test set should be the same or similar to obtain a neural network with high prediction accuracy. The neural network is a data-driven algorithm, and enough data is the premise of the CNN algorithm. In the work of combining CNN and RNN to predict the absorption spectrum, the neural network prediction results of the sample in Fig. 9A are in good agreement with the numerical simulation results. One of the important reasons is that the number of samples in the dataset is large enough to reach 105. If the data are not enough, data enhancement, migration learning, or topology optimization of generative models should be used to make up for the shortage of the dataset. Fig. 9B depicts a sample in the dataset used for the prediction of the transmission spectrum by the generative neural network. If the number of samples in the dataset is not enough, the neural network can generate more samples to meet the needs. When the nanostructure array has spatial translational symmetry or the optical response difference between adjacent unit structures is small enough, the boundary conditions around the nanostructure are usually chosen to be periodic. The dataset derived from periodic boundary conditions is only suitable for periodic structure arrays or spatial gradual varying arrays, and the application range of training data is limited by the nature of the nano design platform. In the dataset, more samples should be collected in the area where the optical response changes sharply with the structural parameters, and fewer samples should be collected in the area where the optical response changes slowly. Because the intensive sampling in the sharply changing area can improve the learning accuracy in this area, the sparse sampling in the slowly changing area can save computing resources and prevent overfitting. In addition, the selection of training datasets should meet the needs of practical

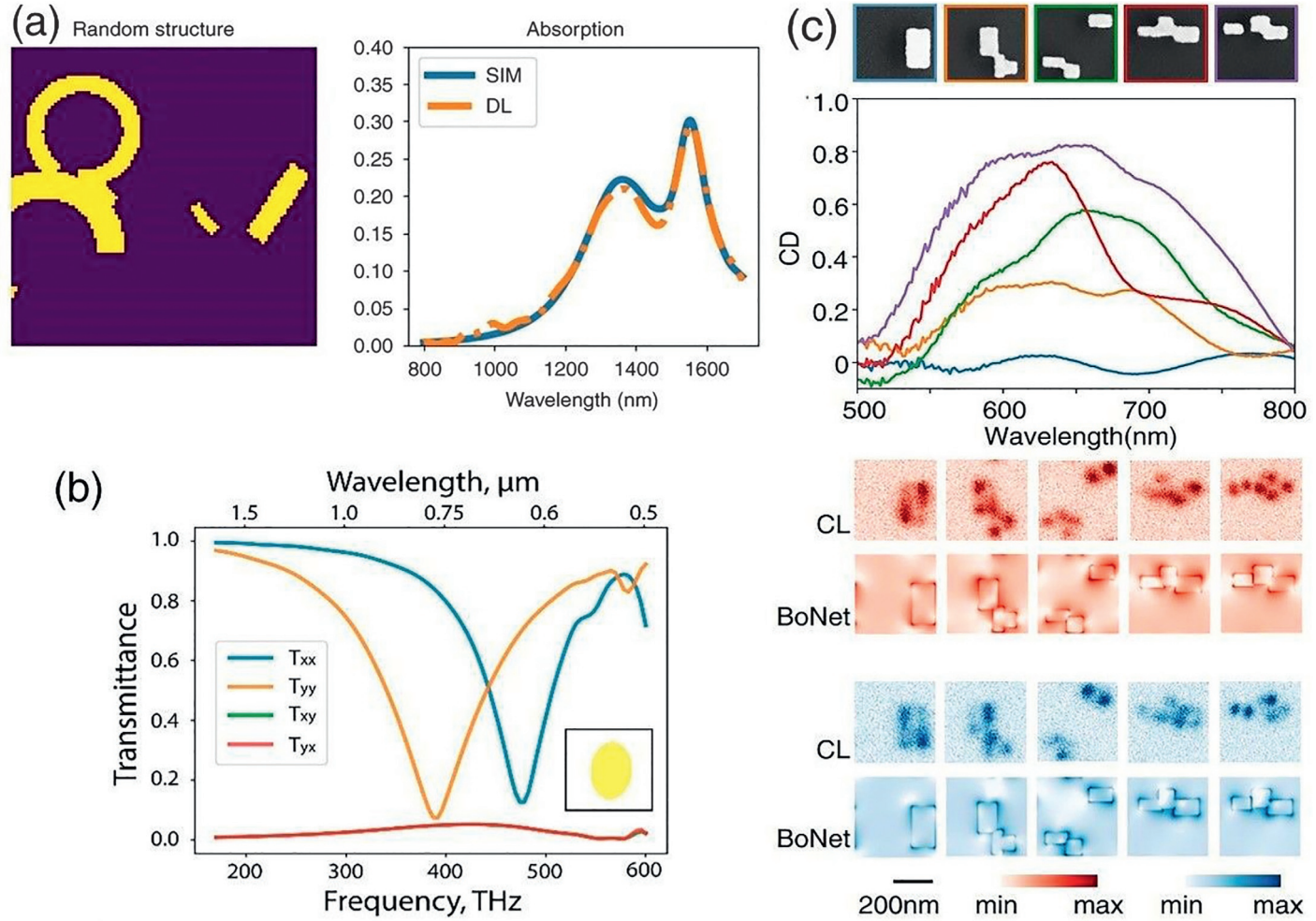


Fig. 9 (A) From top to bottom, there are five structure samples marked by five colors, the CD value of the far-field reflection spectrum, the cathodoluminescence (CL) experimental results of the left-handed circularly polarization component and the BoNet prediction structure, and the CL experimental results of the right-handed circularly polarization component and the BoNet prediction structure, correspondingly. (B) A randomly generated structure, the picture on the left is the structure picture and the absorption curve is on the right. The *orange dotted line* is the result of deep learning prediction, and the *blue solid line* is the result of numerical simulation. (C) A sample in the dataset, the inserted small image is the structure image, and the curve is the transmission spectrum of different linearly polarized incidents and transmission light. *(From 2018 American Chemical Society, 2019 American Physical Society.)*

problems. For example, in the automatic driving system based on artificial intelligence, the probability of complex road conditions that may lead to accidents during the operation of products may not be high, but the training dataset should increase this situation to improve the ability of the system to cope with dangerous and complex road conditions. After sufficient epochs of training of the verification set and training set, the prediction ability of the neural network needs a test dataset or experimental results to verify. As shown in Fig. 9C, in the work that obtains high CD structures by CNN combined with Bayesian optimization, after the CNN successfully finds the structure with high CD, experiments are carried out to verify the algorithm platform, and good experimental results are obtained.

3.2 Ultrafast spectroscopy signal analysis

Besides the mapping between the structure matrix and the spectral curve, the CNN can be applied in new physical fields by adding a new physical quantity. The curve of light intensity with frequency can be discretized into vectors, and the two-dimensional matrix of the time-frequency domain can be obtained after adding the time dimension [22]. The frequency-resolved optical gating (FROG) trace, which represents the ultrafast pulse structure in the time and frequency domain, is a picture of intensity distribution with time and frequency on the longitudinal axis. As shown in Fig. 10A, the CNN can map the FROG trace with the pulse spectrum curve, so the CNN can overcome the problem of ultrafast pulse characterization.

A computing platform for mapping the second harmonic generation (SHG) FROG trace of an ultrafast pulse with the curve of absolute field and phase change with time by CNN is proposed. DeepFROG, the neural network algorithm platform in this work, consists of two modules, FROG net and CNN, and realizes a supervised learning network (Fig. 10B) and an unsupervised learning network (Fig. 10C). The loss function of the total network is obtained by adding the loss functions of the supervised learning module (the former term) and the unsupervised learning module (the latter term), and the proportion of them in the total loss function is controlled by λ.

The function of the FROG net is to obtain the FROG trace according to the pulse spectrum curve, which is defined as follows:

$$\begin{aligned} w^* &= \underset{w}{\operatorname{argmin}}\{\operatorname{loss}(I, E)\} \\ &= \underset{w}{\operatorname{argmin}}\left\{ \begin{array}{l} \|\mathrm{CNN}(I; w) - E\|_1 \\ + \lambda\|\mathrm{FROGNet}(\mathrm{CNN}(I; w)) - I\|_1 \end{array} \right\} \end{aligned} \tag{2}$$

As a pretrained network, the FROG net does not adjust the weight in the process of obtaining the dataset and keeps the internal structure and parameters constant. There are

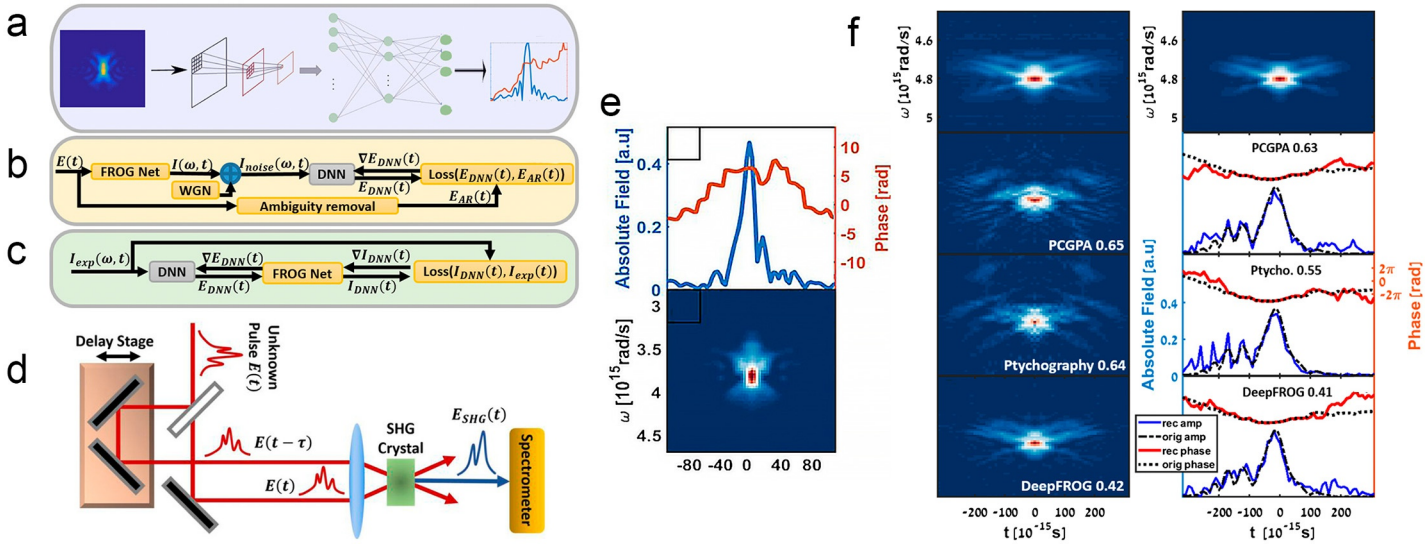


Fig. 10 (A) The FROG net is transformed into a pulse after CNN and other neural networks. (B) Supervised learning network architecture, in which WGN is Gaussian white noise. (C) Unsupervised learning network architecture. (D) Schematic of the experimental device to get the FROG net. (E) A sample in the dataset was calculated by numerical simulation, including pulse and FROG net. (F) The figures on the left side from top to bottom are the result of experimental measurement of the FROG trace, the result of the numerical simulation of PCGPA and ptychography, and the prediction of the FROG trace by DeepFROG. The figures on the right side from top to bottom are the standard FROG net, the pulse reconstructed by PCGPA and ptychography, and the pulse reconstructed by DeepFROG, in which the *black dotted line* is the original reference pulse. The number calibrated is the error between the reconstructed result and the standard sample. *(From 2018 Optical Society of America.)*

60 K samples for training and 10 K samples for testing in the dataset generated by FROG net. Because the algorithm of FROG trace involves linear transformation such as Fourier transform, the fully connected layer is used to complete the function. To obtain the FROG trace, the pulse $\widetilde{E}(\omega) = \sqrt{S(\omega)}e^{i\phi(\omega)}$ is generated first, in which $S(\omega)$ is Gaussian white noise and $\phi(\omega)$ is the randomly generated phase vector with the high-frequency component filtered out. To avoid the multiple solutions problem that one FROG trace corresponds to multiple pulses, the $E(t)$ obtained by Fourier transform is mapped to its corresponding pulse $E_{AR}(t)$ through specific corresponding rules while the original pulse is its corresponding ordinary solution. This allows the FROG trace to be obtained from the pulse. This step is the ambiguity removal shown in Fig. 10B.

In the CNN module, after the data samples pass through the convolution layer and the fully connected layer, the FROG trace as a matrix is transformed into a one-dimensional vector, namely the discrete pulse spectrum curve. To predict the pulse, three different CNN schemes were tried, and the best one was selected. The final algorithm module is based on the densely connected continuous network, that is, a short cut is introduced between each convolution layer and the deeper convolution layer, which makes the nonadjacent convolution layers fully connected so that the gradient information can be transferred directly to the deeper convolution layer. This leads to better results. Fig. 10D is the experimental device for measuring the FROG trace and Fig. 10E is one numerical simulation sample from the dataset.

After CNN training, the test and experimental verification are carried out. Fig. 10F shows the test details for a sample at a low signal-to-noise ratio (SNR). The left top figure is the experimental measurement of the FROG trace. The left middle two are the results of two mature numerical simulation methods of computing FROG trace, which are the principal component general projections algorithm (PCGPA) and ptychography, and the errors with the standard data are 0.61 and 0.59, respectively. The left bottom figure is the FROG trace predicted by CNN in DeepFROG, and the error with the standard data is 0.22. The top right figure is the original reference FROG trace. The right middle two figures are the pulse curves reconstructed by PCGPA with the error of 0.49 from standard data, and by ptychography with the error of 0.45 from the standard data. The right bottom figure is the pulse curve reconstructed by DeepFROG, and the error with the standard data is 0.21. The dotted lines in the last three graphs are the original reference pulse curves. Therefore, compared with PCGPA and ptychography, the reconstructed pulse curve and the FROG trace are more accurate. The number in the reconstruction result graph is the error with the reference pulse. This work realized the FROG net platform from supervised learning to unsupervised learning, which is used to predict the pulse from the FROG trace and still has a good performance in the lower SNR.

4. Generative models for spectra design

How to predict a credible spectrum of structure using a CNN is explained in Section 3. In this section, a more practical application problem is discussed—spectral reverse design, To solve this reverse structure design problem with CNN, we first explain upsampling and unpooling, two important kinds of mapping relationships between the spectral vector and the structure matrix. We then discuss the principle of generative adversarial networks (GANs) and their variant called progressive growing of GANs (PGGANs) as well as two practical applications. Another important branch of the generation model as autoencoders will be discussed in the next section.

4.1 Generative CNNs

Generally, a spectrum usually can be described by a vector, but a structure is often parameterized by a matrix. For example, to design a transmission spectrum of a 400 nm periodic structure, if the design precision is 5 nm, the structure can be parameterized by an (81,81) sized 0–1 binary matrix (0 means no structure, 1 means a structure exists). The spectral bandwidth is 450 to 750 nm, with a 5 nm interval, which corresponds to a (1,61) spectral vector. Convolution layers are especially suitable for feature extraction of input matrix, while pooling operation is effective in reducing feature dimensionality, shown as the left part of Fig. 11A, the geometric patterns of structure matrix can be extracted into spectra vectors. While in the right part, through the deconvolution layers and unpooling operation, the spectra can link to a structure matrix. As shown in Fig. 11B [23], deconvolution is the inverse process of convolution, which can be also named as transposed convolution. The weights of deconvolution layers can be trained automatically. By combining deconvolution layers with unpooling operation, we can gradually increase the feature map dimension until the output size matches the input structure matrix.

Unpooling is often used in a CNN to represent the inverse operation of max-pooling. Its operation process is shown in Fig. 11C. During the unpooling process, the maximum location information is retained in the max-pooling process. Then in the unpooling process, the position information can be utilized to fill the feature map. However, upsampling is different. In this method, the location information of max-pooling is not used, but the content is copied and populated directly into the feature map.

For the problem of structural inverse design, the prediction and generation of an 0–1 parameterized structure matrix are equivalent to many algorithms in the field of computer vision for black/white or grayscale image processing. There are two important branches, autoencoders (AE) and GANs, that are well suited to migrate into the field of structural reverse design. Because the AE is a kind of semisupervised learning or unsupervised learning using artificial neural networks, it is particularly suitable for

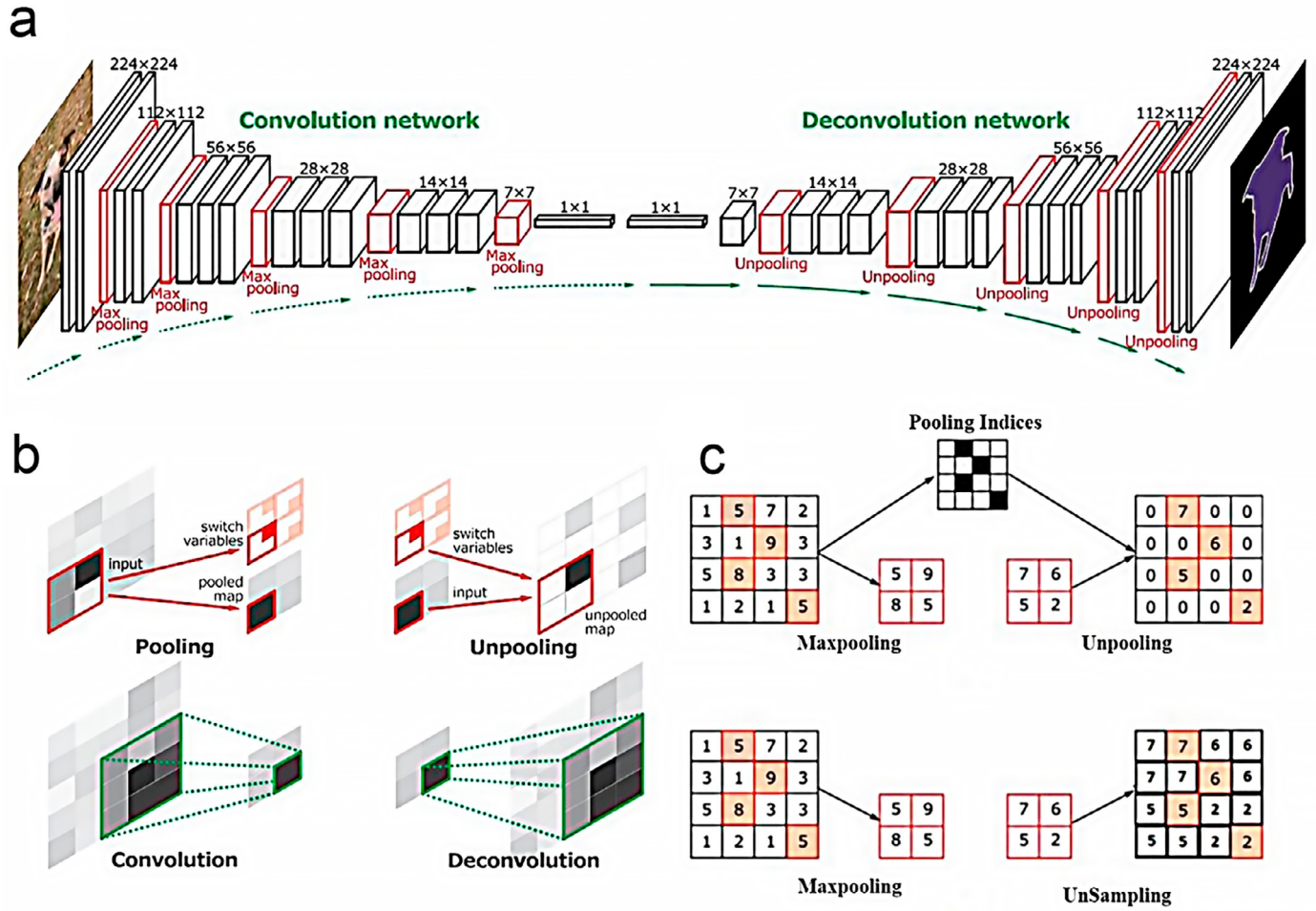


Fig. 11 (A) Deep convolution and deconvolution neural network architecture. In the right part, spectral vectors are expanded into structural images through deconvolution and unpooling operations. (B) Schematic diagram of deconvolution and unpooling. (C) Schematic diagram of unpooling and upsampling.

characterizing the input information and can be migrated to the structural design in the latent space [24,25].

GANs combine generators with discriminators, and learn from real data to gradually produce fake data within the same distribution. Demonstrations of a generator and discriminator are shown in Fig. 12A. The training GAN needs to reciprocate between the generator and discriminator, where the generator randomly creates a structure and the

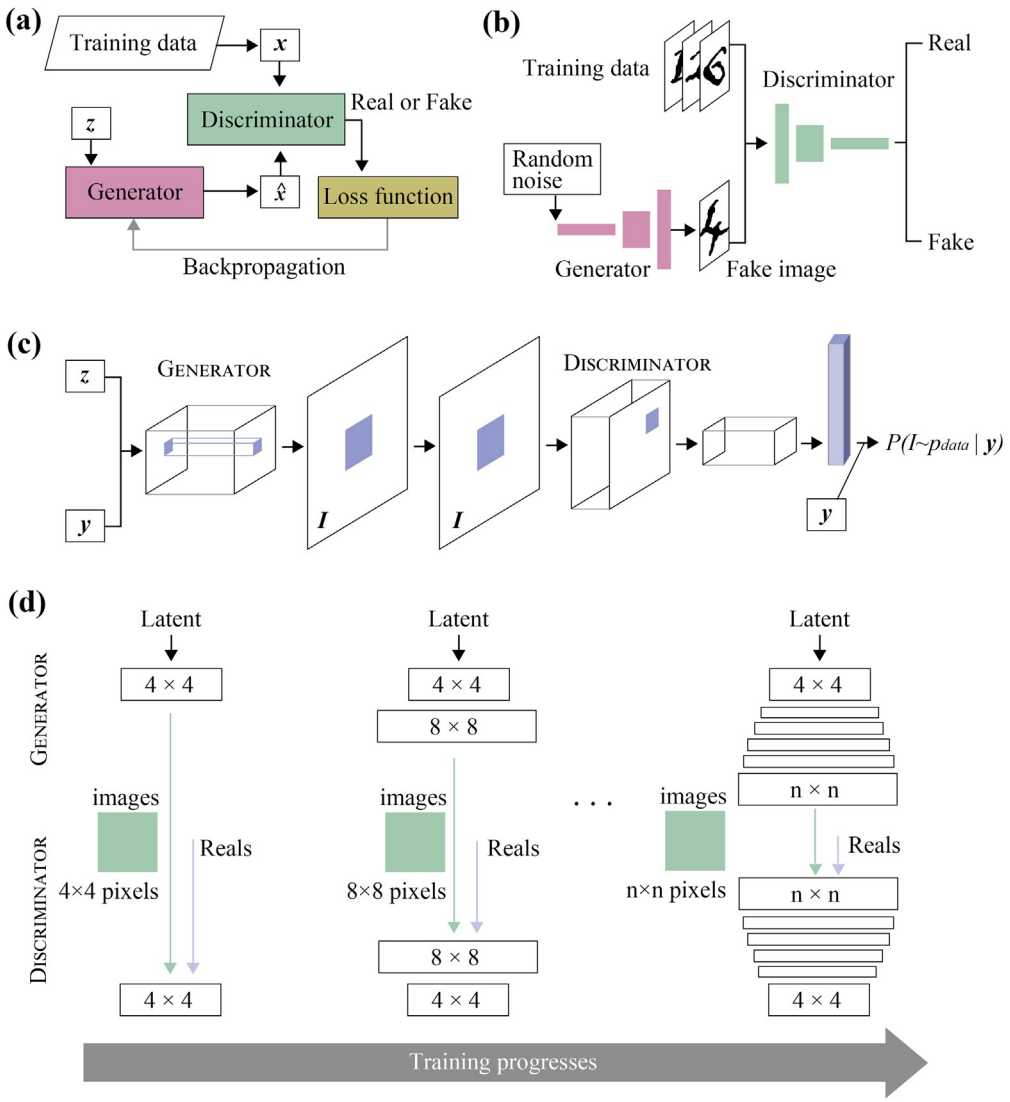


Fig. 12 Generating adversarial network (GAN). (A) Sketch of GAN training process. (B) Sketch of generator and discriminant in one epoch. (C) Sketch of CGAN. (D) Sketch of PGGAN training process.

discriminator needs to judge whether the structure is real or fake. If the discriminator judges correctly, the generator needs to evolve through learning from a sample distribution, and strives to generate a better structure to deceive the discriminator in the next generation. Generator and discriminator are competitive relationships, shown as Fig. 12B. The generator generates fake handwritten numbers from random noise while discriminator needs to judge this number is real or fake according to the real handwritten numbers distribution. A more popular explanation of the generator can be understood as counterfeiting, while discriminator can be understood as money detectors. Because of the existence of money detector, the generator needs to constantly improve the level of manufacturing counterfeit money, and counterfeit detector also enhances the ability to identify genuine and counterfeit banknotes, after several rounds of competition, counterfeit money that deceives ordinary people can be made.

The training process of a typical GAN can be referred to as follows:

(1) Create a dataset $P_{\text{data}}(x)$, where x represents a sample corresponding to structure and its spectral response, which can be obtained by numerical simulations.

(2) Initialization parameters for generator θ_G and discriminator θ_D.

(3) Select samples $\{x^1, x^2, x^3 \cdots x^m\}$ acquired from $P_{\text{data}}(x)$, and m is the batch size, which needs to be selected according to the situation and can generally be set as 32 or 64.

(4) Generate a noise vector $\{z^1, z^2, z^3 \cdots z^m\}$ sample from a distribution, such as Gaussian or normal distribution.

(5) Taking noise vector z in Step 4 as the input, get generated data through the generator $\left\{\widehat{x}^1, \widehat{x}^2, \widehat{x}^3 \cdots \widehat{x}^m\right\}, \widehat{x}^i = G(z^i)$.

(6) Update θ_D for the discriminator D to maximize $\widehat{\mathrm{V}}$.

$$\mathrm{Max}\left(\hat{\mathrm{V}} = \frac{1}{m}\sum_{i=1}^{m}\log D\left(x^i\right) + \frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D(\hat{x}^i)\right)\right) \tag{3}$$

$$\theta_D \leftarrow \theta_D + \varepsilon\nabla\widehat{\mathrm{V}}\left(\theta_D\right) \tag{4}$$

In this function, the discriminator D is a binary classifier hoping to maximize the distance between the real distribution and the fake distribution acquired from the generator. Maximizing $\widehat{\mathrm{V}}$ is to force $D(x^i)$ to approach 1 while $D\left(\widehat{x}^i\right)$ approaches 0. ε is a hyperparameter that can be adjusted according to the actual design problem and can be set as 0.001 in general.

(7) Cycle Steps 3–6 to train discriminator D, which can be updated several times to achieve a better optimization effect.

(8) Sample noise is extracted from a distribution with m vectors $\{z^1, z^2, z^3 \cdots z^m\}$. The noise vector $\boldsymbol{z}$ here needs to satisfy the same distribution in Step 5.

(9) Update θ_G for the generator to minimize $\widehat{V}$.
(10) Repeat Steps 3–9 until the generator achieves a good result.

In the spectral structure design, finding the structure design consistent with the target spectral response quickly and stably through the NN is expected. This structure design algorithm can be completed by CGAN [26], as shown in Fig. 12C. In addition to the noise vector, CGAN inputs the required spectral data into the generator as a condition, so that the generated structure can satisfy the expected spectral response. It is a challenging task to directly use GAN to generate a high-definition structure because the training process has poor stability. To solve this problem, a GANs variant-PGGAN—has been developed [27]. As shown in Fig. 12D, PGGAN divides the learning process into several periods to gradually improve the resolution. For example, expecting to design a periodic structure of 1 μm × 1 μm with a 5 nm interval number of pixels of structure can be parameterized as 201 × 201. PGGAN can be used to gradually generate the structure matrix 11 × 11 with a 100 nm interval. Then it transitions to 51 × 51, 101 × 101, and finally arrives at 201 × 201, which can effectively improve the stability in NN training. Detailed examples of CGAN and PGGAN applications are discussed in Section 4.2.

4.2 Applications of GCNN

In this section, two typical examples of GCNN are introduced to show how to use CGAN and PGGAN to realize the spectral structure design. The optical metasurface can be designed to customize the transmittance and phase for any wavelength by careful structure design. Many important applications have been realized, such as a perfect absorber, super-resolution imaging, beam manipulation, and nonlinear optical devices. In traditional structural design, finite element modeling (FEM) or finite-difference time-domain (FDTD) methods are often used to calculate the structure spectrum, and then optimization algorithms are used to find a better structure design. These methods require a large number of computational resources, and parametric scan methods limit the optimization freedom. Using DNNs can greatly open the freedom degree of reverse design and generate a structural matrix rather than settled parameters of the structure [28,29].

As shown in Fig. 13, a gold metasurface with a thickness of 50 nm and a period of 340 nm on a glass substrate is constructed by a CGAN algorithm to generate nanostructures with specific spectra, such as the four spectra of different polarization incidence and different polarization emission shown in the figure. The DNN is used to predict the spectrum from the structure and generates the structure from the spectrum [20]. The CGAN architecture of the structure design task is shown in Fig. 13B. The network can be divided into three parts: the simulator (S), generator (G), and discriminator (D). The basic principle of CGAN is described in Section 4.1. G generates specific structures through data noise and corresponding spectral conditions. D determines whether the generated structures are real or fake by designing the geometric distribution of

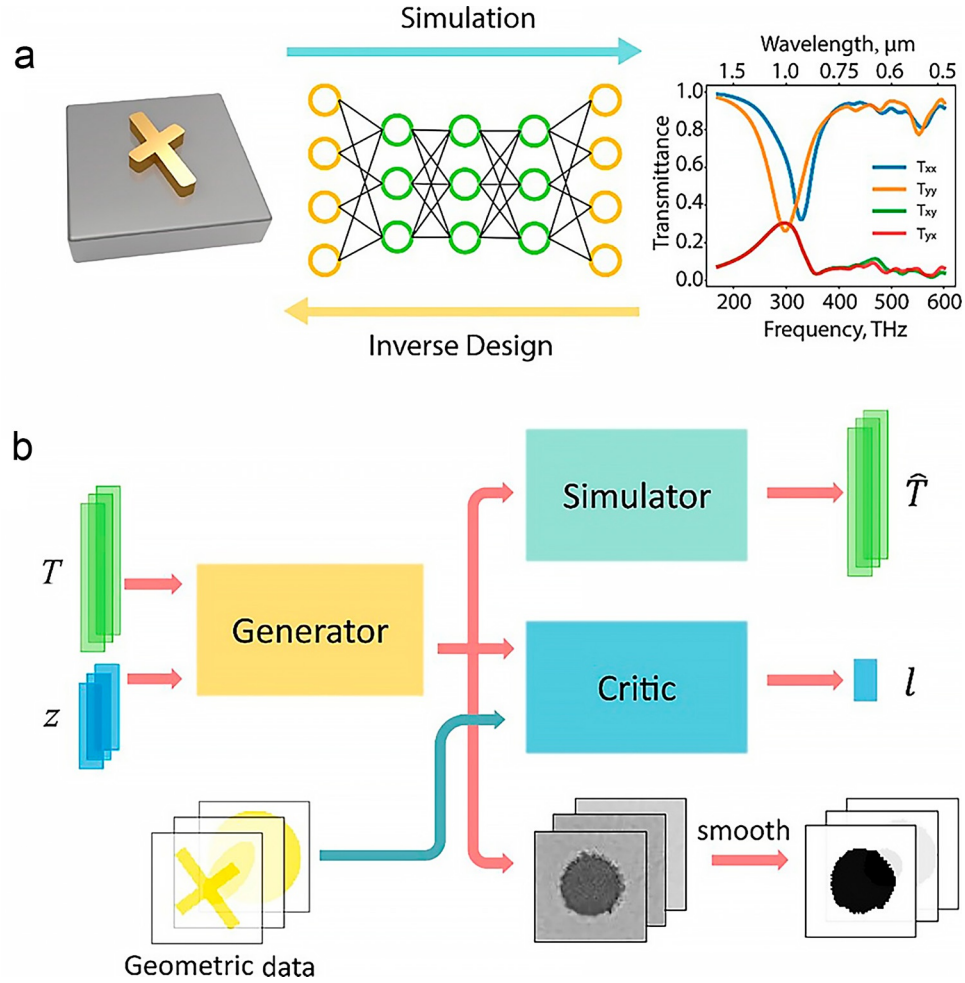


Fig. 13 Spectral design of the metasurface using CGAN. (A) The structure-spectral correspondence of the metasurface; the forward process is the spectrum simulation and the backward process is the reverse design replaced by the deep neural network. (B) The network architecture based on CGAN optical structure design. *(From 2018 American Chemical Society.)*

the structure in advance, forcing the structure generated by the generator to be like the structural dataset such as purposely generating an ellipse, rectangle, or cross structure. The simulator needs to be executed by the NN corresponding to the pretrained structure spectrum, which can be set up by referring to the third section of this chapter. By adding the simulator to the corresponding loss function, the structure generated by the generator can be forced to have the same response spectrum corresponding to the input

conditions. During the output of the structure, Gaussian filters can be added to the last layer of the NN to smooth the generated structure to make the generated structure more convenient for experiments.

The discriminator is important for structure generation. As shown in Fig. 14A, under the guidance of adding a cross-shaped structure information to the discriminator, the generated structure can be gradually changed into a cross-shaped structure. However, after the discriminator is removed, the interpretability for the generated structure is lost. The generator can find multiple structures for the same spectrum, which makes it difficult for the generator to choose which structure is better. Both the simulator and the discriminator have a certain forward guidance effect to ensure that the generated structures maintain high design freedom and can be easily fabricated in the experiment. The most ideal input for the discriminator should be defined by the dataset, including various easily fabricated structures. However, this dataset is difficult to find, so it is often guided by a prior specific geometric structure such as a cross, ellipse, etc. The performance of the network is shown in Fig. 14B–E. Fig. 14B shows the spectrum of the particular number 5 with handwritten numbers as the data input. In the generator process, as shown in Fig. 14C, number 5 is excluded from the dataset. The generated structure is number 3, which has a spectrum that is in good agreement with the target spectrum. When researchers need to designa structure with the target spectral, for example, a gaussian spectrum in Fig. 14D. Even there is no satisfying structure in the training data, GAN can still extract rules from the bidirectional mapping relationship between structure and spectrum and generate a structure which can roughly satisfy the target spectrum in Fig. 14E.

The following part is a practical application of a PGGAN. The PGGAN has better training stability than GAN, which is especially suitable for the design of structures with large pixel matrices. As shown in Fig. 15A, a metagrating structure is expected to emit light at a specific angle under the condition of incident light with a specific wavelength. We use a glass substrate and the dielectric material Si to parameterize the structure by an 0–1 matrix, where 1 represents silicon and 0 represents air. Fig. 15B shows the PGGAN train cycles, firstly, using the topology optimization method get structures in different wavelength and different diffraction Angle to prepare the training data set, and then through the DNN training, after the generator generated structures, using electromagnetic field simulation software calculate its optical response, and then select efficient super grating structure imported into the training set for data update, cycled several times to get the final structure generation [30].

The framework and training cycles of the PGGAN network are shown in Fig. 16A–C. In the initial cycle, the wavelength, diffraction angle, and random noise are fed into the network, and then it is expected to obtain an 8×16 pixel structure matrix

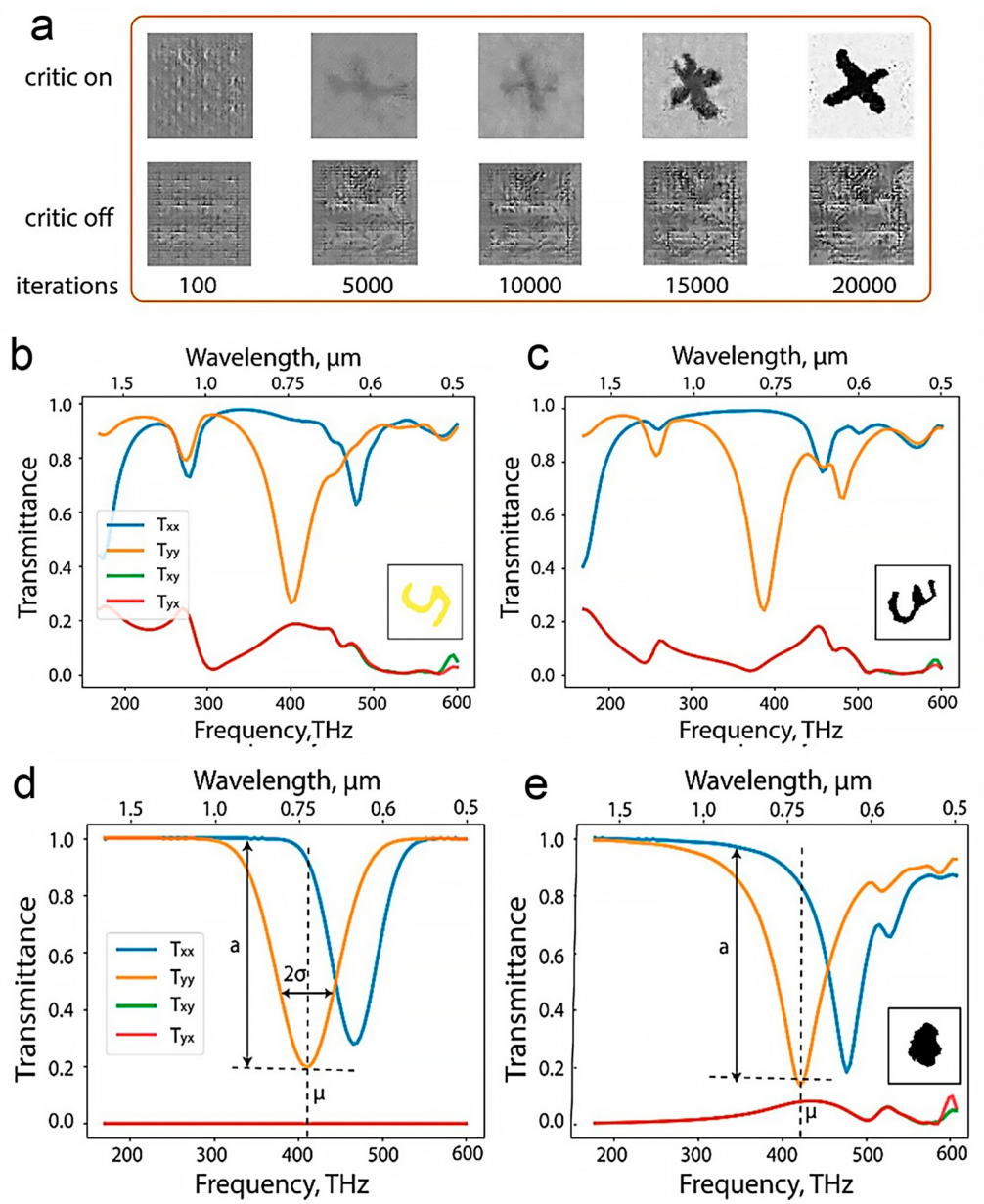


Fig. 14 Display of results of CGAN metasurface structure design. (A) Influence of discriminator in structure generation. (B, C) Handwritten number dataset as a preinput for the discriminator; the corresponding structure generated by the generator. (D, E) Artificial designed Gaussian input spectrum; generator generated with the corresponding structure design. *(From 2018 American Chemical Society.)*

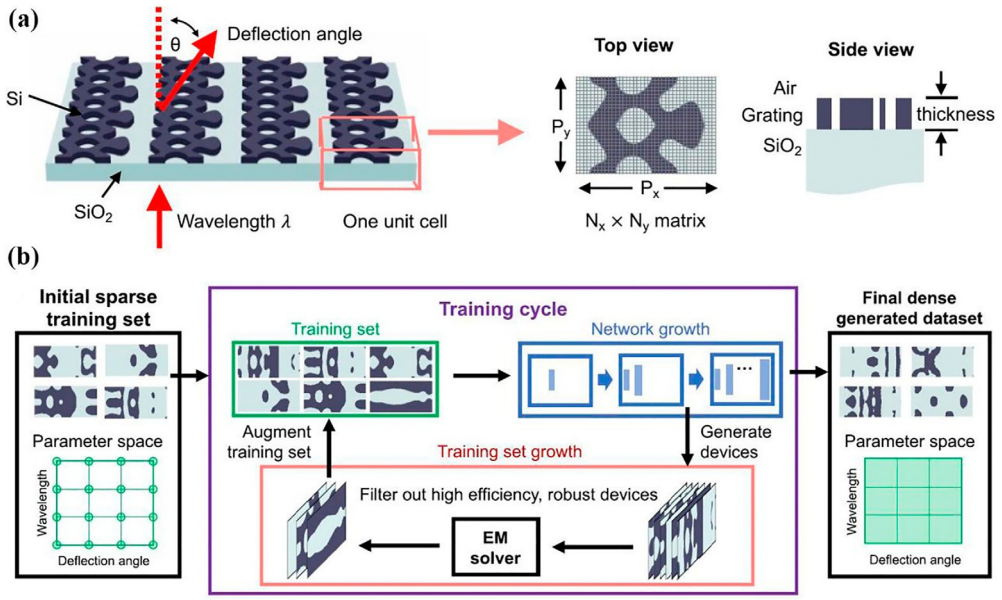


Fig. 15 (A) Diffraction metagrating composed of Si/SiO_2; a 3D schematic diagram, and its top and side views. (B) PGGAN network framework and training process. *(From © 2020 American Chemical Society.)*

with a large interval. The initial structure is generated through deconvolution and the self-attention layer, and then the discriminator is used for recognition, performing a classical GAN training process. The computation rules for a self-attention layer are shown in Fig. 16D. Structures are transformed into vectors and then extracted from the feature map. The attention layer is an essential supplement to the convolution layer for extracting global structural information. Then, we use electromagnetic field simulation to obtain the optical response for the generated structures of each generation. Better structures are selected to update the dataset. More detailed structural design can be gradually obtained through cyclic iteration. The generator in PGGAN combines deconvolution and upsampling to generate structures. The upsampling process is directly filled for the structure, which is the same as the structures in the last cycle, and the deconvolution process tends to generate more elaborate structures as a new cycle of the training process of GAN. Alpha increases gradually to decrease the upsampling proportion, making the structures update with high stability. Thus, the structure design from the relatively rough structure can be smoothly transferred to the elaborate structure design. Similarly, the same improvements are the same for the discriminator to accommodate the training process of PGAGN. The final efficiency of the structures generated by PGAGN is shown in Fig. 16E. Compared with GAN, PGAGN shows great stability for structure optimization with the increase of training cycles.

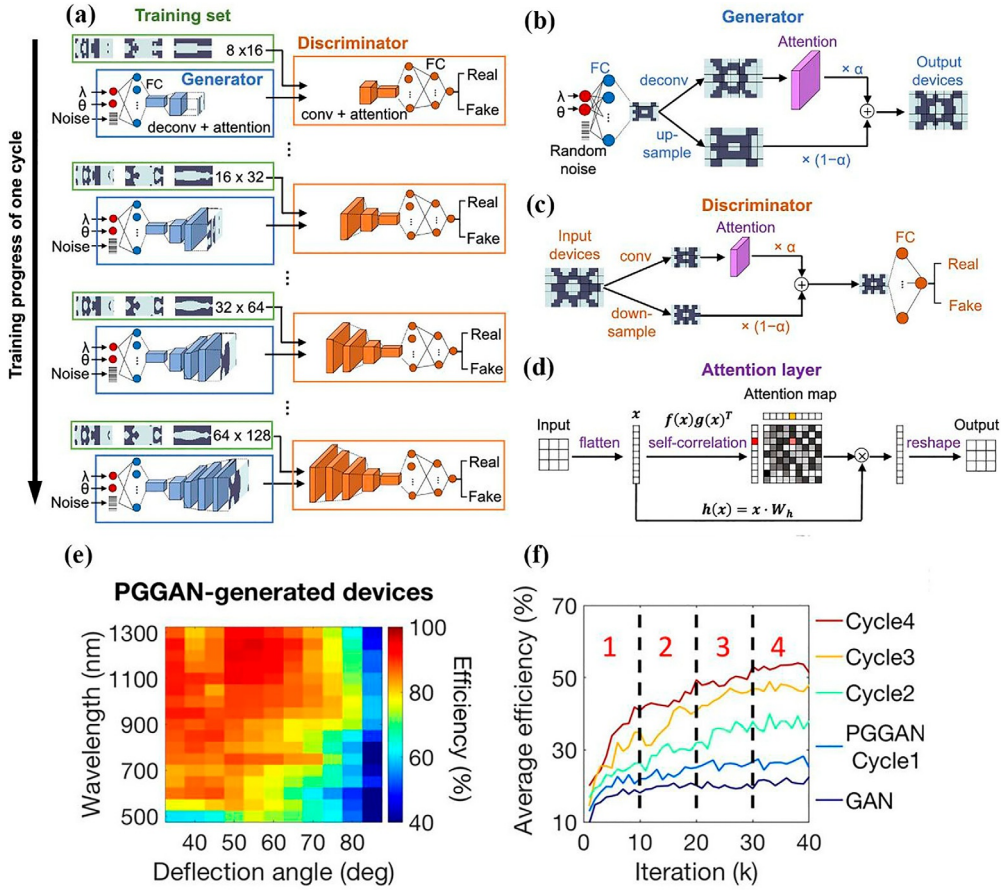


Fig. 16 The training cycle and results of the PGGAN network. (A) Schematic diagram of a single cycle of network training, and the resolution of the generator and discriminator gradually increases. (B, C) Training processes of the generator and discriminator. (D) Principle for the self-attention layer. (E) Efficiency display of a PGGAN-generated structure. (F) Average efficiency comparison between PGAGN and GAN. *(From © 2020 American Chemical Society.)*

5. Dimensionality reduction models for optical property extraction

For the models in Chapter 2, the training data include the nanostructure matrix, and the output is the spectral response obtained by experiments or numerical simulations. These methods belong to supervised learning. It can utilize the relationship between specific structure symmetry and spectra for data expansion, or apply other machine learning algorithms for efficient sampling. In general, the above models still need to provide a large amount of training data to achieve high-precision prediction, and it is easy to produce overfitting or underfitting problems in complex coupling systems. Another strategy is

not to solve the mapping relationship between the structure and spectral response, but to find the patterned relationship or clustering features between variables, which is called unsupervised learning. To realize unsupervised feature recognition of nanostructures, a CNN can be used to build dimensionality reduction models to compress the sparse features from high-dimensional space into low-dimensional space [31]. Through the dimensionality reduction methods, a CNN can not only realize spectral prediction and reverse design, but also realize spectral pattern recognition and other tasks of assisting researchers in physical analysis.

5.1 Unsupervised learning with dimensionality reduction models

The sample distribution can be very sparse in high-dimensional variable space, thus the distance between individuals is long and the variation characteristics of parameters are discontinuous; this is not conducive to recognizing patterns from the original feature space. By using the dimensionality reduction method, a large number of sparse features can be compressed while retaining original feature information as much as possible [32]. Before CNNs were widely used, dimensionality reduction methods based on projection, manifold learning, principal component analysis (PCA) [33], and supervised learning-based linear discriminant analysis (LDA) were used [34]. These have shown application value in some spectroscopic analysis tasks. Among them, traditional unsupervised dimensionality reduction algorithms such as the PCA are used to find the low-dimensional hyperplane with minimal projection distances of the dataset in the original space and then to find the sample correspondence after dimensionality reduction in the projection space. This algorithm can maximize the difference of training data, which is also the variance, and retain the first n_{th} PC according to the requirements of the algorithm for dimension retention and the choice of compression accuracy. However, PCA can only process parameters in vector format, which is not compatible with complex structures defined by the matrix.

As discussed in Chapter 2, a CNN can manipulate the number of output features by max-pooling layers and controlling the number of channels, and the feature relationship between adjacent pixels is stored in the convolutional kernels. Therefore, by building a CNN with the appropriate number of max-pooling layers and convolutional layers, the dimension of data features with the matrix structure can be gradually reduced and mapped into a low-dimensional space called latent space, which realizes a dimensionality reduction model. Because the training purpose of this kind of NN is to minimize the difference between output and input, these models are also called autoencoders.

A basic convolutional autoencoder includes an encoder with max-pooling layers and a decoder with upsampling layers, which are designed for suppressing original features into lower dimensions and reconstructing original features from suppressed space, respectively. It should be noted that these functions are often realized by several blocks of hidden layers that consist of several convolutional layers and one layer of a max-pooling layer or upsampling layer.

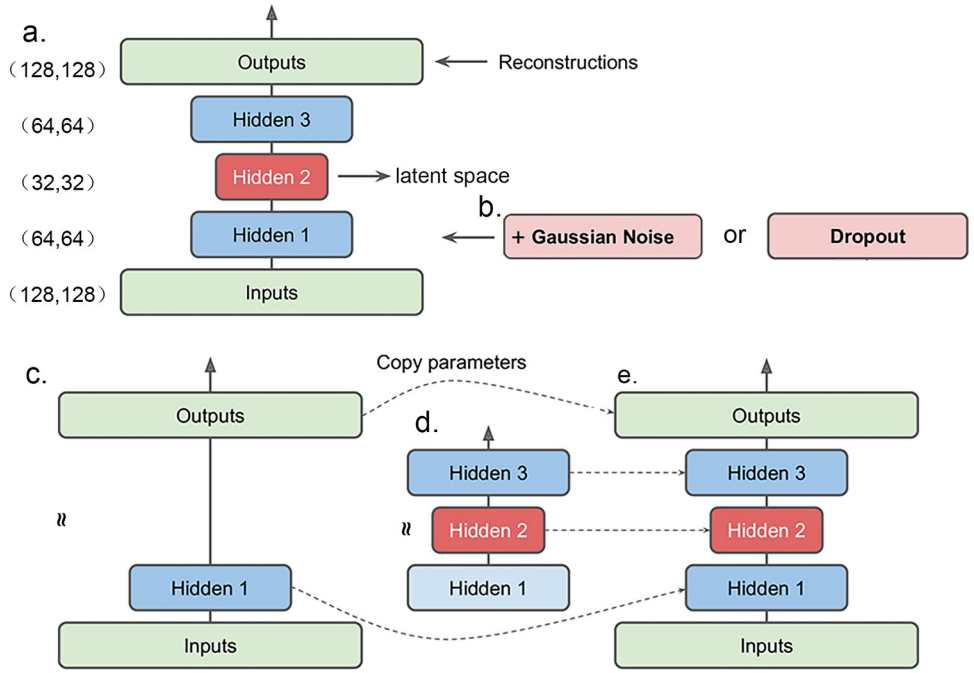


Fig. 17 (A) Demonstration of a typical convolutional autoencoder. (B) Introducing Gaussian noise or the dropout method to realize a denoising autoencoder. (C-E) A layer-by-layer training method for autoencoders.

As shown in Fig. 17A, the input data with the feature size of (128,128) can be compressed into the size of (64,64) by the first block of hidden layers (Hidden1), and the suppressed feature map with a size of (32,32) can be finally obtained at the bottleneck of the network (Hidden2), which is only 1/16 of the input feature size. Finally, by stacking additional hidden layers or jointing with FCNN, the feature space dimension can be compressed to even one dimension. Generally speaking, the higher the feature dimension of bottleneck reserves, the lower the possibility of information loss.

As shown in Fig. 17B, autoencoders can be used as noise attenuators by intentionally introduce Gaussian noise into some feature dimensions of inputs to get x', where the expected value of noise is 0 and brings no bias to the model [35]. Meanwhile, the original data x without noise is maintained as the training target to realize the minimum of $\text{Loss}_{x}{}'$. This method forces the autoencoder to ignore the noise in x' and extract the most important features as well as robust representations from the input data. In addition, dropout can also be added to randomly disable some neurons at the cost of decreasing the fitting ability of the model to reduce the possibility of NN overfitting or learning distracting noise.

In practice, convolutional autoencoders often consist of more than 20 convolutional layers with the number of trainable parameters in the order of millions or even tens of millions. Those models may perform better but can also be hard to train because gradient explosion or gradient disappearance tends to appear in deep models. Therefore, in training, a layer-by-layer training method can be adopted. In this method, each training round only influences one hidden block (Hidden1, as shown in Fig. 17C). After the training process of Hidden1, both the input layer and output layer are discarded while the input layer of the next layer is replaced by Hidden1. By repeating the circulation, complex models with dozens of layers can be trained from shallow to deep (Fig. 17D and E). After circulation, all the hidden layer structures and parameters are extracted as the initialization parameters of the deep autoencoder and the final weights can be acquired with a few rounds of fine-tuning.

Apart from dimensionality reduction, convolutional autoencoders can also be extended to implement generative models and realize the structural design (similar to Chapter 4). Among them, a variational autoencoder (VAE) is a well-performed generative network architecture based on the variational Bayes (VB) method.

Different from conventional autoencoder models, VAE suppresses the original features into data distributions in latent space rather than a specific variable, which realizes a continuous and smooth expression of the latent space and shows exceptional potential for generating new designs [24]. For an original dataset $X=\{x_i\}_{i=1}^{N}$, of which each sample is randomly generated and the parameters are independent variables, the target of the autoencoder is suppressing features into low-dimensional latent space and reconstructing the generative dataset $X'=\{x_i'\}_{i=1}^{N}$. When analyzing the suppression process from the perspective of Bayesian statistics, the encoder is a posterior distribution inference of latent variable z given the condition of original sample x:

$$q_{\varnothing}(z|x) \tag{5}$$

Then, the conditional distribution of the corresponding decoder model is:

$$P_{\theta}(z)P_{\theta}(x'|z) \tag{6}$$

Given the input data of x, the output of VAE should be the posterior distribution of Pzx. Generally speaking, it is hard to calculate Pzx directly, thus an alternative distribution of $q(z|x)$ fitted by NN is applied to approach Pzx. To describe the similarity between the two distributions, the Kullback-Leibler divergence (DKL) is introduced for quantification:

$$\min D_{\mathrm{KL}}(q(z|x)\|p(z|x)) \tag{7}$$

Considering the normalization of probability distribution:

$$\sum_z q(z|x) = 1 \tag{8}$$

The logarithmic form of the function is introduced as:

$$L = \log\left(p(x)\right) \tag{9}$$

$$= \sum_z q(z|x) \log\left(\frac{p(z,x)}{p(z|x)}\right) \tag{10}$$

$$= \sum_z q(z|x) \log\left(\frac{p(z,x)}{q(z|x)} \frac{q(z|x)}{p(z|x)}\right) \tag{11}$$

$$= \sum_z q(z|x) \log\left(\frac{p(z,x)}{q(z|x)}\right) + \sum_z q(z|x) \log\left(\frac{q(z|x)}{p(z|x)}\right) \tag{12}$$

$$= L^V + D_{KL}(q(z|\,x)\|p(z|\,x)) \tag{13}$$

Because $D_{KL}(q(z\,|\,x)\|p(z\,|\,x)) \geq 0$, and the distribution of $p(x)$ can be deduced, L should be a constant, thus the target of minimizing D_{KL} can be converted to maximizing L^V, which is to minimize:

$$\sum_z q(z|x) \log\left(\frac{p(z,x)}{q(z|x)}\right) \tag{14}$$

$$= -D_{KL}(q(z|\,x)\|p(z)) + E_{q(z|\,x)}(\log(p(x|\,z))) \tag{15}$$

Because $q(z\,|\,x)$ can always be approached by NN as an encoder, and suppose the prior z as Gaussian distribution, the original target can be replaced by minimizing $D_{KL}(q(z\,|\,x)\|\,p(z))$. In the training process of NN, the latent space is constrained as a multivariate Gaussian distribution (Fig. 18), where a pair of features are defined as the expected values and variances. Apart from reconstruction performance, the minimal distance between latent space distribution and multivariate Gaussian distribution min $D_{KL}(q(z\,|\,x)\|p(z))$ is also considered in the VAE. Compared with conventional autoencoders, the reconstruction performance of VAE may be worse. However, because the latent space of VAE is a distribution rather than specific numbers, which makes sampling additional individuals possible, the generative performance of VAE is better.

5.2 Applications of dimensionality reduction models

In previous cases of CNN used in nanophotonics research and structural design, almost all training procedures were performed by the supervised learning method, which requires a

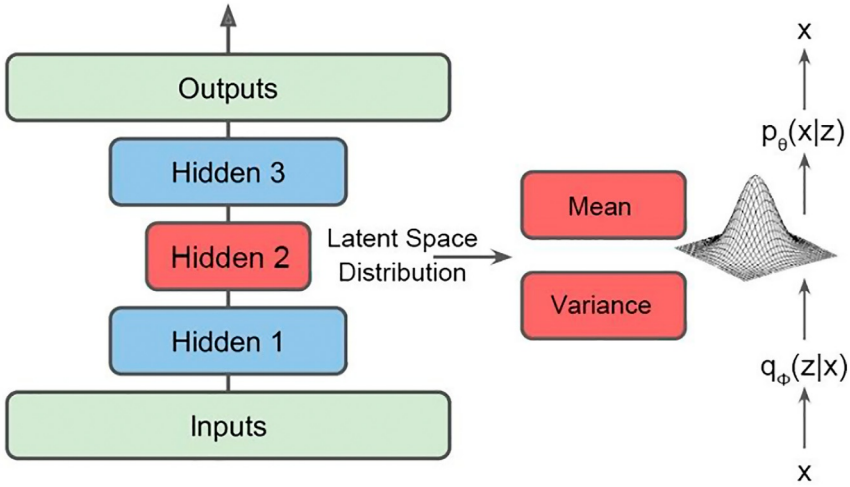


Fig. 18 Demonstration of a variational autoencoder: Features in Hidden2 are mapped into a latent space defined by Gaussian distributions.

high training set size and sampling quality. Therefore, researchers introduced VAE to reduce the dimensionality of feature space, compress structural features by unsupervised learning, and combine with supervised learning supported by a minority of structures with calculated spectra [25]. This method is called semisuperself-supervised learning and realizes both training accuracy and training efficiency.

As shown in Fig. 19A, the input designs consist of several geometries, including arc, bowtie, cross, split ring, H-shape, rectangle, L-shape, and ellipse [36]. Those structures as the unit cell of periodic arrays (period = 2 μm × 2 μm) are parameterized into a matrix with the size of (64,64). For each geometry type, 2000 simulations are performed and the total dataset is comprised of 16,000 labeled structures. Under a normal incidence of unpolarized light, the properties of reflective polarization conversion are acquired as three spectra: R_{xx}, R_{xy}, and R_{yy} (x-polarization in and x-polarization out, x-polarization in and y-polarization out, y-polarization in and y-polarization out). Under the definition of 1 Hz-step and a frequency range of 40–100 THz, the total optical properties are parameterized as three vectors with a length of 61.

To compress the features of structural designs, the (64,64) sized matrix is input into the encoder first. Different from a conventional VAE, the bottom neck of the neural network is comprised of both 20-dimensional latent distributions and 3×61-dimensional predicted spectra. Therefore, given an input design, the neural network can compress the structural features and predict the spectrum at the same time. This architecture leads to three training losses: $\text{Loss}_{\text{K-L}}$, $\text{Loss}_{\text{Spectra}}$, and $\text{Loss}_{\text{Recon}}$. $\text{Loss}_{\text{K-L}}$ defines the K-L divergence between latent space distribution and the prior Gaussian distribution. $\text{Loss}_{\text{Spectra}}$ defines the prediction accuracy of the spectra. $\text{Loss}_{\text{Recon}}$ defines the difference between

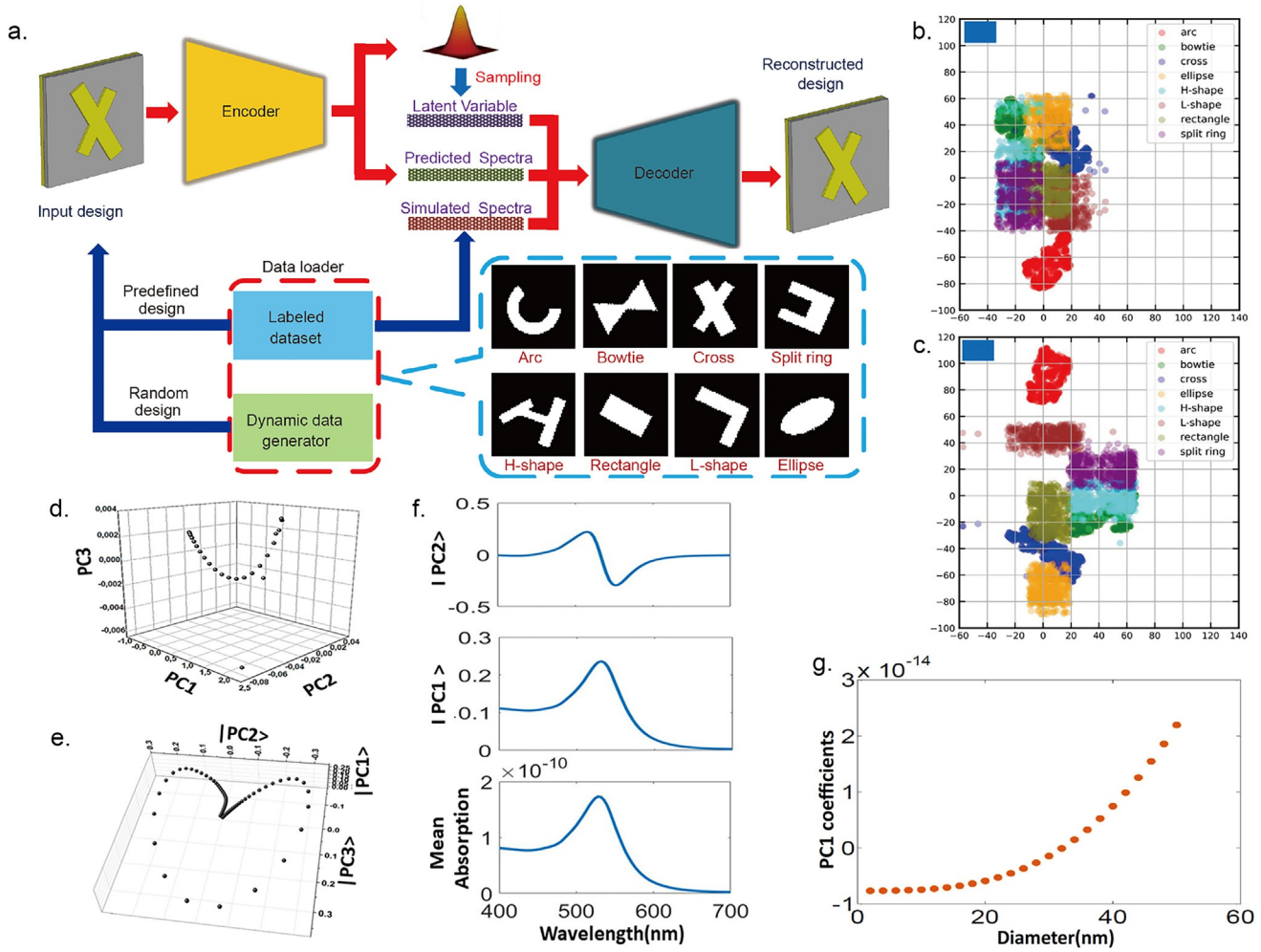


Fig. 19 (A) An architecture of VAE to realize spectra calculation and reverse design. (B) Distribution of different types of structures in the latent space of a conventional VAE. (C) Distribution of different types of structures in the latent space acquired by self-learning. (D) The compressed features of nanospheres in a 3D PC space of PC1, PC2, PC3, and PC4. (E) The distribution of a vector field in the PC space. (F) Spectra of the mean of the initial data, and the first two dominant eigenvectors (PC1 and PC2) of the covariance matrix. (G) Plot of the PC1 coefficient with the function of nanosphere diameters.

the input and the output, namely the reconstruction performance of the autoencoder. According to the choice of training losses, both supervised and unsupervised learning are used in this method.

During the loading procedure of network training, both labeled and unlabeled samples are generated with the same probability, which leads to no differences for encoders because they can still output latent distributions and predicted spectra. The bottom neck as the input of the decoder is slightly different. For labeled data with spectra acquired by numerical simulations, the ground truth spectra are chosen as input to ensure the generation performance. For unlabeled data without any simulated spectra, the encoder-predicted spectra are input as pseudolabels, which can decrease the cost of simulation calculation. By integrating unlabeled data as the training set, self-supervised learning is achieved, which realizes the balance between training accuracy and sampling cost. When executing the gradient descent algorithm, the $\text{Loss}_{\text{Spectra}}$ is only contributed by labeled samples because there is no ground truth for unlabeled samples. However, unlabeled data contribute both $\text{Loss}_{\text{Recon}}$ and $\text{Loss}_{\text{K-L}}$ by combining the total training loss as:

$$\text{Loss}_{\text{total}} = \text{Loss}_{\text{K-L}} + \text{Loss}_{\text{Recon}} + \alpha * \text{Loss}_{\text{Spectra}} \tag{16}$$

The shared network weights are tuned simultaneously to minimize all three kinds of loss; the prediction accuracy can be improved by even unlabeled data. Moreover, the features of structures suppressed from (64,64) into 20-dimensional latent space exhibit exceptional information. A neighbor embedding algorithm is performed to further reduce the dimensions of latent space into two for visualization without changing the raw 20-dimensional data structure (Fig. 19B and C). When suppressing features with conventional VAE, samples in the latent space, as shown in Fig. 19B, form clusters according to their specific geometry, which shows the ability of networks to extract structural information with an unsupervised method. Meanwhile, when unlabeled pseudosamples are integrated, clusters are better dispersed, as shown in Fig. 19C. In the classification problem, this means a smaller misclassification rate. Therefore, by combining the geometric features of major unlabeled samples with the spectra of minor labeled samples, a balance between sampling cost and model performance can be achieved. This can also provide new ideas for interpretation of the specific electromagnetic mode and promote data-driven spectra analysis.

Although there is little work on spectral research using the CNN dimension reduction model, the unsupervised learning method between nanosphere properties and spectral response characteristics using neural networks or other machine learning algorithms has been discussed [37]. When analyzing a database comprised of 25 spectra of gold nanospheres with diameters from 2 to 50 nm, PCA is employed. By detecting subspaces that contain vectors with the largest variances, a set of orthogonal eigenvectors is defined as PCs, and original samples can be expressed by the linear combination of PCs. As shown in Fig. 19D, when parameters of nanospheres are suppressed in a 3D subspace, PC

coefficients exhibit a parabola shape. Under the illumination of light, Fano resonances from the scattering component and Poynting vector bifurcations are often accompanied. When analyzing the spectra plot in the subspace, a homoclinic orbit with a fixed saddle point is exhibited in Fig. 19E, which indicates the existence of chaotic strange attractors. By representing mean absorbance and spectra with eigenvectors, an asymmetric Fano line shape is observed in the PC2 vector space (Fig. 19F), which indicates that the dimensionality reduction model can extract the Fano resonances of plasmons according to the spectral information. By extracting the PC1 of spectral clustering data, the nonlinear relationship between the PC1 and nanosphere diameters can be observed, as shown in Fig. 19G, which is consistent with the theoretical calculation results of Mie scattering.

In addition to analyzing the spectral properties of structures using unsupervised clustering, the dimensionality reduction algorithm can also be used to optimize the design of nanostructures by reducing the degree of freedom. Under this condition, the space sparsity and average distance between samples are decreased while clustering patterns between specific parameter collection and corresponding spectra responses can be extracted, which can improve the performance of optimization methods as gradient descent.

In most optimization missions, only a minority of samples are of targeted optical properties. Samples initialized by a fully randomized method cannot exhibit optical properties with important features or clustering effects. Thus, the training samples for unsupervised learning can be acquired by optimization methods directly in the original design space [38]. Under this condition, samples are exhibited as continuously moving points with optimization generation growth (Fig. 20A) because in most of the gradient-dependent optimization methods, new samples are mutated from better individuals in the previous generation, which makes the changes continuous. Under resetting optimizations with a random initial condition, multiple corresponding paths can be formed in the original design space. The color of the sample changes from blue to red, which means the enhancement of the optical responses. As noted in Fig. 20A, there is only one global maximum in the parameter space. Under this condition, all optimization paths are promised to reach the same location. However, in real situations, there can be quantities of local maxima and saddle points in the high-dimensional space, thus the optimization paths can lead to different locations.

Projections of original samples in the subspace (Fig. 20B) can be obtained after performing the dimensionality reduction method. By calculating the exhaustive mapping between suppressed parameters and targeted optical responses, it can be proved that the continuity and derivability are maintained in the subspace (Fig. 20C). Thus, the gradient-dependent optimization algorithms can also be applied in the subspace; by accumulating gradient information in fewer dimensions, the efficiency can be increased.

The above-mentioned conception is performed in a silicon-on-insulator platform to realize the maximized coupling efficiency by choosing specific parameters of the grating

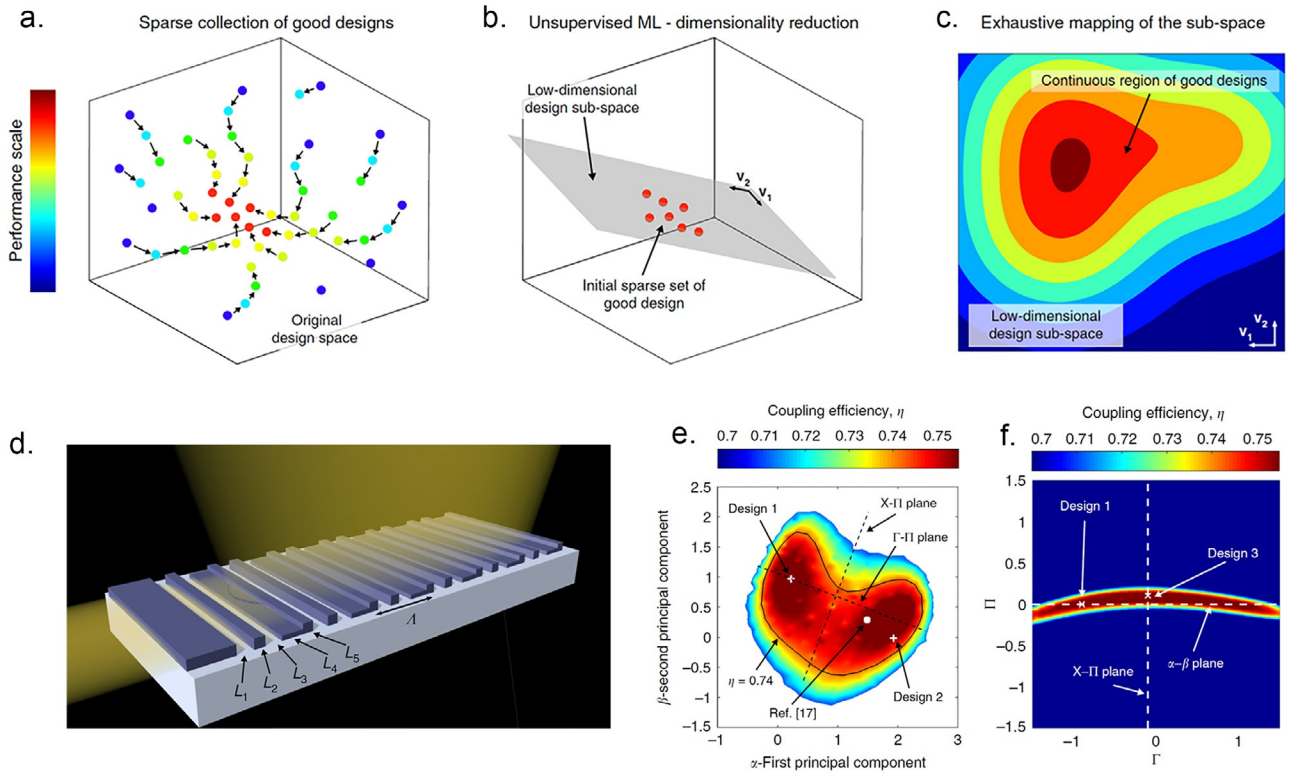


Fig. 20 (A) The initial design collection (acquired by several random restart optimizations) in original high-dimensional parameter space. (B) Demonstration of dimensionality reduction that represents designs in the lower-dimensional subspace (hyperplane defined by V1 and V2). (C) Exhaustively mapped subspace that exhibits a continuum of the designs and the corresponding spectra performance. (D) Schematic view of the grating coupler platform. (E) Mapping between coupling efficiency and extracted 2PCs acquired from original five-dimensional parameters. (F) Mapping between simulated coupling efficiency and latent parameters in Γ-Π and X-Π hyperplanes. *Dashed white lines* are the intersections between Γ-Π and X-Π hyperplanes and orthogonal 2PCs hyperplanes. *(From Springer Nature.)*

coupler structure [39]. More specifically, the grating coupler structure is a periodic one-dimensional waveguide (period $=\Lambda$) that consists of a 220 nm-height pillar and an L-shaped stripe with two top surfaces at 110 and 220 nm. Considering the length and location of the pillar and the etch-section of the L-shaped stripe, the number of design parameters is chosen as five (L1–L5 in Fig. 20D). TE-polarized light with a 1550 nm wavelength from a single-mode optical fiber vertically incident on the structure should realize the maximum coupling efficiency η while constraining the back-reflection r for less than -15 dB.

To train the machine learning model for dimensionality reduction, particle swarm optimizations are performed to produce samples with higher η. For each satisfied sample, 1000 generations are needed on average. Fed by sufficient samples, the trained PCA model can decrease the original five-dimensional design parameters into 2PCs while maintaining the representing accuracy. By plotting the mapping between 2PCs and corresponding coupling efficiency, as shown in Fig. 20E (for samples with $\eta > 0.7$ only), a variety of designs with $\eta > 0.74$ (region surrounded by a black solid line) can be discovered in the subspace; this is usually hard to notice in the high-dimensional original space. To prove that 2PCs are sufficient for design representation, additional two-dimensional hyperplanes of Γ-Π and X-Π orthogonal to the PC plane are chosen and shown as black dashed lines in Fig. 20E. When projecting samples with high coupling efficiency into those hyperplanes, a thin stripe distribution with only a slight curve is observed (Fig. 20F), which exhibits the two-dimensional characteristics of the subspace.

6. Perspectives and outlooks

CNN and its various derivative architectures have been implemented for wide spectra analysis applications, including reflection/transmission spectra analysis, time response ultrafast spectra characterization, optical reverse design, property clustering analysis, and mode analysis. Those methods show the potential of data-driven research in the field of nanophotonics. With the development of nanofabrication techniques, advanced machine learning algorithms, and calculation resources, CNNs can deepen the physical understanding of light-matter interactions and make better preparation for the development of practical photonic devices. The future development direction of the discipline includes the following points:

(1) To learn more optical responses beyond far-field spectra. Because the output of a CNN can also be a matrix, CNNs can also be applied to directly calculate the near-field distributions. Combining with optimization algorithms, strong energy concentration, and an electromagnetic hot spot can be supported. Selective activations of those phenomena can be further realized by tuning the polarization site, incident angle, and wavelength, which can be used in applications such as efficient solar cells, photocatalysis, hot electron injection, and surface-enhanced Raman spectroscopy.

(2) Better exploitation of parameterization space. The parameterization method so far represents structures with only a single layer into a 0/1 binary matrix, which is far from fully utilizing the parameterization ability of the model. By using continuous pixel value, introducing voxels, or material property defined pixels as discussed in Chapter 2, it can be easier to find a better optical response in larger parameter space. Meanwhile, fabrication techniques as well as NN architectures should be improved to match the whole designing-fabrication workflow.

(3) To establish a standard light-matter interaction database. There is still a lack of standard datasets for the interdisciplinary study of CNN-assisted spectra analysis, and these are crucial for the quantitative assessment of sampling efficiency, prediction accuracy, and generalization ability. Therefore, a standard and comprehensive database that maps from multiple design parameters to various light field properties should be developed, and the research and design process should be gradually standardized to lay the foundation for future industrial applications.

(4) To quantify the experimental feasibility in the model for real-world applications. At present, the majority of CNN models for spectra analysis or manipulation have only been proved by numerical simulations while many arbitrary patterns designed by the model may consist of complex components that cannot be fabricated. Thus, it is necessary to consider the requirements of fabrication technology comprehensively and integrate the influences of fabrication error and spectral error as quantitative indicators into the workflow.

References

[1] D.K. Gramotnev, S.I. Bozhevolnyi, Plasmonics beyond the diffraction limit, Nat. Photonics 4 (2) (2010) 83–91, https://doi.org/10.1038/nphoton.2009.282.

[2] S. Jahani, Z. Jacob, All-dielectric metamaterials, Nat. Nanotechnol. 11 (1) (2016) 23–36, https://doi.org/10.1038/nnano.2015.304.

[3] M. Pilo-Pais, A. Watson, S. Demers, T.H. LaBean, G. Finkelstein, Surface-enhanced Raman scattering plasmonic enhancement using DNA origami-based complex metallic nanostructures, Nano Lett. 14 (4) (2014) 2099–2104, https://doi.org/10.1021/nl5003069.

[4] N. Yu, F. Capasso, Flat optics with designer metasurfaces, Nat. Mater. 13 (2) (2014) 139–150, https://doi.org/10.1038/nmat3839.

[5] Q. Wang, Q. Xu, X. Zhang, C. Tian, Y. Xu, J. Gu, Z. Tian, C. Ouyang, X. Zhang, J. Han, W. Zhang, All-dielectric meta-holograms with holographic images transforming longitudinally, ACS Photonics 5 (2) (2018) 599–606, https://doi.org/10.1021/acsphotonics.7b01173.

[6] Y. Yang, P. Moitra, I.I. Kravchenko, D.P. Briggs, J. Valentine, All-dielectric metasurfaces, in: 2015 Conference on Lasers and Electro-Optics Pacific Rim, Optical Society of America, 2015, p. 25E2_1. http://www.osapublishing.org/abstract.cfm?URI=CLEOPR-2015-25E2_1.

[7] J.F. Mas, Mapping land use/cover in a tropical coastal area using satellite sensor data, GIS and artificial neural networks, Estuar. Coast. Shelf Sci. 59 (2) (2004) 219–230, https://doi.org/10.1016/j.ecss.2003.08.011.

[8] S.P. French, S. Muthukumar, Advanced technologies for earthquake risk inventories, J. Earthq. Eng. 10 (2) (2006) 207–236, https://doi.org/10.1080/13632460609350594.

[9] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, S.-C. Zhang, Learning atoms for materials discovery, Proc. Natl. Acad. Sci. U. S. A. 115 (28) (2018) E6411, https://doi.org/10.1073/pnas.1801181115.
[10] P. Wilding, M.A. Morgan, A.E. Grygotis, M.A. Shoffner, E.F. Rosato, Application of backpropagation neural networks to diagnosis of breast and ovarian cancer, Cancer Lett. 77 (2) (1994) 145–153, https://doi.org/10.1016/0304-3835(94)90097-3.
[11] L. Deng, D. Yu, Deep learning: methods and applications, Found. Trends Signal Process. 7 (3–4) (2014) 197–387, https://doi.org/10.1561/2000000039.
[12] M.D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, Springer International Publishing, 2014, pp. 818–833.
[13] H. Navidan, P.F. Moshiri, M. Nabati, R. Shahbazian, S.A. Ghorashi, V. Shah-Mansouri, D. Windridge, Generative adversarial networks (GANs) in networking: a comprehensive survey & evaluation, Comput. Netw. 194 (2021), https://doi.org/10.1016/j.comnet.2021.108149, 108149.
[14] H. Habibi Aghdam, E. Jahani Heravi, H. Habibi Aghdam, E. Jahani Heravi, Guide to Convolutional Neural Networks, Springer International Publishing, 2017, pp. 85–130, https://doi.org/10.1007/978-3-319-57550-6_3.
[15] T. Asano, S. Noda, Optimization of photonic crystal nanocavities based on deep learning, Opt. Express 26 (25) (2018) 32704–32717, https://doi.org/10.1364/OE.26.032704.
[16] L. Pilozzi, F.A. Farrelly, G. Marcucci, C. Conti, Machine learning inverse problem for topological photonics, Commun. Phys. 1 (1) (2018) 57, https://doi.org/10.1038/s42005-018-0058-8.
[17] W. Ma, F. Cheng, Y. Liu, Deep-learning-enabled on-demand design of chiral metamaterials, ACS Nano 12 (6) (2018) 6326–6334, https://doi.org/10.1021/acsnano.8b03569.
[18] I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, H. Suchowski, Plasmonic nanostructure design and characterization via deep learning, Light Sci. Appl. 7 (1) (2018) 60, https://doi.org/10.1038/s41377-018-0060-7.
[19] I. Sajedian, J. Kim, J. Rho, Finding the optical properties of plasmonic structures by image processing using a combination of convolutional neural networks and recurrent neural networks, Microsyst. Nanoeng. 5 (1) (2019) 27, https://doi.org/10.1038/s41378-019-0069-y.
[20] Z. Liu, D. Zhu, S.P. Rodrigues, K.-T. Lee, W. Cai, Generative model for the inverse design of metasurfaces, Nano Lett. 18 (10) (2018) 6570–6576, https://doi.org/10.1021/acs.nanolett.8b03171.
[21] Y. Li, Y. Xu, M. Jiang, B. Li, T. Han, C. Chi, F. Lin, B. Shen, X. Zhu, L. Lai, Z. Fang, Self-learning perfect optical chirality via a deep neural network, Phys. Rev. Lett. 123 (21) (2019), https://doi.org/10.1103/PhysRevLett.123.213902, 213902.
[22] T. Zahavy, A. Dikopoltsev, D. Moss, G.I. Haham, O. Cohen, S. Mannor, M. Segev, Deep learning reconstruction of ultrashort pulses, Optica 5 (5) (2018) 666–673, https://doi.org/10.1364/OPTICA.5.000666.
[23] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1520–1528, https://doi.org/10.1109/ICCV.2015.178.
[24] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv.Org (2013), https://doi.org/10.48550/arXiv.1312.6114.
[25] W. Ma, F. Cheng, Y. Xu, Q. Wen, Y. Liu, Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy, Adv. Mater. 31 (35) (2019) 1901111, https://doi.org/10.1002/adma.201901111.
[26] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: International Conference on Learning Representations, arXiv:1511.06434, 2016, https://doi.org/10.48550/arXiv.1511.06434.
[27] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, arXiv.Org (2018), https://doi.org/10.48550/arXiv.1710.10196.
[28] S. An, B. Zheng, H. Tang, M.Y. Shalaginov, H. Zhang, Multifunctional metasurface design with a generative adversarial network, Adv. Opt. Mater. 9 (2021) 2001433, https://doi.org/10.1002/adom.202001433.
[29] J. Jiang, D. Sell, S. Hoyer, J. Hickey, J.A. Fan, Free-form diffractive metagrating design based on generative adversarial networks, ACS Nano 13 (8) (2019) 8872–8878, https://doi.org/10.1021/acsnano.9b02371.

[30] F. Wen, J. Jiang, J.A. Fan, Robust freeform metasurface design based on progressively growing generative networks, ACS Photonics 7 (8) (2020) 2098–2104, https://doi.org/10.1021/acsphotonics.0c00539.
[31] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504, https://doi.org/10.1126/science.1127647.
[32] H. Liu, H. Motoda, H. Liu, H. Motoda, Feature Transformation and Dimensionality Reduction, Springer US, 1998, pp. 151–187, https://doi.org/10.1007/978-1-4615-5689-3_6.
[33] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemom. Intel. Lab. Syst. 2 (1) (1987) 37–52, https://doi.org/10.1016/0169-7439(87)80084-9.
[34] Z.-P. Liu, Linear Discriminant Analysis, Springer, New York, 2013, pp. 1132–1133, https://doi.org/10.1007/978-1-4419-9863-7_395.
[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2010) 3371–3408.
[36] W. Ma, Y. Liu, A data-efficient self-supervised deep learning model for design and characterization of nanophotonic structures, Sci. China Phys. Mech. Astron. 63 (8) (2020), https://doi.org/10.1007/s11433-020-1575-2, 284212.
[37] B. Karlik, M.F. Yilmaz, M. Ozdemir, C.T. Yavuz, Y. Danisman, A hybrid machine learning model to study UV-vis spectra of gold nanospheres, Plasmonics 16 (1) (2021) 147–155, https://doi.org/10.1007/s11468-020-01267-8.
[38] Y. Grinberg, D. Melati, M.K. Dezfouli, S. Janz, J.H. Schmid, P. Cheben, S.-P. Alejandro, J. Gonzalo Wangüemert-Pérez, I. Molina-Fernández, A. Ortega-Moñux, X. Dan-Xia, Reaping the benefits of machine learning pattern recognition in nanophotonic component design, in: Proc. SPIE, vol. 10921, 2019, https://doi.org/10.1117/12.2506787.
[39] D. Melati, Y. Grinberg, M. Kamandar Dezfouli, S. Janz, P. Cheben, J.H. Schmid, A. Sánchez-Postigo, D.-X. Xu, Mapping the global design space of nanophotonic components using machine learning pattern recognition, Nat. Commun. 10 (1) (2019) 4775, https://doi.org/10.1038/s41467-019-12698-1.

CHAPTER 7

Nanoscale electronic synapses for neuromorphic computing

Zhongwei Xu and Fushan Li
Institute of Optoelectronic Technology, Fuzhou University, Fuzhou, China

1. Introduction

It is a long-term pursuit of scientists to realize an electronic device that is efficient in space and energy that can flexibly run various complex calculations and is easy to expand and integrate [1,2]. As proposed by Intel founder Gordon Moore in 1965 in his famous Moore's law, at the same cost, the computing power and transistor density of new processors will double about every 24 months [3]. John von Neumann designed an architecture for an electronic computer that has achieved great success in the era of Moore's Law. At present, almost all our computers and electronic devices for entertainment, leisure, and work are all realized with this architecture. However, this law is nearing its end as the semiconductor industry has gradually entered the "king is dying" phase [4]. The reason is the extremely high cost (energy consumption, speed loss) of data movement between memory and processor. With the increasing density of transistors, the bottleneck of data transmission and distribution will become insurmountable, ultimately limiting the computing power. This is usually called the von Neumann bottleneck [5], and is shown in Fig. 1A.

The synapse comes from the human brain, which is the most advanced computing system to date after thousands of years of evolution [6]. With the improvement of electronic technology and the self-knowledge of our brains, it has been shown that the reason for learning and memory is the connection between each nerve [7]. In the human brain, there are about 10^{11} neurons, which are the basis of thinking, memory, and emotional changes. These neurons are tightly connected to each other with a synapse; each neuron consists of an axon and multiple dendrites. Dendrites collect and transmit signals to neurons, and axons conduct neuronal signals to the next neuron. The strength of this connection (synaptic weights) changes with neural activity, as shown in Fig. 1B. Researchers use a variety of models to interpret neurobehavior and apply the research results to the machine learning task [8–12]. From the single-layer neural network to the multilayer neural network [13], many algorithms such as artificial intelligence and deep learning have shown great potential in many scenarios. These technologies are widely used today for face recognition, optimal decision-making, data mining, etc. Although these

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00007-X
Copyright © 2023 Elsevier Inc.
All rights reserved.

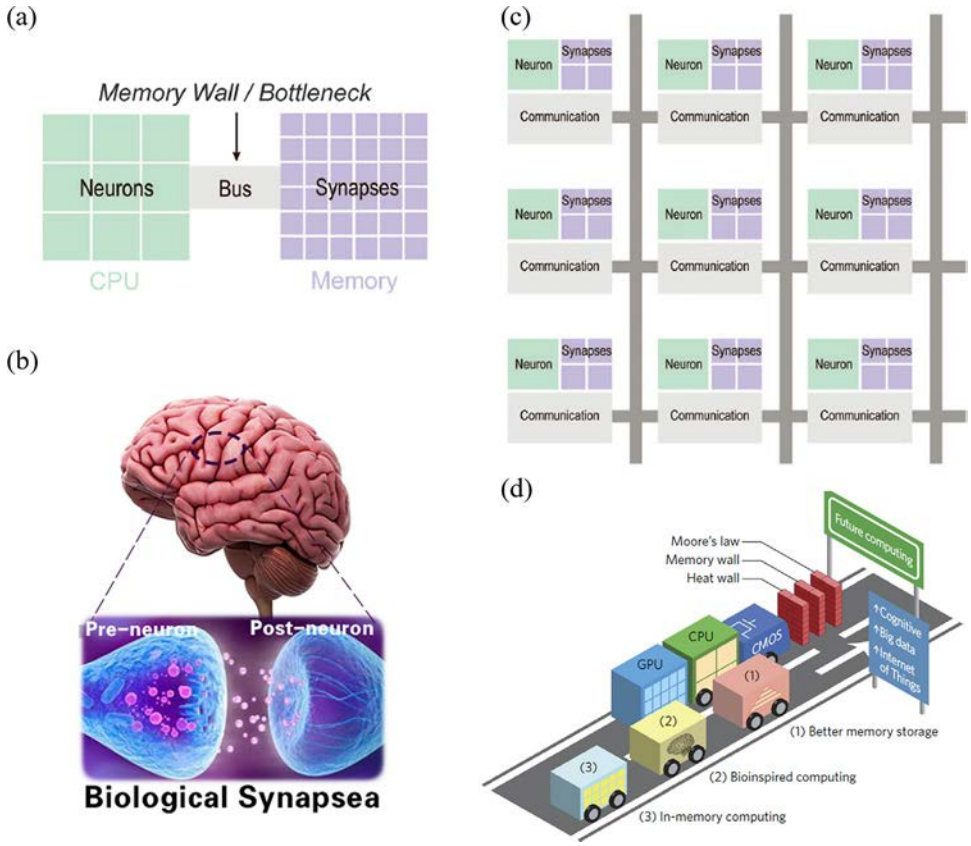


Fig. 1 (A) The current computer architecture acts as a neuron (CPU) and a synapse (memory). Under the increasing performance of the CPU, the data exchange between memory and CPU increases greatly, resulting in a decrease in efficiency and high energy consumption. When simulating a highly interconnected network that cannot run on a single processor, it will result in CPU loss. (B) Human brain and a synapse. A large number of synapses achieve complex functions and are the basis of mental activity. (C) Conceptual blueprint of an architecture that, like the brain, tightly integrates memory, computation, and communication in distributed modules that operate in parallel and communicate via an event-driven network. (D) The race toward future computing solutions. Developments in electronic synapse technology may provide an alternative path that enables hybrid memory logic integration, bio-inspired computing, and efficient reconfigurable in-memory computing systems.

technologies are similar to the way the brain works, they are actually built on software, and they are all applied on the basis of the von Neumann machine. Eventually it is difficult to avoid the barrier of the hardware architecture (Fig. 1A).

An electronic synapse is considered a solution to the above problems. Using custom calculation elements to simulate neurobiology shows excellent results in terms of

calculation amount and energy consumption, with examples including the IBM TrueNorth chip [14], the Intel Loihi chip [15] and SpiNNaker [16]. More and more devices have been developed while various materials and mechanisms have been adopted to realize neuromorphic devices. Electronic synapses usually have a two-terminal or three-terminal architecture, and their internal electronic properties can change with the outside world. When external stimuli are applied (electrical, optical, or even mechanical stimuli, etc.), the material properties are reconfigured (resistance, conductance, transmittance, etc.) and memorized. This function can be used to store or process data directly. At present, artificial synapses represented by memristors can be manufactured in the nanoscale [17], and can maintain extremely long storage states [18], subnanosecond switching speed [19], extremely high durability [20], and very low energy consumption [21]. A large portion of the neuromorphic devices have evolved from traditional storage devices [22] such as static random access memory (SRAM), magnetic RAM (MRAM), ferroelectric RAM (FeRAM), and resistive RAM (ReRAM).

The combination of memristors and neuromorphic engineering shows unprecedented potential in practical applications [23]. The integrated artificial neural network has a low thermal budget on the processor through a very high-density local interconnect. It eliminates off-chip communication between the memory and the processor, as shown in Fig. 1C [4]. Neuromorphic computing can implement large-scale spiking neural networks that are efficient, scalable, and flexible within today's technologies. There is also a great chance to solve problems (as shown in Fig. 1D) such as the increase in calculation demand, the approach of physical limits, etc. [24]. The implementation of neuromorphic engineering on the hardware level can be realized by lots of artificial electronic synapses [25–27]. In this work, we will review the progress of electronic synapse research and give a detailed introduction to its application in integrated neuromorphic engineering.

2. Realization of artificial synapses

2.1 Ion migration

Ion migration is considered to be one of the main mechanisms for realizing artificial synapses, and it has been intensively studied recently. The types of migration generally include cations and anions, which come from electrodes and interlayer materials, respectively. In addition, electrolytes can also induce the device performance change.

2.1.1 Cation migration

The internal characteristics of the device (such as resistance and conductance) could change due to the movement of cations under the stimulation of the outside world. Devices based on cation migration usually use a sandwich structure such as a metal/semiconductor material/metal structure, where the anode uses active metals

generally such as silver and copper while the semiconductor uses a cation-rich material (the electrode can be an inactive metal at this time). It must be ensured that cations can undergo redox reactions in the medium [28]. The device is exposed to external stimuli, such as bias voltage, and the overall conductivity of the device shows a significant change. The change is related to external stimulus intensity, frequency, and other factors.

A typical example is the research conducted by Jo and his team in 2010 [29], as shown in Fig. 2. The study constructed a composite film of silicon and silver with a properly designed Ag/Si mixture ratio gradient. Under the effect of external voltage bias, silver ions diffuse in the direction of the electric field, the silver-rich region expands, and the overall conductivity of the device rises, gradually changing from the high resistance state (HRS) to the low resistance state (LRS). Reverse voltage bias can also be applied to recover from any process state to HRS. The change in conductivity also depends on the frequency and width of the pulse, realizing some functions of the biological nerves such as long-term potentiation and depression (LTP), short-term potentiation and depression (STP), and spiking-time dependent plasticity (STDP). There are some reports of similar mechanisms such as Ag/polyurethane [30], Ag/TiO_2 [31], and Cu/SiO_2 [32]. However, the uniform migration of metal ions in the medium has not been strongly proven (the diffusion of silver-rich and silver-poor regions).

This type of diffusion is often referred to as "electrochemical metallization (ECM) cells" [33]. In this case, the voltage applied to the anode promotes the oxidation of the metal, providing the migrated metal cation (in order to maintain electrical neutrality, other places of the device may also participate in the reaction [33]). The movement of metal ions in the medium causes the resistance change. However, the specificity of the internal environment (for example, electrodes are irregular in microscopic appearance, electric field nonuniformity, etc.) results in the formation of atomic-level metal ion channels connecting the cathode and anode (conductive filament, CF) [34]. Because the distance between the electrodes is very small, it can bring a large electric field, which ensures that the metal ions are oxidized and integrated into the semiconductor layer to finally form a conductive filament. The growth and rupture of metal channels lead to nonuniform resistance changes inside the device, showing the overall state of change of the device [35,36]. For example, silver ions can stably replace the position of titanium under the electric field to form a CF. [37] As shown in Fig. 2D, we can clearly see that the CF composed of the Ag element connects the two electrodes [31]. However, this is a redox reaction. As shown in Fig. 2E [33], the metal of the anode is continuously oxidized into cations and released into the medium; the corresponding cathode surface undergoes a reduction reaction to turn the free cations into metals. This process remains electrically neutral. There are also many studies showing the growth of metal filaments at the cathode [31,38]. For example, from the research of Guo (Fig. 2F), it can be seen that the silver produced from the cathode is rapidly expanding [39].

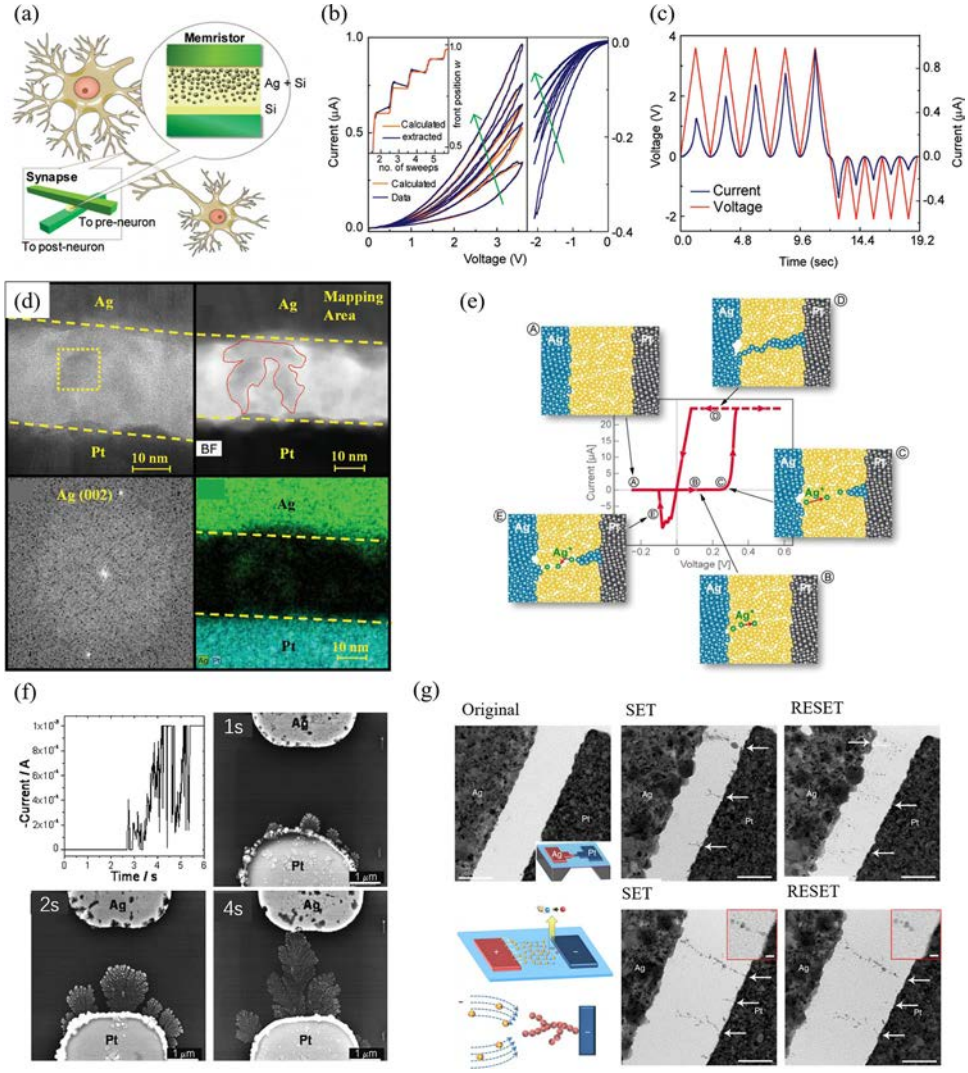


Fig. 2 (A) Schematic of the concept of using memristors as synapses between neurons. The insets show the schematics of the two-terminal device geometry and the layered structure. (B) Measured (*blue lines* (dark gray in print version)) and calculated (*orange lines* (gray in print version)) I–V characteristics. (C) The current and voltage data vs time for the device. (D) HRTEM and EDS results show that there are clear conductive filaments at protrusion-like regions formed connecting the top electrode and the bottom electrode. (E) Sketch of the steps of the set and reset operations of an electrochemical metallization memory cell. (F) SEM images showing the Ag dendrite growth after applying 1 V for about 1, 2, and 4 s, and the corresponding I–V curve, respectively. (G) Observation of conducting filament dynamics in SiO_2-based resistive memories. It shows the device set and reset of two widths and the working schematic of the device. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In fact, the CF formed by the migration of metal ions may display different shapes due to different materials, different interfaces, different electric fields, and different frequencies [40]. Therefore, electronic synapses can have time- and drive-dependence plasticity, such as STDP. Similarly, there are also reports that some special structures and materials could be used to control the CF in order to realize the stability of the device, such as a nanogrid [41], graphene barrier [42,43], etc.

Similarly, after forming the conductive filament, as long as the reverse voltage is applied, the conductive filament is broken and the conductivity returns to the original state [44]. For example, under the action of reverse voltage, the metal grown on the Pt cathode breaks and dissolves into balls, then the original device state is restored (Fig. 2G) [40]. Actually, the CF will naturally break even if it is not stimulated at all, or form a small agglomeration to reduce the interface energy [45]. Therefore, short-term plasticity (STP) and long-term plasticity (LTP) of the neuromorphic calculation become feasible.

2.1.2 Anion migration

In this case, anion migration serves as the dominant operating mechanism for the electronic synapses: under the stimulation of the bias voltage, the conducting characteristics of the active layer gradually change. In general, a CF is formed between the two electrodes of the device. This channel is generated by the increase in vacancy concentration, such as oxygen vacancy after the application of an external electric field [46]. The specific process is shown in Fig. 3A [47]:

1) Under the stimulation of a high electric field, dielectric soft-breakdown occurs and oxygen ions drift to the anode interface. If the anode material is easy to oxidize, it is possible to form an interface oxide layer (oxygen ions are discharged as neutral non-lattice oxygen) to store oxygen ions [46]. The CF is formed by oxygen vacancy, the resistance drops significantly, and the device changes to LRS.
2) When the recovery bias is applied, the oxygen ion returns to its original region, and the device returns to the HRS, but does not return to the original state. This is because the as-generated CFs cannot be completely oxidized. The remaining CF fragments are often called "virtual electrodes" and can promote the formation of conductive filaments in the future [47].

These processes, including the formation of conductive filaments (Fig. 3B), rupture (Fig. 3C), and fragments (Fig. 3D) have been fully proven by Chen et al. [48] Similar results have been reported as well [49–51]. It is worth noting that there are two different modes in this process: unipolar switching modes represented by Pt/NiO/Pt [52], Pt/ZnO/Pt [53], and Pt/Al2O3/Pt [54] as well as bipolar switching modes represented by Pt/$NiO/SrRuO_3$ [55] and TiN/ZnO/Pt [56]. The unipolar switch mode can realize the electrical state change through the voltage of any polarity. If such a unipolar device can work symmetrically, we can call it a nonpolar device. Bipolar switching mode is

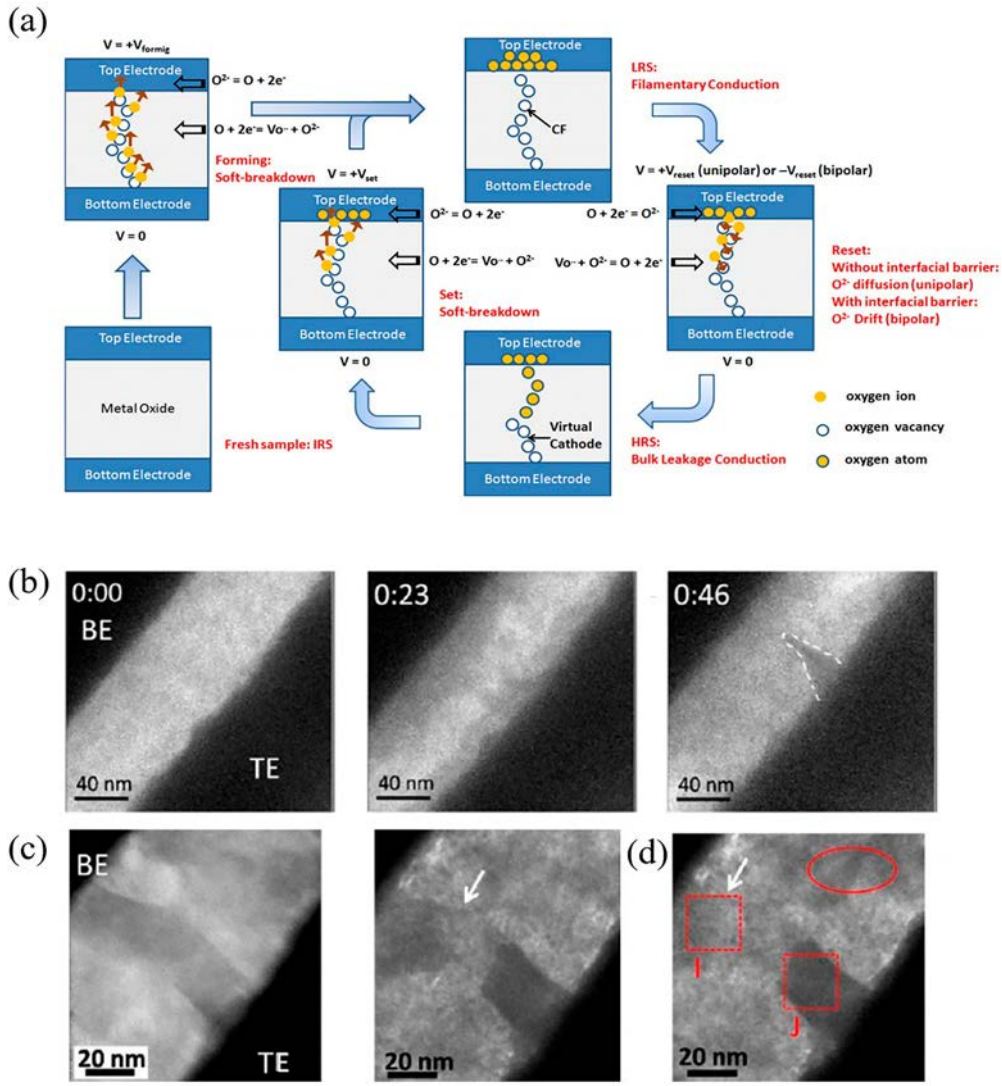


Fig. 3 (A) Schematic of the switching process in the simple binary metal-oxide RRAM. (B) The formation process of conductive filaments was shown by TEM. Ruptured (C) and fragments (D).

when the device changes depending on the polarity of the applied voltage. In bipolar switching mode, the oxidized interface may have a stronger barrier to limit the diffusion of oxygen ions, and the reset of oxygen ions requires the help of a reverse electric field [57].

However, the conduction mechanism for the device is rather complicated. According to Lim and Yu [58], the conduction mechanism in this sandwiched structure can be

divided into nine types: (1) Schottky emission; (2) Flower-Nordheim tunneling; (3) direct tunneling; (4) Poole-Frenkel emission; (5) space-charge-limited conduction; (6) ionic conduction; (7) Ohmic conduction; (8) hopping conduction; and (9) trap-assisted tunneling. In actual devices, multiple mechanisms might take effect simultaneously, and further investigation on the electrical characteristics needs to be carried out to distinguish the dominant mechanism (Table 1).

Although research on such conductive filaments is extensive, the randomness of CF growth cannot be ruled out. Therefore, research on device uniformity has also been a hot topic in recent years, especially for devices with nonconductive filaments [59,60]. The realization of such a method is usually based on a redox reaction, which only occurs at the interface of the layers. The redox reaction causes the change of interface thickness or position as well as the electrical properties of the device. For example, a PCMO/TiN device forms an intermediate layer of TiO_xN_y gradually after 20 stimuli (Fig. 4C) as an electronic barrier to limit the current [60]. When a reverse bias is applied, the thickness of the intermediate layer will decrease, thereby achieving the conversion of HRS to LRS. This method eliminates the uncertainty of conductive filaments and ensures good device-to-device and cycle-to-cycle uniformity.

2.1.3 Electrolyte-gated transistor

Three-terminal transistors with an inducing electrolyte have also been adopted to achieve the synapse effect. As shown in Fig. 5A [61], the transistor has a high similarity to biological tissue. There are roughly two types of ion transport in transistor devices, ion induction and ion implantation. The gate voltage can drive the ions in the electrolyte to the channel interface (ion induction) or even to the inside of the channel material (ion implantation), resulting in a change in channel conductivity. Usually, transistors use the electric double layer (EDL) [62] method to achieve electronic synapses. EDL devices are controlled by thin film capacitors. The applied gate voltage can cause the movement of ions, which gather on the surface of the channel. In order to maintain electrical neutrality, induced charges are also generated inside the channel, forming an ultrathin capacitor [62,63]. Channel characteristics can be controlled by the intensity and frequency of the gate pulse. The main feature of the device is that the driving voltage is very low, which is due to the strong electric field between the ultrathin capacitors (between the ion layer and the channel) [64]. The leakage current could only appear between the resistive material (postsynapse) and the control terminal (presynapse), so this device will be very energy-efficient. It has been reported that the device works at 0.3 V and has very low energy consumption [65]. It could be combined with traditional transistor technology so it is easier to integrate. Because the ions are gathered by the external electric field, they will flow back when the external electric field is removed, and the device returns to the original state. The aggregation and dissipation of ions would affect the conductivity of the device directly. Various materials such as indium oxides [66–69],

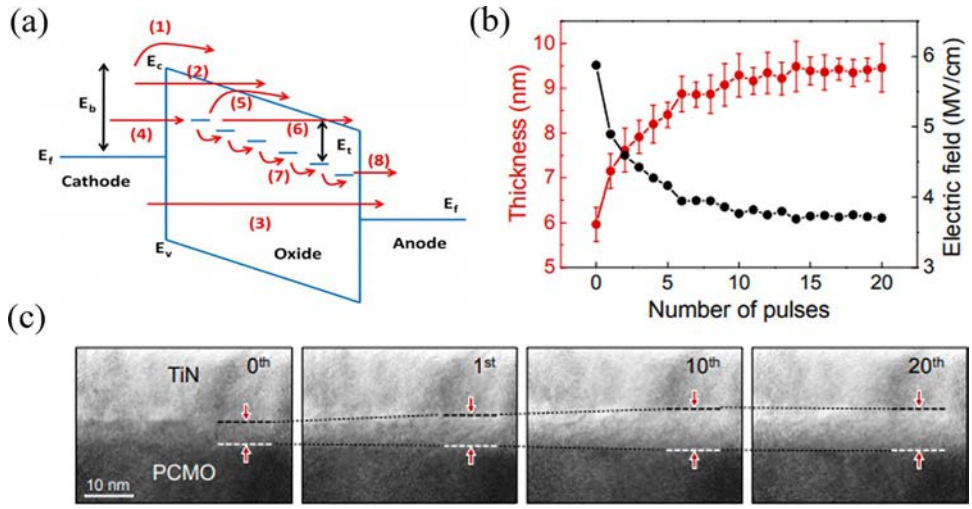


Fig. 4 (A) Schematic of the possible electron conduction paths through an MIM stack. (1) Schottky emission: thermally activated electrons injected over the barrier into the conduction band. (2) Fowler Nordheim (F–N) tunneling: electrons tunnel from the cathode into the conduction band; this usually occurs at a high field. (3) Direct tunneling: the electron tunnels from the cathode to the anode directly; this usually occurs when the oxide is thin enough. If the oxide has a substantial number of traps (e.g., oxygen vacancies), trap-assisted tunneling contributes to additional conduction, including the following steps: (4) tunneling from cathode to traps; (5) emission from trap to conduction band, which is essentially the Poole Frenkel emission; (6) F N-like tunneling from the trap to the conduction band; (7) trap to trap hopping or tunneling, maybe in the form of Mott hopping when the electrons are in the localized states or maybe in the form of metallic conduction when the electrons are in the extended states depending on the overlap of the electron wave function; and (8) tunneling from traps to the anode. (B) In situ TEM resistance switching to HRS at a voltage pulse mode. The thickness of the a-TiO_xN_y layer is marked by red arrows. (C) Change in thickness of the a-TiO_xN_y layer (*red* (gray in the print version)) and the electric field across the a-TiO_xN_y layer (*black*) with the voltage pulse. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

phosphorosilicate glass [69–71], organic materials [72], etc., have been used in the devices, and there have been reports of the use of two-dimensional materials and ionic gels [73,74].

On the other hand, under the action of the electrolyte in the electric field, ions enter the channel material and chemically react, which is generally called electrochemical doping [75]. It can also drive ions into the target material to change the energy band structure of the channel material, thereby changing the conductivity of the channel material, which is called ion intercalation. However, ion penetration and doping cannot be spontaneously restored. Devices based on electrochemical doping generally use a combination of oxides and ionic liquids [61,76]. It is worth noting that the above devices can only

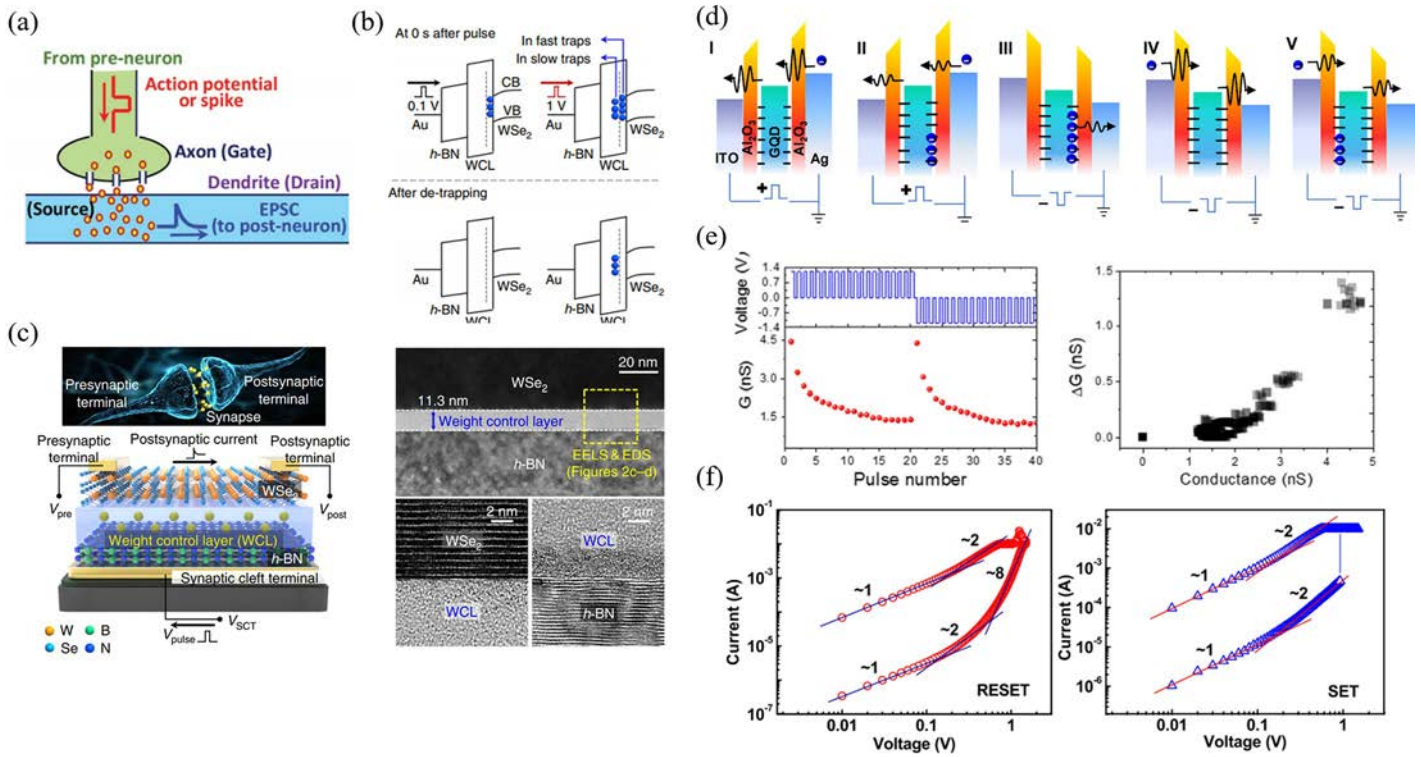


Fig. 5 (A) Schematic illustration of a biological synapse. The plasticity between a pair of pre- and postneurons is modulated by the action potential or spikes from the preneuron. (B) Illustration of energy band diagrams after pulse and detrapping shows the band bending after the stimulus, changing the electrical conductivity of the device. (C) Structure of the h-BN/WSe2 synaptic device and the X-TEM image of the WSe2/WCL/h-BN structure, and the high-resolution images corresponding to the WSe2/WCL and WCL/h-BN interfaces. (D) Schematic of the working mechanism of the e-synapses, which have the mechanism of trapping-detrapping. (E) Switching behaviors under a sequence of positive pulses. The positive pulse can decrease the conductance *G* of the e-synapse, resulting in synaptic depression (*left*). Conductance variation *ΔG* as a function of the initial conductance (*right*). (F) I–V curves of the electroforming, reset, and set processes as well as the fitting results.

work below a specific frequency. A capacitor can be formed between the two electrodes. When the frequency of the electric field is relatively low, the influence of capacitance will be reduced, and the role of the electrolyte will become dominant [61].

2.2 Electronic migration

Generally, such a device mainly includes electron migration induced by the external field and photogenerated carrier migration. Electronic migration is known for its high operating stability and integration. We will be discussing pure electronic devices and some photonic devices below.

2.2.1 Electron migration by external field

Electron migration is compatible with current semiconductor technology, and electron migration devices are more stable and faster than the above ion-dependent devices. Electron migration devices require electron-trapping materials, and can be realized in the form of transistors or two-terminal devices. In a transistor, when an electric field is applied to the gate, electrons will be trapped in the dielectric layer. As shown in Fig. 5B, the dielectric layer is negatively charged and is accompanied by electrical control of the channel [77]; electron accumulation causes the band to bend. But this control is changeable in different material systems, and it will quickly be detrapped in some material systems. When the gate is given a reverse electric field, the trapped charge is released and the entire device returns to its original state. Using this process can realize the various functions of electronic synapse. New materials have also been used in these devices, such as the application of two-dimensional WSe_2 [78], carbon nanotubes, HfO_2 [75], etc. There are also some examples of carefully designed structures to achieve multilayer charge trapping [79]. As shown in Fig. 5C, the device can capture multiple states of charge while achieving STP and LTP.

In case of two-terminal structures, the working mechanism is also based on charge trapping and detrapping by using special materials such as graphene, quantum dots, and gold nanoparticles [80,81]. As shown in Fig. 5D, it is proposed that the nanomaterial doped inside the medium can automatically trap electrons. In the initial state, a large number of traps in the material seriously affect carrier transport, resulting in a high resistance state of the device. When these traps are filled, the device can transition to a low resistance state under the applied bias voltage. However, the entrapment of too many electrons in a material may cause a decrease in conductivity (repulsion), as shown in Fig. 5E. Research suggests that the change in conductivity is due to the trap-controlled space charge limited conduction (SCLC) mechanism and the Ohmic conduction mechanism [82]. The material using this mechanism has a multilayer trap structure, and the trap–detrap process is voltage-dependent. The voltage gradually increases, and the trap filling range gradually expands. When the voltage or the number of pulses reaches the threshold and the trap is filled, the current will suddenly change, [82] as shown in

Fig. 5F. Other materials such as PO_x [83], MoS_2 [84], CuPc [85], etc., have been used for electron migration devices. In addition, quantum dot materials [86] such as graphene quantum dots, graphene oxide quantum dots [87], and hybrid quantum dot systems have been developed to achieve complex neurobehavior [88].

2.2.2 Photogenerated carrier migration

Neuromorphic photonic devices are developed from traditional devices, and their applications have been fully demonstrated [89,90]. This kind of device can be roughly divided into photogenerated carrier devices, photosensitive material devices (such as phase change materials) and optical resonant cavities. As shown in Fig. 6A [91], a photogenerated carrier migration device generates time-dependent carriers by light to change the conductivity of the device. Photogenerated carriers must be stored inside the device before they can change the electrical characteristics of the device. Impurities can provide traps, and the presence of oxygen vacancies, interface defects, and impurity doping can provide traps for carrier storage. When light with an appropriate wavelength is irradiated on the device, the active layer absorbs photons and generates electron and hole pairs. Due to the concentration gradient or the presence of an electric field, the carriers diffuse and fill the traps in the device gradually. This process gradually changes the overall conductivity of the device. When reverse stimulation is applied, carriers are released or detrapped after a period of time, and the device can return to its original state gradually.

Oxide materials with high light absorption efficiency, such as ZnO [92], SiO_2 [93], Al_2O_3 [94], etc., have been investigated for application in optoelectronic synapses. Various heterojunctions such as In_2O_3/ZnO [95], ion liquid-PVP dielectric/ZnO interfaces [96], indium zinc oxide (IZO) [97], graphene/diamond interfaces [98], etc., were also adopted to realize optoelectronic synapses. Under light stimulation, the generated electrons and holes will accumulate on both sides of the heterojunction and get trapped, resulting in conductivity changes. There are also examples of realizing the memory effect through photons [99]: holes generated in the MgO layer were injected into the ZnO thin film and gave electroluminescence. Moreover, ultraviolet (UV) irradiation can promote CF growth [100].

Due to the stimulation of photons, phase change in a certain material occurs gradually, and its conductivity changes accordingly. This change can also be applied to electronic synapses. The phase change caused by light is mainly based on the crystallization rate and light absorption of the material. A typical phase change material is $Ge_2Sb_2Te_5$ (GST). Previous GST-based studies have demonstrated its potential to be applied in neural networks and have demonstrated time-dependent plasticity [101,102]. As shown in Fig. 6C, the light passes through the waveguide and generates heat, causing the GST to undergo a phase transition. However, this phase change-based waveguide neural device will face some challenges in integration. For example, the device size is too large to integrate. With the development of photon computing, there are also studies on the use of optical

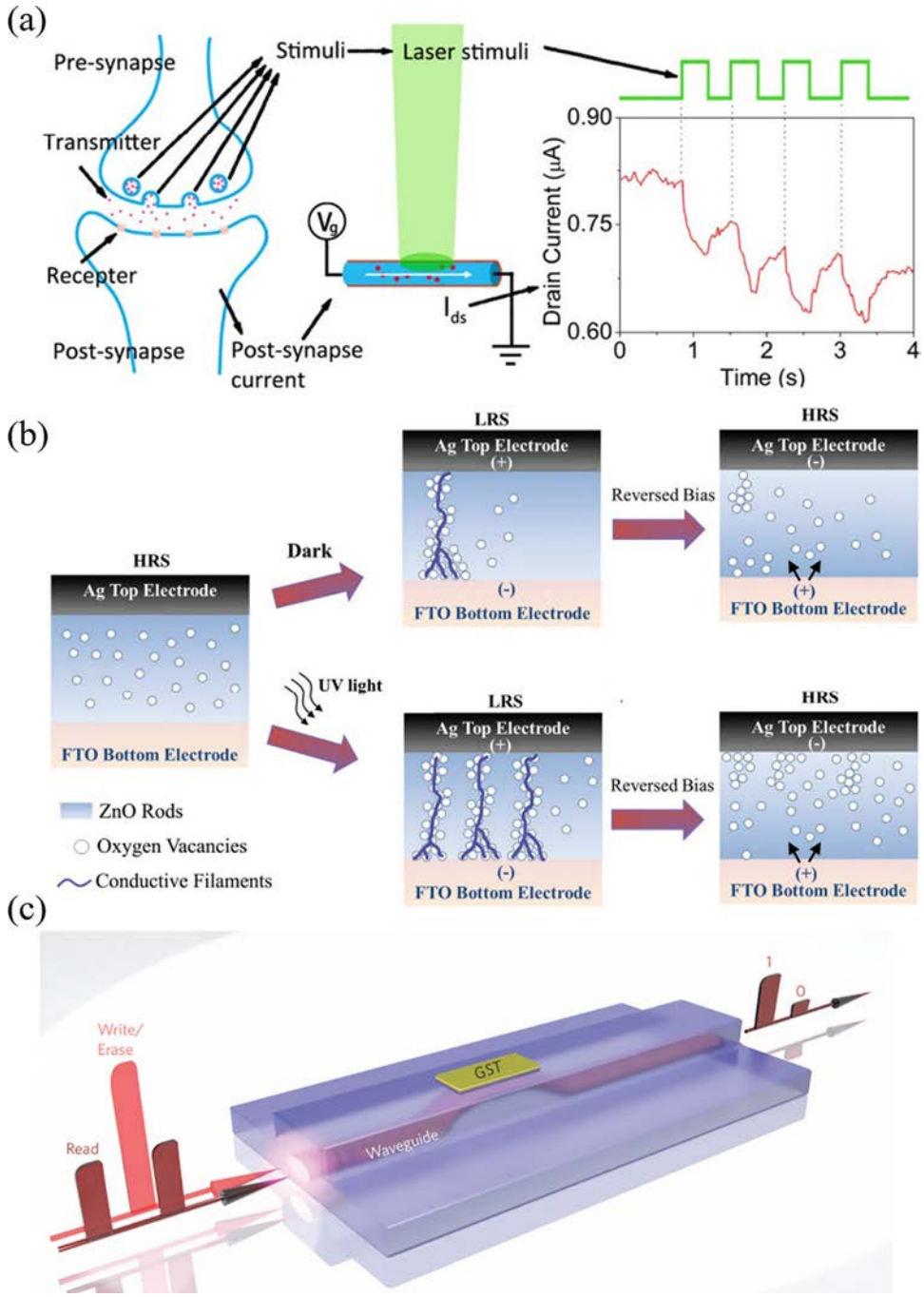


Fig. 6 (A) Schematic illustration of the mimicking of a biological synapse. The drain current (the postsynaptic current) of the phototransistor is recorded in response to the laser pulse train (the presynaptic spikes). The interval of the laser pulse train is random for mimicking the generation of transmitters. (B) The resistive switching mechanism of the Ag/ZnOrods/FTO device. More oxygen vacancies are formed, which leads to the formation of multiple conductive filaments. (C) Operation principle of the all-optical on-chip memory device. Information is stored in the phase state of the GST section on top of the nanophotonic waveguide. Both reading and writing of the memory can be performed with ultrashort optical pulses because the guided light interacts with the GST via its evanescent field. In the readout, data are encoded in the amount of optical transmission through (along) the waveguide because the two crystallographic states of the GST exhibit a high contrast in optical absorption.

amplifiers to achieve artificial synapses [101,102]. Because this is a photonic device [103], it has extremely high operation speed and efficiency [104,105]. However, it still needs further investigation for integrated applications.

2.3 Phase transition

Devices based on phase change have achieved great success in data storage, such as digital optical disks and traditional nonvolatile memory [106]. As shown in Fig. 7A [107], the conductivity of the phase change device is changed through crystallization and amorphization, showing a lower resistance (LRS) in the crystalline state while showing a higher resistance (HRS) in the amorphous state. The crystallization process is usually divided into two categories: growth-dominated crystallization represented by $Ge_2Sb_2Te_5$ (GST) that has been discussed above, and nucleation-dominated crystallization represented by AgInSbTe (AIST). The nucleation-dominated process incubates in the amorphous area and continues to grow until the amorphous area is completely converted to polycrystalline. Growth-dominated crystallization is carried out at the interface of the crystal and amorphous area, and finally the amorphous area transforms into polycrystalline and further into single crystal [108]. However, such a change requires external heat; crystallization occurs when the temperature reaches 500–900 K. When the temperature continues to rise, the crystals will melt. Therefore, the device changes from HRS to LRS when wide and low-frequency voltage pulses are applied to the device; short and high-frequency pulses can change the device from LRS to HRS (the molten amorphous area will quickly cool and affect the device conductivity) [109].

Current research focuses on the switching speed and the energy consumption of the devices based on phase change materials. Traditional materials such as GST [110] and AIST [111] as well as new materials such as Ti–Te bonds [112] and Sc–Te bonds [113] have achieved breakthroughs in crystallization speed. It has been reported that ScSbTe (SST) [113], GeSbTe, [114] and TiSbTe [115] can form a fast switching state of crystals in less than 500 fs. On the other hand, the SiSbTe alloy [116], GeSb alloy [117], GeTe alloy [118], etc., can be used to reduce energy consumption. In 2010, Duygu Kuzum et al. demonstrated the typical neuroengineering application of the phase change materials [119]. Fig. 7B–D shows the diagram of the array of synaptic devices and the transmission electronic microscopy (TEM) image of the phase change layer. The device can also implement neural device functions such as LTP, STP, and STDP.

2.4 Ferroelectric

A typical ferroelectric device uses a ferroelectric material and two electrodes to form a sandwich structure. Applying an electric field to the electrode changes the polarity of the ferroelectric material [120,121], which can significantly change the conductivity of the device to achieve the switching of the two conducting states. For example, in

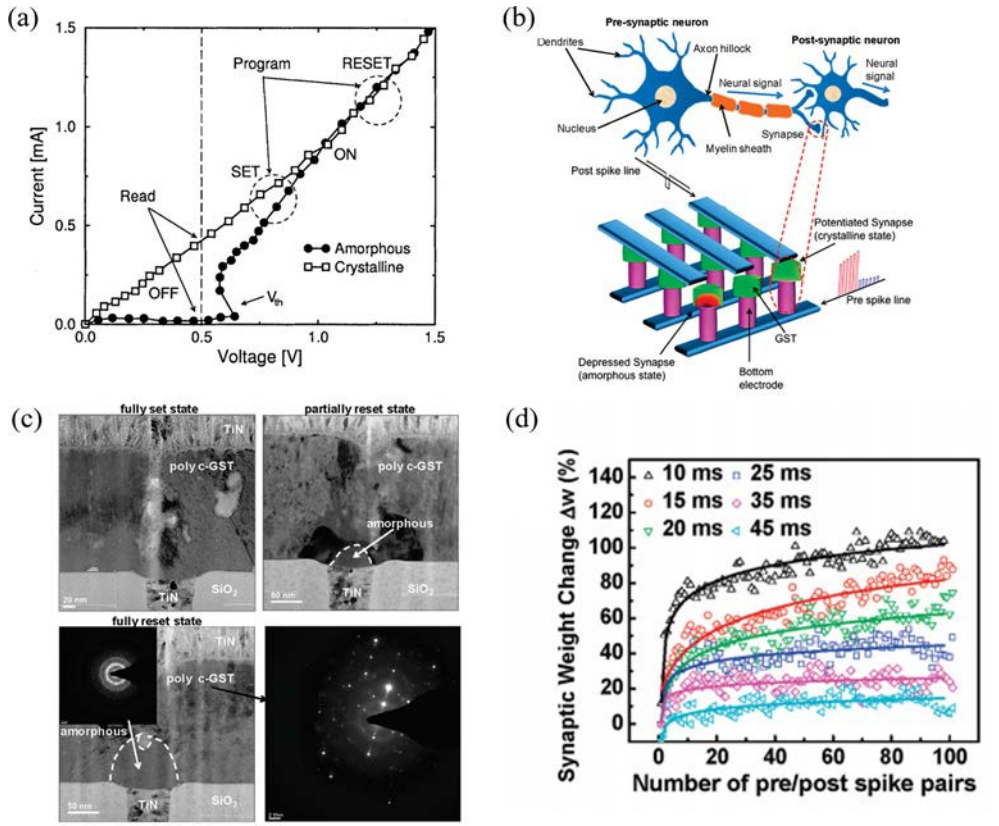


Fig. 7 (A) Experimental I–V curves for the crystalline and amorphous chalcogenide. (B) Illustration and characteristics of bio-inspired electronic synapses. The cross-sections of depressed (mushroom-shaped amorphous region shown in *red* (dark gray in the print version)) and potentiated synapses are shown in the schematic. (C) Cross-section TEM images of electronic synapses made of GST are shown. In transition from a set state to a reset state, the volume of the amorphous region at the top of the bottom electrode grows; when it fully covers the bottom electrode, the fully reset state is reached. The inset of the fully reset state shows a diffraction pattern for the mushroom-shaped amorphous region. The diffraction pattern away from the bottom electrode shows that the GST is polycrystalline. (D) Synaptic weight change as a function of a number of applied pre/postspike pairs is shown for different pre/postspike timings. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the work of Boyn et al. [122], the device resistance changes with the proportion of polarization orientation. It has been indicated that the device has the function of STDP (see Fig. 8B). For practical application in artificial synapses, a thick active layer and low electrical conductivity have become a bottleneck for ferroelectric materials. Therefore, research has been conducted to achieve an ultrathin ferroelectric layer [123] such as a ferroelectric tunnel junction [124]. In order to achieve high device stability, new

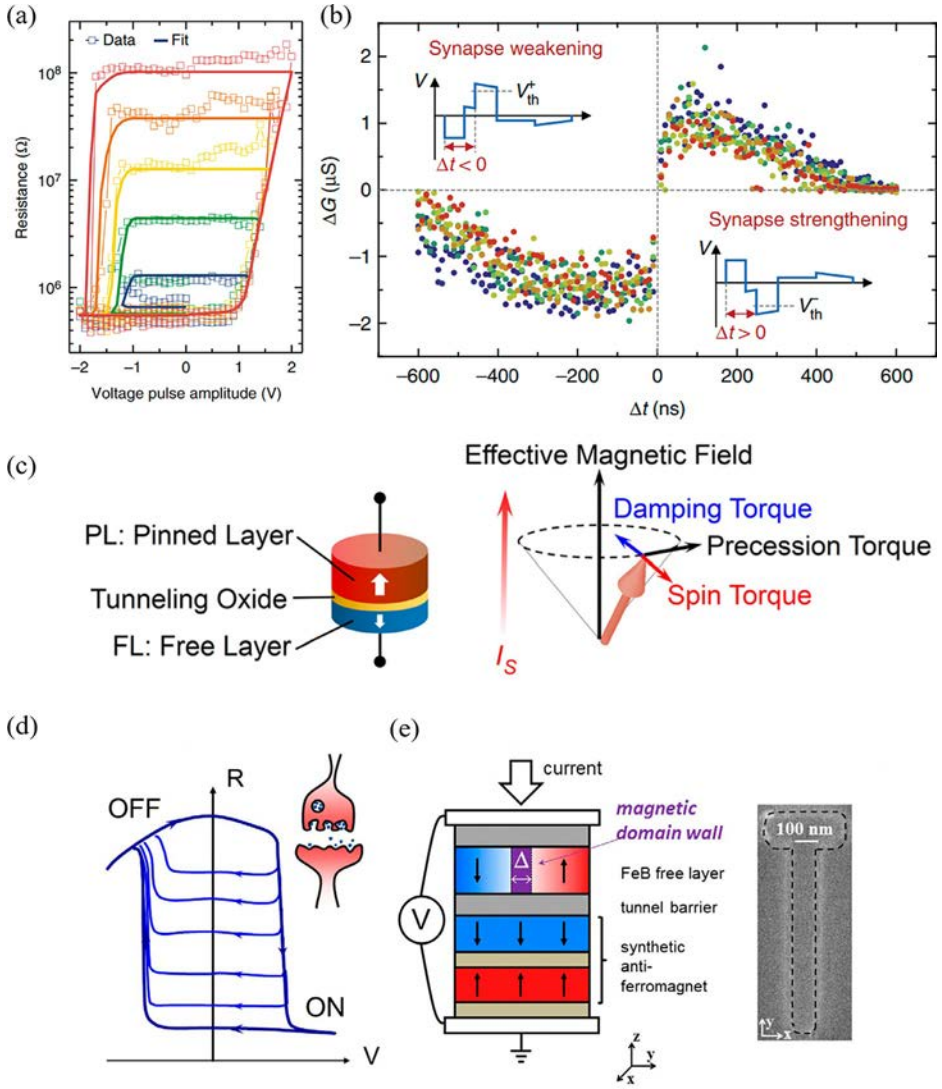


Fig. 8 (A) Multiple hysteresis loops of a ferroelectric memristor showing a clear dependence of resistance switching with the maximum pulse amplitudes. (B) Measurements of STDP in the ferroelectric memristor. (C) Vertical spin valve: a magnetic tunnel junction consists of two ferromagnets, the free layer (FL) and the pinned layer (PL), separated by a tunneling oxide barrier. The magnetization dynamics evolve under the influence of the damping torque, precession torque, and spin torque due to an input spin current, I_S. (D) Typical resistance versus voltage cycles characteristic of a magnetic memristor. (E) *Side view*: Schematic of our MgO-based magnetic tunnel junction with the domain wall in the FeB free layer. Δ is the width of the domain wall. *Top view*: Scanning electron microscope image of the sample, with a *black dashed line* to emphasize its contour.

materials such as $BaTiO_3$ [125], $Hf_{0.5}Zr_{0.5}O_2$ [126], etc., have been adopted as the active layer for ferroelectric devices.

2.5 Magnetic

Unlike ferroelectric devices, spin transfer torque-based devices have two ferromagnetic layers separated by a paraelectric insulating layer (as shown in Fig. 8C [127]). The conductivity of the device depends on the magnetization directions of these two ferromagnetic layers. For example, the device contains a thick ferromagnetic layer whose polarization direction is relatively fixed, called the pinned layer, and a thin ferromagnetic layer whose polarization direction is controlled by external stimuli, called the free layer. When the current is injected into the free layer and the relative polarization changes in the opposite direction, the conductivity can be adjusted. This three-layer structure is known as a magnetic tunnel junction (MTJ) [128–130].

In an MTJ device, adding appropriate integrating domain walls to the free layer can achieve multiple resistance states, as shown in Fig. 8D and E. For example, the processing system established by Lequeux et al. based on similar devices can realize cognitive functions [131]. In comparison with devices based on other mechanisms, the use of magnetic force to control the movement of electrons is very stable, and the durability is almost perfect because the device does not involve the movement of atoms [132].

3. Realization of neuromorphic engineering

3.1 Synaptic plasticity

One of the basic functions of an electronic synapse is neuroplasticity [133]. Based on the function of biological nerves, it is roughly divided into the following categories: long-term potentiation and depression (LTP/LTD), short-term potentiation and depression (STP/STD), spiking-time dependent plasticity (STDP), and other plasticity.

3.1.1 Long-term potentiation and depression (LTP/LTD)

In organisms, neurons can achieve long-term changes by regulating the output of neurotransmitters or controlling the number of receptors (such as α-amino-3-hydroxy-5-methyl-4-isoxazole-propionicaci, AMPAR) [133]. This behavior is referred to as LTP/LTD. Now, electronic devices are able to achieve similar functions, which means that they can maintain a state for a period of time after stimulation. This long-term plasticity is often thought to be related to biological learning behavior and is also of great significance in contemporary neural computing. This plasticity is widespread in most artificial synaptic devices. For example, the devices with conductive filaments form a stable conductive channel, which can exist for a long time, and the device works in a low resistance state. Due to the different materials and mechanisms, the LTP/LTD process may exhibit different electrical characteristics. Generally, the current–voltage (I–V) curve is

nonlinear for the devices using CFs. The reason for this may be that the formation of CFs is not a uniform process, and the conductivity will suddenly increase when the CFs are completed. This would lead to the nonuniformity of the electronic synapse arrays that consist of a large number of artificially protruding devices.

3.1.2 Short-term potentiation and depression (STP/STD)

Biological synapses can increase or decrease neurotransmitters/bioelectrical signals to achieve short-term enhancement or inhibition of neural links. This short-term plasticity has a limited duration and eventually returns to its original state. It is considered to be related to the spatiotemporal state information of the organism, and is also of great importance to expand the current computing function [134]. Short-term plasticity is ubiquitous in electronic synaptic devices. The device is in an unstable state and recovers by itself to restore the memory state by the dissolution of CFs, the dissipation of bound electrons, etc. The plasticity may be reflected in conductivity in a short time, and is significant in neural computing. However, nonlinear changes and device instability are challenges for practical applications.

Table 1 Expression and field/temperature dependency of the typical conduction mechanism in the dielectric film.

Device structure	Current density expression	Electric field dependency
Schottky emission	$J_{SE} = \frac{4\pi q m^* k T^2}{h^3} \exp\left[\frac{-q\left(\Phi_B - \sqrt{qE/4\pi\varepsilon}\right)}{kT}\right]$	$J_{SE} \propto T^2 \exp\left(A\frac{\sqrt{E}}{T} - B\right)$
Flower-Nordheim tunneling	$J_{FN} = \frac{q^2}{8\pi h \phi_B} E^2 \left(\frac{-8\pi\sqrt{2qm^*}}{3hE}\phi_B{}^{3/2}\right)$	$J_{FN} \propto E^2 \exp\left(\frac{-A}{E}\right)$
Direct tunneling	$J_{DT} \approx \exp\left(\frac{-8\pi\sqrt{2q}}{3h}\left(m^*\Phi_B\right)^{\frac{1}{2}}\kappa \cdot t_{ox,eq}\right)$	$J_{DT} \propto \exp(-\mathrm{A} \cdot t_{ox,\ eq})$
Poole-Frenkel emission	$J_{PF} = q\mu N_c E \exp\left(\frac{-q\left(\Phi_T - \sqrt{qE/\pi\varepsilon}\right)}{kT}\right)$	$J_{PF} \propto \exp\left(\mathrm{A}\frac{\sqrt{E}}{T} - \mathrm{B}\right)$
Space-charge-limited conduction (SCLC)	$J_{SCLC} = \frac{9}{8}\varepsilon_i \mu\theta \frac{V^2}{d^3}$	$J_{SCLC} \propto E^2$
Ionic conduction	$J_{ionic} \propto \frac{E}{T} \exp\left(\frac{-\Delta G^{\neq}}{kT}\right)$	$J_{ionic} \propto \frac{E}{T} \exp\left(\frac{-A}{T}\right)$
Ohmic conduction	$J_{ohmic} = \sigma E = q\mu N_c E \exp\left(\frac{-(E_c - E_F)}{kT}\right)$	$J_{ohmic} \propto E \cdot \exp\left(\frac{-A}{T}\right)$
Nearest neighbor hopping	$J_{NNH} = \sigma_0 \exp\left(\frac{-T_0}{T}\right) \cdot E$	$J_{NNH} \propto E \cdot \exp\left(\frac{-A}{T}\right)$
Variable-range hopping	$J_{VRH} = \sigma_0 \exp\left(\frac{-T_0}{T}\right)^{1/4} \cdot E$	$J_{VRH} \propto E \cdot \exp\left(\frac{-A}{T}\right)^{\frac{1}{4}}$
Trap-assisted tunneling (TAT)	$J_{TAT} = A \exp\left(\frac{-8\pi\sqrt{2qm^*}}{3hE}\phi_T{}^{\frac{3}{2}}\right)$	$J_{TAT} = \exp\left(\frac{-A}{E}\right)$

3.1.3 Spiking-time dependent plasticity (STDP)

Another common biological synaptic behavior is spike timing dependent plasticity (STDP). However, it have been shown that STDP behavior varies among organisms, and it has many manifestations [85]. In artificial synapses, the weight of synapses is affected by the interval time between two pulses. In fact, in many studies, this plasticity may be a combination of long-term plasticity and short-term plasticity, which is regarded as a Hebbian synaptic plasticity [83]. Hebb's rule or cell assembly theory believes that the continuation and repetition of reflex activity will lead to the improvement of neuron stability: when the axon of neuron A is very close to neuron B and participates in the repeated and continuous excitation of B, these two neurons or one of them will undergo growth processes or metabolic changes. As a result, A is one of the cells that can excite B, and its efficacy is enhanced. Li et al. [135] fitted the process of STPD with the following equation:

$$\Delta\omega = \begin{cases} A_+ e^{-|\Delta t|\tau_+} \\ A_- e^{-|\Delta t|\tau_-} \end{cases} \tag{1}$$

where $A_\pm$ and $\tau_\pm$ represent the scaling factor and the time constant of the function, respectively. Many studies have been conducted to investigate the STDP process in conjunction with the unique properties of the material, for example through pulse width [29]. The behavior of artificial synapse STDP is classified as asymmetric STDP, symmetric STDP, anti-STDP, and triplet STDP [85]. For example, as shown in Fig. 9A [136], with the time interval as the x-axis and the enhancement as the y-axis, the curve is symmetric about the center or the y-axis, which is considered to be a symmetric STDP, while the opposite is an asymmetric STDP (see Fig. 8B). When suppression occurs, this is known as anti-STDP (see Fig. 9B [29]). In addition, when there are three stimuli, it can be called triplet STDP [137,138]. This plasticity can reflect the inevitable interaction of pulse pairs in a sequence through short-term suppression effects.

3.1.4 Other plasticity

There are other plasticities such as metaplasticity [139], spiking-rate dependent plasticity (SRDP) [140], etc., that have been reported so far. Metaplasticity cannot shape the state of synapses, but it can change the plasticity of synapses, which can be called plasticity of plasticity. This phenomenon has been proved in various experiments, and the reaction speed and intensity of synapses can be adjusted by previous stimulations [50,141]. Although there are many articles demonstrating this plasticity, investigations of the potential applications of this kind of plasticity need to be conducted [141,142]. SRDP [140] is the dependence of presynaptic stimulation frequency on postsynaptic neuron. This kind of plasticity is similar to STDP: when the frequency of the stimulus signal meets the frequency response of the material, the output signal increases, and vice versa [137,138].

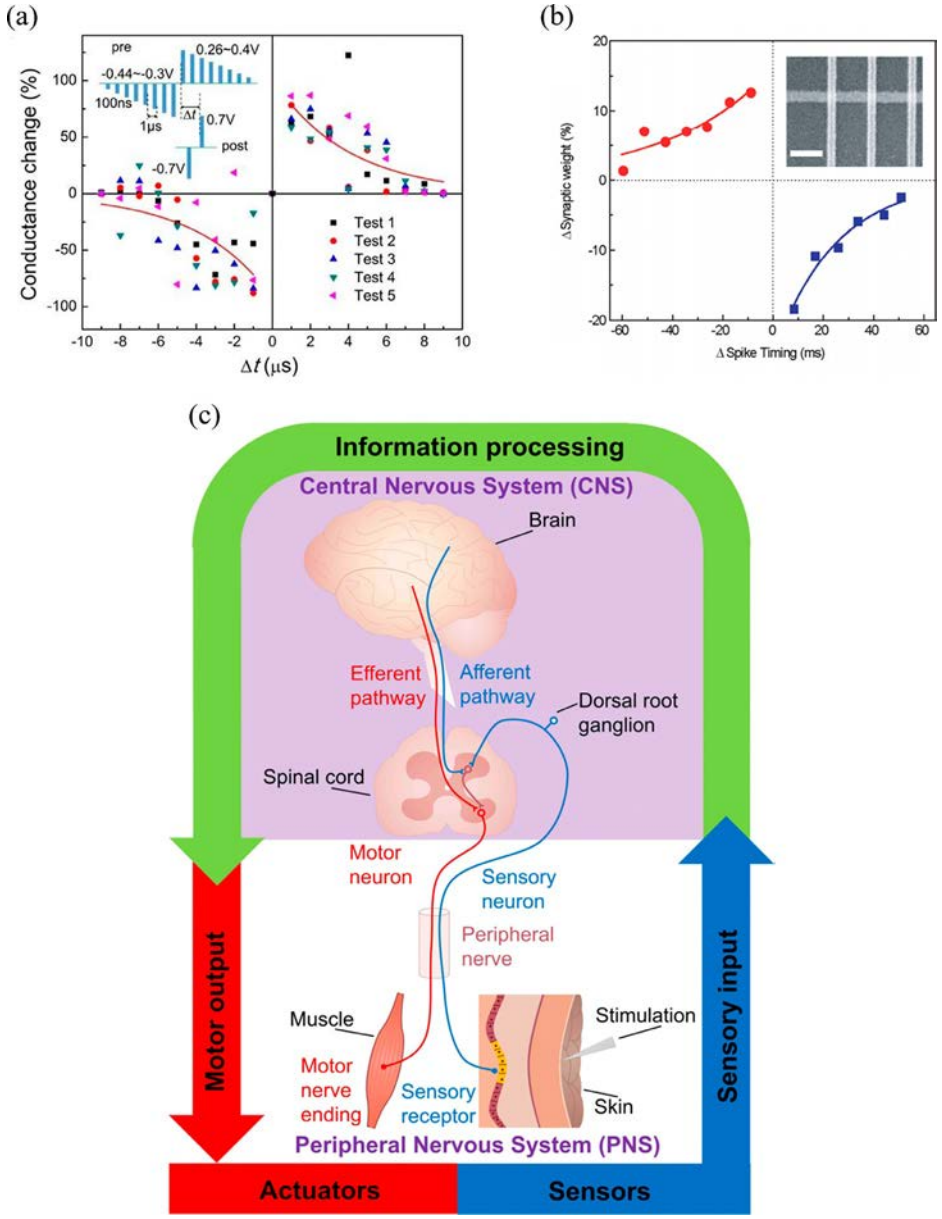


Fig. 9 (A) STDP behavior measured in a TiN/SiO_2-1 nm/TaO_x/Pt device; the *inset* shows the applied spikes to the presynaptic and postsynaptic terminals. (B) The measured change of the memristor synaptic weight vs the relative timing of the neuron spikes. (C) Schematic of the biological neuromorphic system (*inside*) and the artificial neuromorphic system (*outside*).

3.2 Integration and cognitive functions

The aforementioned synaptic devices use various mechanisms and can mimic biological nerve activity. Nonetheless, these devices are still different from the real biological nervous system. If they cannot be integrated like biological systems, it will be difficult to achieve a significant improvement in computation and energy efficiency. At present, traditional susceptors, effectors, and processors are based on microelectromechanism system (MEMS) technology. Therefore, the integration of basic synaptic devices into components with realistic neural engineering has also been the focus of recent studies.

As shown in Fig. 9C [143], the nervous system in bionic engineering should have receptors, effectors, and processing systems. The sensor receives the information and sends the information to the processing system through the sensory nerve. The processing system (central nervous system) receives the signal and generates a response signal to the outside world. The response signal is transmitted to the effector through the action nerve and finally realizes the processing of the external information.

There have been some studies using artificial synapses on the central nervous system to deal with simple tasks or as computational units. The Intel Loihi chip [144] and the IBM TrueNorth [145] chip use thousands of cores and millions of synaptic originals with extremely energy-saving properties and a small size to identify different dangerous materials, detect weapons and explosives, and find drugs and carbon monoxide [146]. The receptors are used to feel the external environment as well as to detect external movement, stimulation, deformation, and pain. It will be very important if a robot has external cognitive ability. For example, an artificial perception alarm system based on the migration of silver ions and the connection of receptors and processing systems has shown the possibility of integrating this pain-sensing device into a robot or tissue [147]. Neural engineering devices based on optical sensors have also received extensive attention. Such devices are generally based on the detection of photons, capturing photons, and converting them into electrical signals. This device realizes the characteristics of optical graphics memory, which may promote optical communication and optical computing.

3.2.1 Computing with synapses

There have been many studies combining the functions of the central nervous system and mathematical calculations, and various calculation methods have been proposed. Here, we divide the algorithms applied to synaptic devices into supervised learning, unsupervised learning, and reinforcement learning according to the training data with or without labels.

Supervised learning is a method of machine learning, the purpose of which is to learn the features in training data and build a learning model [148,149]. By mapping the synaptic weights to the crossbar array, the analog vector of voltage is obtained by conductivity matrix multiplication, as shown in Fig. 10D and E. The advantage of using an

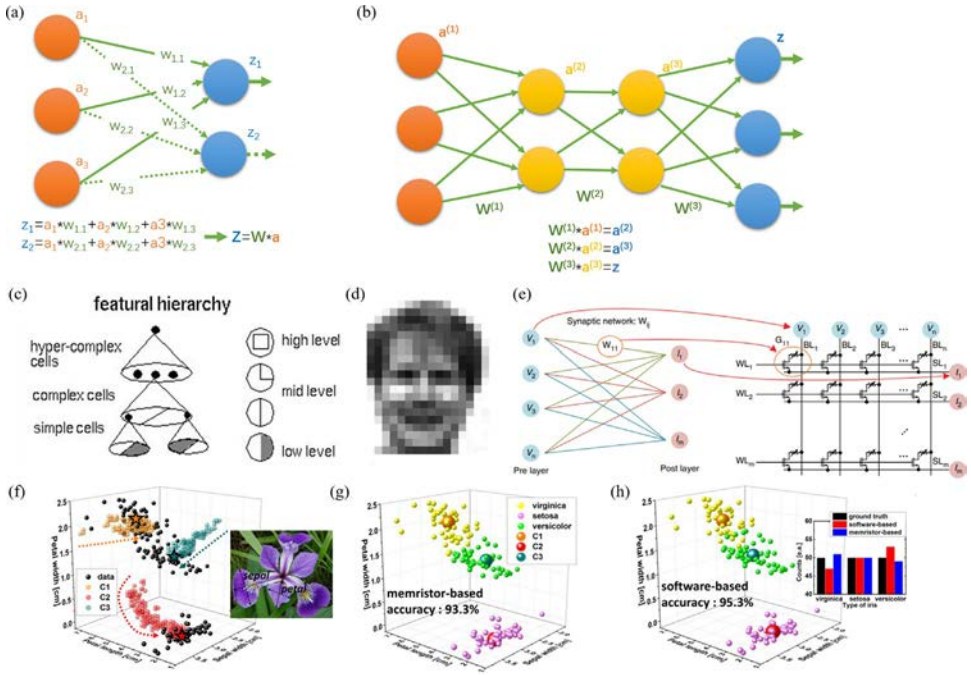


Fig. 10 (A) Schematic of a single layer neural network (perceptron). (B) Schematic of a multilayer neural network (deep learning). (C) diagram of a multilayer neural network (feature learning). (D) Input face training data (one of them). (E) Neural network mapping and hardware foundation. (F) Evolution of the centroid locations during online training with the unlabeled 3D IRIS dataset. (G) Clustering results using the memristor-based network after training. (H) Results from K-means analysis obtained from software. *Inset*: Comparison of results obtained from software- and memristor-based methods for the three types of flowers, with reference to the ground truth.

artificial synapse array is that it can speed up the core operation of the vector matrix and the efficient parallel operation, which is the most important and intensive part of most neuromorphic calculations [150]. For example, Yao used 1000 one-transistor-one-resistive memory (1T1R) cross matrices to realize face classification [151].

Reinforcement learning is a field in machine learning that emphasizes how to act based on the environment to maximize the expected benefits [152]. The inspiration comes from the behaviorist theory in psychology: how creatures, under the stimulus of rewards or punishments given by the environment, form expectations of the stimuli and produce habitual behaviors that can get the most benefit. The key is to use the artificial synapses of the array to realize the construction of the parameter Q. For example, the work of Zhongrui Wang [153] used memristors to form a 1T1R network and realized a hybrid analog-digital platform. It can acquire knowledge and solve problems without human input or supervision, obtain the Q value from the state, and solve complicated tasks.

Unsupervised learning is another kind of machine learning method. Given no previously labeled training examples, it can automatically classify or cluster the input data [154]. The main applications of unsupervised learning include cluster analysis, association rule, and dimensionality reduction. It is an alternative to previous strategies such as supervised learning and reinforcement learning. This learning method can also be mapped to the artificial synapses array. For example, *K*-means clustering (dividing n points into *k* clusters), which is often used in data mining, can be implemented using artificial neural networks. Jeong et al. [155] built a system that can greatly accelerate the speed of neural computing, implement the *K*-means clustering algorithm through online learning, and produce a high classification accuracy (93.3%) for the standard IRIS dataset.

3.2.2 Artificial neuromorphic devices

Artificial neuromorphic systems have also received intensive attention. For example, Seunghwan Seo et al. [79] developed an artificial optic-neural synapse by integrating optical sensors and neural networks. By converting external light information into electrical signals (Fig. 11B), it can simulate the function of identifying colors and graphics in biological vision. As shown in Fig. 11A, the author constructed visual sensors and artificial synapses to imitate human vision. They further demonstrated the identification of numbers. Shuai Chen et al. [156] designed an array device that has a memory of the stimuli it has received, similar to human visual remnants, as shown in Fig. 11D.

In addition to vision sense, tactile sense is another common bionic application that has been widely studied. As shown in Fig. 11C, the artificial sensor neural device invented by Wan et al. [156] combines a pressure sensor and ion gel to realize tactile recognition in time and space. Fig. 11E shows how multiple tactile sensors work together in artificial nerve cells. The device used electrical signals as feedback. Kim et al. realized the biological circuit of the sensor to obtain tactile information and finally transmit it to the effector [157], as shown in Fig. 11F. The sensor array can obtain exquisite pressure information,

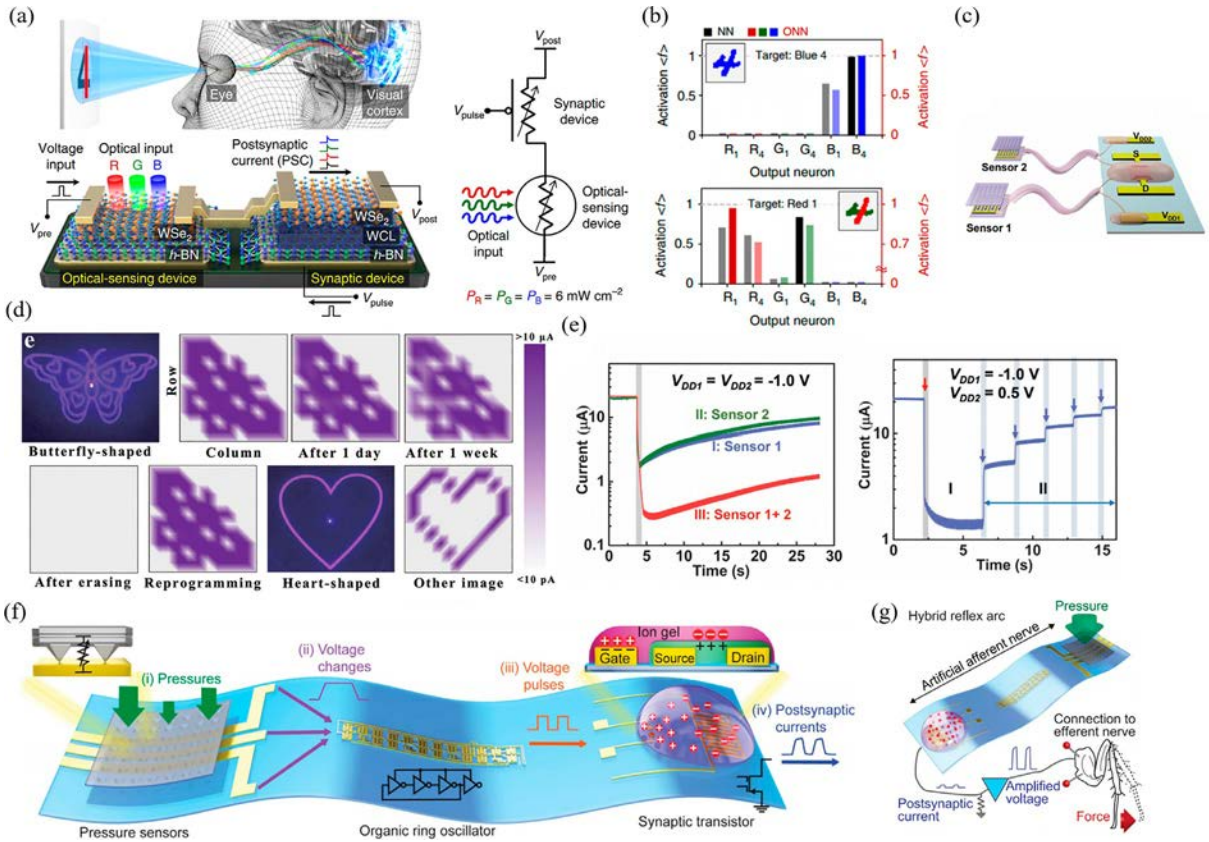


Fig. 11 (A) Schematic of the human optic nerve system, the h-BN/WSe2 synaptic device integrated with the h-BN/WSe2 photodetector, and the simplified electrical circuit for the ONS device. (B) Activation values of each output neuron in cases of a single-colored number (*blue 4* (dark gray in the print version)) and a color-mixed number (*red/green-mixed 4* (gray in the print version)) after the 600th training epoch. (C) Diagram illustrating the configuration of the NeuTap neuron with two pressure sensors. (D) The array of synaptic devices has a memory effect on graphics, for example, The shape of a butterfly can be remembered for a long time, and can retrospective memory. (E) Two NeuTaps working separately and working together (*left*); two NeuTap interactive operations (*right*). (F) An artificial afferent nerve made of pressure sensors, an organic ring oscillator, and a synaptic transistor. (G) Hybrid reflex arc made of an artificial afferent nerve and a biological efferent nerve. Postsynaptic currents are amplified to stimulate biological efferent nerves and muscles to initiate movement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

such as the movement of objects, and connect the artificial afferent nerve to construct a hybrid bioelectronic reflex arc to actuate muscles, as shown in Fig. 11G.

4. Summary

Neuromorphic engineering based on electronic synapses has shown remarkable achievements. Synaptic devices with different operating mechanisms are booming: mature traditional devices have already been commercialized while devices based on new materials and innovative structures are being developed rapidly. At the same time, the application of electronic synapses still requires further exploration. Synapse devices in computing need high precision, linear feedback, high uniformity, high switching speed, and high stability. Furthermore, a new type of circuit design is required to be compatible with this new generation of processing and storage systems, and new algorithms need to be developed for this system to improve the computing capability. So far, there are no computers that use electronic synapses practically. Despite the flourishing of electronic synaptic devices, much application research is still needed to demonstrate the potential of electronic synapses in the field of neuromorphic computing.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant 2016YFB0401305) and the National Natural Science Foundation of China (Grant U1605244).

Conflict of interest

The authors declare no conflict of interest.

References

[1] C. Mead, Analog VLSI Implementation of Neural Systems, Springer, 1989, p. 239.
[2] J. Yon Neumann, The Computer and the Brain, Yale Uni, New Haven, CT, 1958.
[3] G.E. Moore, IEEE Solid State Circuits Soc. Newslett. 11 (2006) 33.
[4] M.A. Zidan, J.P. Strachan, W.D. Lu, Nat. Electron. 1 (2018) 22.
[5] J. Backus, Commun. ACM 21 (1978) 613.
[6] D. Ielmini, Microelectron. Eng. 190 (2018) 44.
[7] D.O. Hebb, The Organization of Behavior: A Neuropsychological Theory, J. Wiley; Chapman & Hall, 1949.
[8] D.O. Hebb, Psychol. Rev. 62 (1955) 243.
[9] J. Hopfield, Proc. Natl. Acad. Sci. 79 (1982) 2554.
[10] E.M. Izhikevich, Dynamical Systems in Neuroscience, MIT Press, 2007.
[11] E.M. Izhikevich, IEEE Trans. Neural Netw. 14 (2003) 1569.
[12] E.M. Izhikevich, IEEE Trans. Neural Netw. 15 (2004) 1063.
[13] C.-S. Cheng, Comput. Ind. Eng. 28 (1995) 51.
[14] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, C. Guo, Y. Nakamura, Science 345 (2014) 668.

[15] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, IEEE Micro 38 (2018) 82.
[16] S.B. Furber, F. Galluppi, S. Temple, L.A. Plana, Proc. IEEE 102 (2014) 652.
[17] B. Govoreanu, G. Kar, Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, 10 × 10 nm2 Hf/HfO x crossbar resistive RAM with excellent performance, reliability and low-energy operation, in: Presented at 2011 International Electron Devices Meeting, 2011.
[18] J.J. Yang, D.B. Strukov, D.R. Stewart, Nat. Nanotechnol. 8 (2013) 13.
[19] B.J. Choi, A.C. Torrezan, J.P. Strachan, P. Kotula, A. Lohn, M.J. Marinella, Z. Li, R.S. Williams, J.J. Yang, Adv. Funct. Mater. 26 (2016) 5290.
[20] K.-H. Kim, S. Hyun Jo, S. Gaba, W. Lu, Appl. Phys. Lett. 96 (2010), 053106.
[21] J. Zhou, F. Cai, Q. Wang, B. Chen, S. Gaba, W.D. Lu, IEEE Electron Device Lett. 37 (2016) 404.
[22] J. Åkerman, Science 308 (2005) 508.
[23] J.J. Yang, M.D. Pickett, X. Li, D.A. Ohlberg, D.R. Stewart, R.S. Williams, Nat. Nanotechnol. 3 (2008) 429.
[24] P. Zikopoulos, C. Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Osborne Media, 2011.
[25] D. Monroe, Commun. ACM 57 (6) (2014) 13.
[26] W. Zhao, G. Agnus, V. Derycke, A. Filoramo, J. Bourgoin, C. Gamrat, Nanotechnology 21 (2010), 175202.
[27] Y. Zhou, S. Ramanathan, Proc. IEEE 103 (2015) 1289.
[28] L. Jamilpanah, S. Azizmohseni, S.A. Hosseini, M. Hasheminejad, N. Vesali, A. Iraji Zad, M. Pourfath, S.M. Mohseni, Phys. Status Solidi Rapid Res. Lett. 12 (2018) 1800418.
[29] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nano Lett. 10 (2010) 1297.
[30] M. Yang, X. Zhao, Q. Tang, N. Cui, Z. Wang, Y. Tong, Y. Liu, Nanoscale 10 (2018) 18135.
[31] X. Yan, J. Zhao, S. Liu, Z. Zhou, Q. Liu, J. Chen, X.Y. Liu, Adv. Funct. Mater. 28 (2018) 1705320.
[32] W. Chen, R. Fang, M.B. Balaban, W. Yu, Y. Gonzalez-Velo, H.J. Barnaby, M.N. Kozicki, Nanotechnology 27 (2016), 255202.
[33] I. Valov, R. Waser, J.R. Jameson, M.N. Kozicki, Nanotechnology 22 (2011), 254003.
[34] R. Waser, R. Dittmann, G. Staikov, K. Szot, Adv. Mater. 21 (2009) 2632.
[35] F. Zhuge, K. Li, B. Fu, H. Zhang, J. Li, H. Chen, L. Liang, J. Gao, H. Cao, Z. Liu, AIP Adv. 5 (2015), 057125.
[36] S. Menzel, U. Böttger, R. Waser, J. Appl. Phys. 111 (2012), 014501.
[37] S. Prada, M. Rosa, L. Giordano, C. Di Valentin, G. Pacchioni, Phys. Rev. B 83 (2011), 245314.
[38] Y. Yang, P. Gao, L. Li, X. Pan, S. Tappertzhofen, S. Choi, R. Waser, I. Valov, W.D. Lu, Nat. Commun. 5 (2014) 1.
[39] X. Guo, C. Schindler, S. Menzel, R. Waser, Appl. Phys. Lett. 91 (2007), 133513.
[40] Y. Yang, P. Gao, S. Gaba, T. Chang, X. Pan, W. Lu, Nat. Commun. 3 (2012) 1.
[41] B.K. You, J.M. Kim, D.J. Joe, K. Yang, Y. Shin, Y.S. Jung, K.J. Lee, ACS Nano 10 (2016) 9478.
[42] X. Zhao, J. Ma, X. Xiao, Q. Liu, L. Shao, D. Chen, S. Liu, J. Niu, X. Zhang, Y. Wang, Adv. Mater. 30 (2018) 1705193.
[43] X. Zhao, S. Liu, J. Niu, L. Liao, Q. Liu, X. Xiao, H. Lv, S. Long, W. Banerjee, W. Li, Small 13 (2017) 1603948.
[44] S. La Barbera, D. Vuillaume, F. Alibart, ACS Nano 9 (2015) 941.
[45] Z. Wang, M. Rao, R. Midya, S. Joshi, H. Jiang, P. Lin, W. Song, S. Asapu, Y. Zhuo, C. Li, Adv. Funct. Mater. 28 (2018) 1704862.
[46] M. Fujimoto, H. Koyama, M. Konagai, Y. Hosoi, K. Ishihara, S. Ohnishi, N. Awaya, Appl. Phys. Lett. 89 (2006), 223509.
[47] H.-S.P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F.T. Chen, M.-J. Tsai, Proc. IEEE 100 (2012) 1951.
[48] J.-Y. Chen, C.-L. Hsin, C.-W. Huang, C.-H. Chiu, Y.-T. Huang, S.-J. Lin, W.-W. Wu, L.-J. Chen, Nano Lett. 13 (2013) 3671.
[49] D.-H. Kwon, K.M. Kim, J.H. Jang, J.M. Jeon, M.H. Lee, G.H. Kim, X.-S. Li, G.-S. Park, B. Lee, S. Han, Nat. Nanotechnol. 5 (2010) 148.

[50] Z.H. Tan, R. Yang, K. Terabe, X.B. Yin, X.D. Zhang, X. Guo, Adv. Mater. 28 (2016) 377.
[51] J.P. Strachan, M.D. Pickett, J.J. Yang, S. Aloni, A. David Kilcoyne, G. Medeiros-Ribeiro, R. Stanley Williams, Adv. Mater. 22 (2010) 3573.
[52] S. Seo, M. Lee, D. Seo, E. Jeoung, D.-S. Suh, Y. Joung, I. Yoo, I. Hwang, S. Kim, I. Byun, Appl. Phys. Lett. 85 (2004) 5655.
[53] W.-Y. Chang, Y.-C. Lai, T.-B. Wu, S.-F. Wang, F. Chen, M.-J. Tsai, Appl. Phys. Lett. 92 (2008), 022110.
[54] K.M. Kim, B.J. Choi, B.W. Koo, S. Choi, D.S. Jeong, C.S. Hwang, Electrochem. Solid-State Lett. 9 (2006) G343.
[55] J. Choi, J.-S. Kim, I. Hwang, S. Hong, S. Jeon, S.-O. Kang, B. Park, D. Kim, M. Lee, S. Seo, Appl. Phys. Lett. 95 (2009), 022109.
[56] N. Xu, L. Liu, X. Sun, C. Chen, Y. Wang, D. Han, X. Liu, R. Han, J. Kang, B. Yu, Semicond. Sci. Technol. 23 (2008), 075019.
[57] M. Bocquet, D. Deleruyelle, C. Muller, J.-M. Portal, Appl. Phys. Lett. 98 (2011), 263507.
[58] E.W. Lim, R. Ismail, Electronics 4 (2015) 586.
[59] V.K. Sangwan, D. Jariwala, I.S. Kim, K.-S. Chen, T.J. Marks, L.J. Lauhon, M.C. Hersam, Nat. Nanotechnol. 10 (2015) 403.
[60] K. Baek, S. Park, J. Park, Y.-M. Kim, H. Hwang, S.H. Oh, Nanoscale 9 (2017) 582.
[61] C.S. Yang, D.S. Shang, N. Liu, E.J. Fuller, S. Agrawal, A.A. Talin, Y.Q. Li, B.G. Shen, Y. Sun, Adv. Funct. Mater. 28 (2018) 1804170.
[62] R. Burt, G. Birkett, X. Zhao, Phys. Chem. Chem. Phys. 16 (2014) 6519.
[63] P. Sharma, T. Bhatti, Energy Convers. Manag. 51 (2010) 2901.
[64] S. Ono, S. Seki, R. Hirahara, Y. Tominari, J. Takeya, Appl. Phys. Lett. 92 (2008) 93.
[65] C.-L. Tsai, F. Xiong, E. Pop, M. Shim, ACS Nano 7 (2013) 5360.
[66] J. Wang, Y. Li, R. Liang, Y. Zhang, W. Mao, Y. Yang, T.-L. Ren, IEEE Electron Device Lett. 38 (2017) 1496.
[67] J. Wang, Y. Li, Y. Yang, T.-L. Ren, IEEE Electron Device Lett. 38 (2017) 588.
[68] X. Wan, Y. Yang, P. Feng, Y. Shi, Q. Wan, IEEE Electron Device Lett. 37 (2016) 299.
[69] J. Wen, L.Q. Zhu, Y.M. Fu, H. Xiao, L.Q. Guo, Q. Wan, ACS Appl. Mater. Interfaces 9 (2017) 37064.
[70] L.Q. Zhu, C.J. Wan, P.Q. Gao, Y.H. Liu, H. Xiao, J.C. Ye, Q. Wan, ACS Appl. Mater. Interfaces 8 (2016) 21770.
[71] Y.M. Fu, C.J. Wan, L.Q. Zhu, H. Xiao, X.D. Chen, Q. Wan, Adv. Biosyst. 2 (2018) 1700198.
[72] G. Wu, P. Feng, X. Wan, L. Zhu, Y. Shi, Q. Wan, Sci. Rep. 6 (2016) 1.
[73] G. Gao, J. Yu, X. Yang, Y. Pang, J. Zhao, C. Pan, Q. Sun, Z.L. Wang, Adv. Mater. 31 (2019) 1806905.
[74] Y. Chen, G. Gao, J. Zhao, H. Zhang, J. Yu, X. Yang, Q. Zhang, W. Zhang, S. Xu, J. Sun, Adv. Funct. Mater. 29 (2019) 1900959.
[75] I. Sanchez Esqueda, X. Yan, C. Rutherglen, A. Kane, T. Cain, P. Marsh, Q. Liu, K. Galatsis, H. Wang, C. Zhou, ACS Nano 12 (2018) 7352.
[76] C.S. Yang, D.S. Shang, N. Liu, G. Shi, X. Shen, R.C. Yu, Y.Q. Li, Y. Sun, Adv. Mater. 29 (2017) 1700906.
[77] X. Gu, S.S. Iyer, IEEE Electron Device Lett. 38 (2017) 1204.
[78] L. Wang, J. Yang, Y. Zhu, M. Yi, L. Xie, R. Ju, Z. Wang, L. Liu, T. Li, C. Zhang, Adv. Electron. Mater. 3 (2017).
[79] S. Seo, S.-H. Jo, S. Kim, J. Shim, S. Oh, J.-H. Kim, K. Heo, J.-W. Choi, C. Choi, S. Oh, Nat. Commun. 9 (2018) 1.
[80] Z. Xu, F. Li, C. Wu, F. Ma, Y. Zheng, K. Yang, W. Chen, H. Hu, T. Guo, T.W. Kim, NPG Asia Mater. 11 (2019) 1.
[81] F. Ma, Z. Xu, Y. Liu, Y. Zheng, W. Chen, H. Hu, T. Guo, F. Li, C. Wu, T.W. Kim, IEEE Electron Device Lett. 40 (2019) 1610.
[82] R. Pan, J. Li, F. Zhuge, L. Zhu, L. Liang, H. Zhang, J. Gao, H. Cao, B. Fu, K. Li, Appl. Phys. Lett. 108 (2016), 013504.
[83] K.D. Miller, D.J. MacKay, Neural Comput. 6 (1994) 100.

[84] H.-X. Wang, R.C. Gerkin, D.W. Nauen, G.-Q. Bi, Nat. Neurosci. 8 (2005) 187.
[85] L.F. Abbott, S.B. Nelson, Nat. Neurosci. 3 (2000) 1178.
[86] S. Aghnout, G. Karimi, M.R. Azghadi, J. Comput. Electron. 16 (2017) 401.
[87] X. Yan, L. Zhang, H. Chen, X. Li, J. Wang, Q. Liu, C. Lu, J. Chen, H. Wu, P. Zhou, Adv. Funct. Mater. 28 (2018) 1803728.
[88] C. Wang, W. He, Y. Tong, Y. Zhang, K. Huang, L. Song, S. Zhong, R. Ganeshkumar, R. Zhao, Small 13 (2017) 1603435.
[89] H.-T. Peng, M.A. Nahmias, T.F. De Lima, A.N. Tait, B.J. Shastri, IEEE J. Sel. Top. Quantum Electron. 24 (2018) 1.
[90] M.A. Nahmias, B.J. Shastri, A.N. Tait, T.F. De Lima, P.R. Prucnal, Opt. Photonics News 29 (2018) 34.
[91] B. Li, W. Wei, X. Yan, X. Zhang, P. Liu, Y. Luo, J. Zheng, Q. Lu, Q. Lin, X. Ren, Nanotechnology 29 (2018), 464004.
[92] J. Park, S. Lee, J. Lee, K. Yong, Adv. Mater. 25 (2013) 6423.
[93] A. Mehonic, T. Gerard, A. Kenyon, Appl. Phys. Lett. 111 (2017), 233502.
[94] M. Ungureanu, R. Zazpe, F. Golmar, P. Stoliar, R. Llopis, F. Casanova, L.E. Hueso, Adv. Mater. 24 (2012) 2496.
[95] M. Kumar, S. Abbas, J. Kim, ACS Appl. Mater. Interfaces 10 (2018) 34370.
[96] R.A. John, J. Ko, M.R. Kulkarni, N. Tiwari, N.A. Chien, N.G. Ing, W.L. Leong, N. Mathews, Small 13 (2017) 1701193.
[97] J. Wang, Y. Chen, L.-A. Kong, Y. Fu, Y. Gao, J. Sun, Appl. Phys. Lett. 113 (2018), 151101.
[98] K. Ueda, S. Aichi, H. Asano, Appl. Phys. Lett. 108 (2016), 222102.
[99] C.S. Wang, C.H. Nieh, T.Y. Lin, Y.F. Chen, Adv. Funct. Mater. 25 (2015) 4058.
[100] P. Russo, M. Xiao, R. Liang, N.Y. Zhou, Adv. Funct. Mater. 28 (2018) 1706230.
[101] Z. Cheng, C. Ríos, W.H. Pernice, C.D. Wright, H. Bhaskaran, Sci. Adv. 3 (2017), e1700160.
[102] C. Ríos, M. Stegmaier, P. Hosseini, D. Wang, T. Scherer, C.D. Wright, H. Bhaskaran, W.H. Pernice, Nat. Photonics 9 (2015) 725.
[103] P.R. Prucnal, B.J. Shastri, T.F. de Lima, M.A. Nahmias, A.N. Tait, Adv. Opt. Photon. 8 (2016) 228.
[104] Y. Wang, Z. Lv, J. Chen, Z. Wang, Y. Zhou, L. Zhou, X. Chen, S.T. Han, Adv. Mater. 30 (2018) 1802883.
[105] Y. Yang, Y. He, S. Nie, Y. Shi, Q. Wan, IEEE Electron Device Lett. 39 (2018) 897.
[106] P. Junsangsri, F. Lombardi, J. Han, Macromodeling a phase change memory (PCM) cell by HSPICE, in: Presented at Proceedings of the 2012 IEEE/ACM International Symposium on Nanoscale Architectures, 2012.
[107] A. Pirovano, A.L. Lacaita, A. Benvenuti, F. Pellizzer, R. Bez, IEEE Trans. Electron Devices 51 (2004) 452.
[108] E.R. Meinders, A.V. Mijiritskii, L. van Pieterson, M. Wuttig, Optical Data Storage: Phase-Change Media and Recording, vol. 4, Springer Science & Business Media, 2006.
[109] P. Hosseini, C.D. Wright, H. Bhaskaran, Nature 511 (2014) 206.
[110] P.K. Khulbe, E.M. Wright, M. Mansuripur, J. Appl. Phys. 88 (2000) 3926.
[111] B.-S. Lee, G.W. Burr, R.M. Shelby, S. Raoux, C.T. Rettner, S.N. Bogle, K. Darmawikarta, S.G. Bishop, J.R. Abelson, Science 326 (2009) 980.
[112] K. Ren, R. Li, X. Chen, Y. Wang, J. Shen, M. Xia, S. Lv, Z. Ji, Z. Song, Appl. Phys. Lett. 112 (2018), 073106.
[113] F. Rao, K. Ding, Y. Zhou, Y. Zheng, M. Xia, S. Lv, Z. Song, S. Feng, I. Ronneberger, R. Mazzarello, Science 358 (2017) 1423.
[114] S. Kohara, K. Kato, S. Kimura, H. Tanaka, T. Usuki, K. Suzuya, H. Tanaka, Y. Moritomo, T. Matsunaga, N. Yamada, Appl. Phys. Lett. 89 (2006), 201910.
[115] M. Zhu, M. Xia, F. Rao, X. Li, L. Wu, X. Ji, S. Lv, Z. Song, S. Feng, H. Sun, Nat. Commun. 5 (2014) 1.
[116] X.-P. Wang, N.-K. Chen, X.-B. Li, Y. Cheng, X. Liu, M.-J. Xia, Z. Song, X. Han, S. Zhang, H.-B. Sun, Phys. Chem. Chem. Phys. 16 (2014) 10810.

[117] Y. Chen, C. Rettner, S. Raoux, G. Burr, S. Chen, R. Shelby, M. Salinga, W. Risk, T. Happ, G. McClelland, Ultra-thin phase-change bridge memory device using GeSb, in: Presented at 2006 International Electron Devices Meeting, 2006.
[118] G.B. Beneventi, L. Perniola, V. Sousa, E. Gourvest, S. Maitrejean, J. Bastien, A. Bastard, B. Hyot, A. Fargeix, C. Jahan, Solid State Electron. 65 (2011) 197.
[119] D. Kuzum, R.G. Jeyasingh, B. Lee, H.-S.P. Wong, Nano Lett. 12 (2012) 2179.
[120] A. Chanthbouala, A. Crassous, V. Garcia, K. Bouzehouane, S. Fusil, X. Moya, J. Allibe, B. Dlubak, J. Grollier, S. Xavier, Nat. Nanotechnol. 7 (2012) 101.
[121] V. Garcia, M. Bibes, Nat. Commun. 5 (2014) 1.
[122] S. Boyn, J. Grollier, G. Lecerf, B. Xu, N. Locatelli, S. Fusil, S. Girod, C. Carrétéro, K. Garcia, S. Xavier, Nat. Commun. 8 (2017) 1.
[123] J.C. Wong, S. Salahuddin, Proc. IEEE 107 (2018) 49.
[124] D.S. Jeong, C.S. Hwang, Adv. Mater. 30 (2018) 1704729.
[125] R. Soni, A. Petraru, P. Meuffels, O. Vavra, M. Ziegler, S.K. Kim, D.S. Jeong, N.A. Pertsev, H. Kohlstedt, Nat. Commun. 5 (2014) 1.
[126] M. Hoffmann, F.P. Fengler, M. Herzig, T. Mittmann, B. Max, U. Schroeder, R. Negrea, P. Lucian, S. Slesazeck, T. Mikolajick, Nature 565 (2019) 464.
[127] A. Sengupta, K. Roy, Appl. Phys. Rev. 4 (2017), 041105.
[128] J.C. Slonczewski, J. Magn. Magn. Mater. 159 (1996) L1.
[129] L. Berger, Phys. Rev. B 54 (1996) 9353.
[130] L. Liu, C.-F. Pai, Y. Li, H. Tseng, D. Ralph, R. Buhrman, Science 336 (2012) 555.
[131] S. Lequeux, J. Sampaio, V. Cros, K. Yakushiji, A. Fukushima, R. Matsumoto, H. Kubota, S. Yuasa, J. Grollier, Sci. Rep. 6 (2016) 1.
[132] N. Locatelli, V. Cros, J. Grollier, Nat. Mater. 13 (2014) 11.
[133] E.R. Kandel, J.H. Schwartz, T.M. Jessell, Department of Biochemistry, M.B.T. Jessell, S. Siegelbaum, A. Hudspeth, Principles of Neural Science, vol. 4, McGraw-hill, New York, 2000.
[134] C. Du, F. Cai, M.A. Zidan, W. Ma, S.H. Lee, W.D. Lu, Nat. Commun. 8 (2017) 1.
[135] Y. Li, Y. Zhong, J. Zhang, L. Xu, Q. Wang, H. Sun, H. Tong, X. Cheng, X. Miao, Sci. Rep. 4 (2014) 4906.
[136] Z. Wang, M. Yin, T. Zhang, Y. Cai, Y. Wang, Y. Yang, R. Huang, Nanoscale 8 (2016) 14015.
[137] R. Yang, H.M. Huang, Q.H. Hong, X.B. Yin, Z.H. Tan, T. Shi, Y.X. Zhou, X.S. Miao, X.P. Wang, S.B. Mi, Adv. Funct. Mater. 28 (2018) 1704455.
[138] W. Cai, F. Ellinger, R. Tetzlaff, IEEE Trans. Biomed. Circuits Syst. 9 (2014) 87.
[139] W.C. Abraham, S.E. Mason-Parker, M.F. Bear, S. Webb, W.P. Tate, Proc. Natl. Acad. Sci. U. S. A. 98 (2001) 10924.
[140] G. Rachmuth, H.Z. Shouval, M.F. Bear, C.-S. Poon, Proc. Natl. Acad. Sci. U. S. A. 108 (2011) E1266.
[141] X. Zhu, C. Du, Y. Jeong, W.D. Lu, Nanoscale 9 (2017) 45.
[142] Q. Wu, H. Wang, Q. Luo, W. Banerjee, J. Cao, X. Zhang, F. Wu, Q. Liu, L. Li, M. Liu, Nanoscale 10 (2018) 5875.
[143] J. Zhu, T. Zhang, Y. Yang, R. Huang, Appl. Phys. Rev. 7 (2020), 011312.
[144] N. Imam, T.A. Cleland, Nat. Mach. Intell. 2 (2020) 181.
[145] K.M. Song, J.-S. Jeong, B. Pan, X. Zhang, J. Xia, S. Cha, T.-E. Park, K. Kim, S. Finizio, J. Raabe, Nat. Electron. 3 (2020) 148.
[146] S. Choi, J.H. Shin, J. Lee, P. Sheridan, W.D. Lu, Nano Lett. 17 (2017) 3113.
[147] J.H. Yoon, Z. Wang, K.M. Kim, H. Wu, V. Ravichandran, Q. Xia, C.S. Hwang, J.J. Yang, Nat. Commun. 9 (2018) 1.
[148] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall Upper Saddle River, 2002.
[149] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Presented at Proceedings of the 23rd International Conference on Machine Learning, 2006.
[150] M.I. Jordan, D.E. Rumelhart, Cogn. Sci. 16 (1992) 307.
[151] P. Yao, H. Wu, B. Gao, S.B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S.P. Wong, Nat. Commun. 8 (2017) 1.

[152] R.S. Sutton, A.G. Barto, Introduction to Reinforcement Learning, vol. 135, MIT Press, Cambridge, 1998.
[153] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, Nat. Electron. 2 (2019) 115.
[154] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009, p. 485.
[155] Y. Jeong, J. Lee, J. Moon, J.H. Shin, W.D. Lu, Nano Lett. 18 (2018) 4447.
[156] C. Wan, G. Chen, Y. Fu, M. Wang, N. Matsuhisa, S. Pan, L. Pan, H. Yang, Q. Wan, L. Zhu, Adv. Mater. 30 (2018) 1801291.
[157] Y. Kim, A. Chortos, W. Xu, Y. Liu, J.Y. Oh, D. Son, J. Kang, A.M. Foudeh, C. Zhu, Y. Lee, Science 360 (2018) 998.

CHAPTER 8

Nanowire memristor as artificial synapse in random networks

G. Milano[a] **and C. Ricciardi**[b]
[a]Advanced Materials Metrology and Life Sciences Division, Italian National Institute of Metrology Research, Torino, Italy
[b]Applied Science and Technology Department (DISAT), Politecnico di Torino, Torino, Italy

1. Introduction

The rapid development of information and communication technologies and the rising of artificial intelligence (AI) is currently challenging the state-of-art of nanoelectronics with the end of the Moore's law in sight. In this framework, there is an ever-growing interest in new electronic devices and architectures for the hardware implementation of computing paradigms beyond von Neumann. Among emerging technologies, memristive devices represent one of the candidates for next-generation electronics [1–3]. The experimental realization of a "memristor" (memory + resistor), theorized by L. Chua in 1971 as a fourth fundament circuit element [4], has been reported in 2008 where the theoretical concept of memristor was associated to resistive switching phenomena by Strukov et al. [5]. Memristive devices are two-terminal capacitor-like devices where an active material (typically an insulating metal-oxide) is sandwiched in between two metal electrodes [6–9]. In these devices, the reconfiguration of the active material upon electrical stimulation results in a dependence of the internal resistance state on the history of applied voltage and current. In this framework, the change in the physical parameters of the device and their dynamics can be exploited to both store and process information. For this reason, they represents a potential breakthrough in the areas of on-chip memory and storage, in-memory computing and for the hardware realization of biologically inspired computing architectures [2,10–16].

In this framework, memristive devices based on nanowires (NWs) have attracted a growing interest as the building blocks for a memristive-based nanoelectronics [17–20]. In particular, single NWs have been considered good platforms for the investigation of physicochemical phenomena underlying resistive switching events and for neuromorphic-type of data processing by emulating typical features of biological synapses. Furthermore, NW networks can be exploited for the realization of brain-inspired architectures for the implementation of unconventional type of computing paradigms.

In this chapter, we report on memristive devices and nanoarchitectures based on NWs. In the first section, fundamentals of NW memristive devices are discussed from

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00008-1
Copyright © 2023 Elsevier Inc.
All rights reserved.

the NW synthesis to device fabrication strategies. Also, resistive switching mechanisms including electrochemical metallization mechanism (ECM) and valence change mechanism (VCM) are discussed, showing that single NW devices represents good platform for the investigation of physicochemical mechanisms underlaying these switching effects. In the second section, neuromorphic functionalities of single-NW-based devices are discussed, showing that these devices can emulate typical features of biological synapses. In the third section, nanoarchitectures based on NW networks are presented by discussing emergent synaptic plasticity effects, criticality, and avalanches effects toward the implementation of unconventional computing paradigms.

2. Fundaments of NW-based memristive devices

In this section, the synthesis of NWs as well as the fabrication of NW-based memristive devices is briefly reported by discussing different device fabrication strategies. Also, physical mechanisms of switching are discussed, with a focus on electrochemical metallization switching mechanism (ECM) and valence change switching mechanism (VCM) observed in single NWs.

2.1 NW synthesis

The synthesis methods of metal-oxide and metallic NWs can be divided in top-down and bottom-up approaches. The taxonomy of NW growth methods is reported in the schematization of Fig. 1. For a detailed description of the synthesis methods, refer to several previous review articles [21–24]. In general, top-down approaches require lithographic

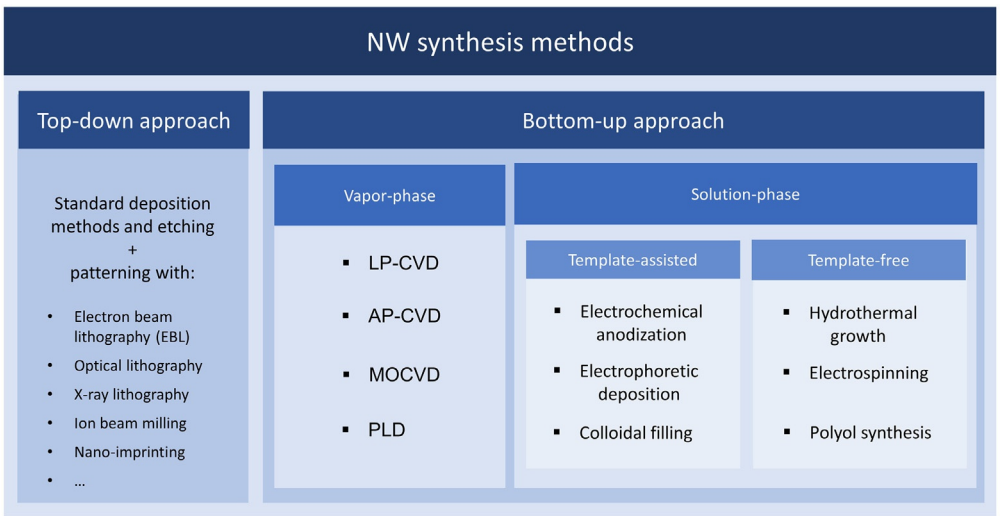


Fig. 1 Taxonomy of NW growth methods.

techniques such as electron beam lithography (EBL), optical lithography, X-ray lithography, ion beam milling or nanoimprinting and deposition of materials to define the NW geometry. Instead, bottom-up approaches are based on chemical synthesis or on the assembly of molecular building blocks. The bottom-up approach is much more interesting due to the possibility of fine control of NW dimensions, purity, and doping level at low costs. Bottom-up synthesis of NWs includes vapor-phase and solution-phase growth methods. The vapor-phase growth refers to synthesis that can be performed in vacuum chamber or in atmospheric pressure under controlled gas flow and temperature. Instead, the solution-phase synthesis, that have the advantage that can be performed usually at low temperature, can be template-assisted as in the case of electrochemical anodization, electrophoretic deposition, and colloidal filling or template-free as in the case of hydrothermal growth, electrospinning, and polyol synthesis.

2.2 Fabrication strategies of NW-based memristive devices

Memristive devices are realized by sandwiching an insulator layer (also termed as active material) in between two metal electrodes to form a metal-insulator-metal (MIM) structure as schematized in Fig. 2A [5]. In this framework, different device geometries such as the common bottom electrode structure and cross-point devices have been reported to study resistive switching devices (refer to Ref. [25]). Also, memristive devices organized in large-scale crossbar array architectures have been exploited for the implementation of efficient in-memory and brain-inspired computing paradigms directly exploiting physical laws [10]. In case of NWs, different strategies can be adopted for the realization of NW-based resistive switching devices. In case of single NW memristive devices, the single nanostructure (acting as the active material for switching) is usually contacted by means of metal electrodes in a planar structure as schematized in Fig. 2B. A different strategy consists instead in contacting a NW array, where metal electrodes contacts multiple NWs in parallel as schematized in Fig. 2B. Despite the ease of this fabrication process (usually the bottom electrode is realized by directly contacting the growth substrate), the multiple NWs in parallel make difficult to unveil the localization of the switching mechanism. As an alternative, memristive devices can be realized by considering NW networks as schematized in Fig. 2D. While single-NW-based devices represents suitable model systems for localizing and investigating physicochemical phenomena underlaying memristive functionalities as discussed in following Sections 4 and 5, NW networks can be exploited for the realization of memristive architectures for the implementation of unconventional type of computing paradigms as discussed in Section 4.

2.3 ECM mechanism

The memristive behavior based on electrochemical metallization switching mechanism rely on the electrochemical dissolution and subsequent deposition of metal atoms from an

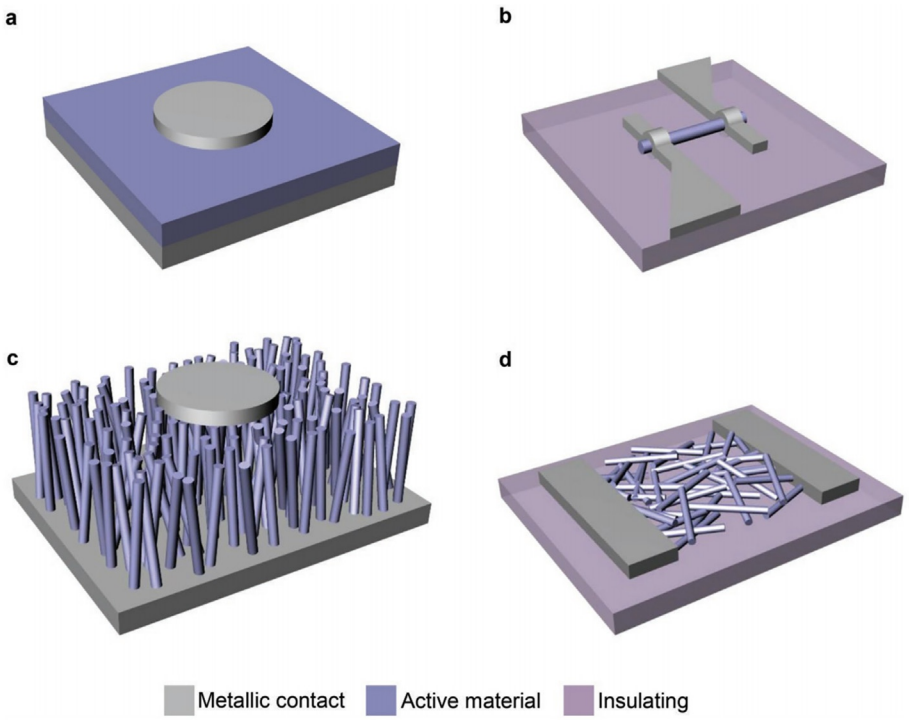


Fig. 2 Schematization of NW-based memristive device fabrication strategies. (A) Conventional memristive devices where an active material is sandwiched in between two metal electrodes. Memristive devices based on (B) Single NWs, (C) NW arrays, and (D) NW networks. *(Adapted with permission from G. Milano, S. Porro, I. Valov, C. Ricciardi, Adv. Electron. Mater. 5 (2019) 1800909. Copyright 2019, Wiley-VCH.)*

active metal electrode under the action of an applied electric field to form a conductive filament bridging the two electrodes [26,27]. In this case, the memristive cell is realized by sandwiching an active material in between an electrochemical active metal electrode (usually Ag or Cu) and an inert electrode. The ECM switching mechanism described in the following is schematized in Fig. 3A. Initially, the memristive cell is in a high resistance state (HRS) due to the insulating nature of the switching material. When a positive voltage is applied to the active electrode (represented in blue in Fig. 3A), anodic dissolution of metal atoms occurs according to the reaction:

$$M \rightarrow M^{n+} + ne^{-}$$

Then, drift of metal ions occurs across the insulating layer toward the counter electrode under the action of the applied electric field. Then, reduction and deposition of metal ions occur at the inert counter electrode according to the reaction:

$$M^{n+} + ne^{-} \rightarrow M$$

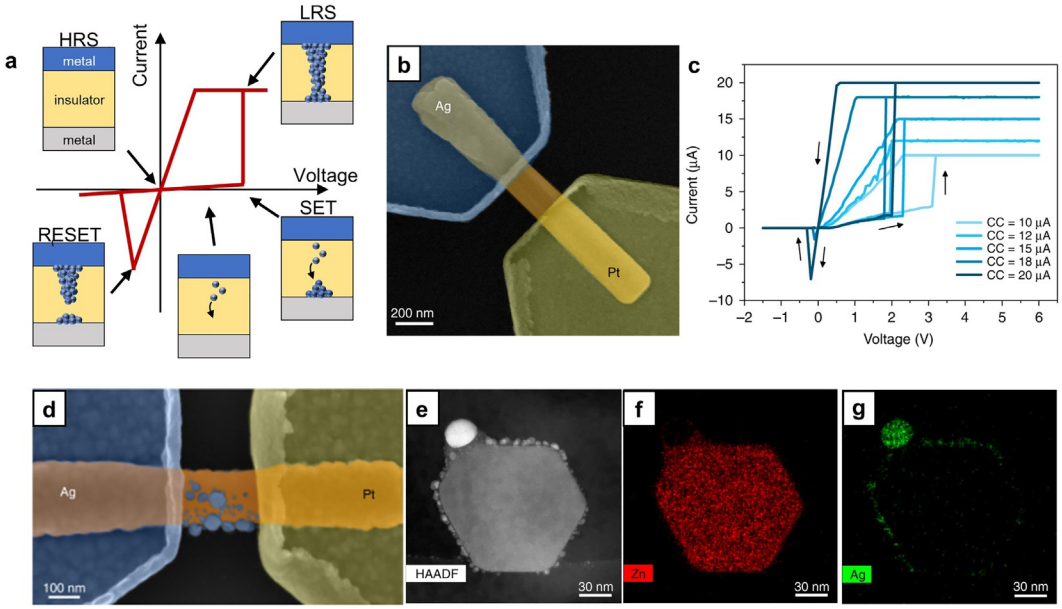


Fig. 3 Electrochemical metallization switching mechanism (ECM) in single NWs. (A) Schematic representation of the working principle of an ECM cell, where the bipolar switching mechanism that relies on the formation and rupture of a conductive filament bridging the two electrodes results in the pinched hysteresis loop in the *I-V* plane. The electrochemically active electrode metal is represented in *blue*. (B) SEM image of an ECM cell based on single NWs, where a single ZnO NW is contacted by an electrochemical active Ag electrode and an inert Pt electrode. (C) Bipolar resistive switching characteristics of Ag/ZnO NW/Pt devices with multilevel storage capability obtained by modulating the compliance current (CC) externally imposed to the device. (D) SEM image after resistive switching, showing the presence of Ag nanoclusters along the ZnO NW surface. (E) TEM image of a NW cross-section obtained by cutting the NW in the electrode spacing after resistive switching and corresponding (F) Zno and (G) Ag EDS maps. *(Panels B–G adapted from G. Milano, M. Luebben, Z. Ma, R. Dunin-Borkowski, L. Boarino, C.F. Pirri, R. Waser, C. Ricciardi, I. Valov, Nat. Commun. 9 (2018) 5151 under the terms of Creative Commons Attribution 4.0 License.)*

The electrocrystallization and deposition of these metal ions are responsible for the formation of a metallic filament bridging the two electrodes, changing the device resistance to a low resistance state (LRS). This causes a current jump in the *I-V* plane during the SET transition. Note that during SET, a current compliance is usually externally imposed to the device to prevent a full breakdown of the cell. The change of resistance is reversible, since the initial HRS can be restored when an opposite polarity is applied to the active electrode, causing a so-called RESET process. This mechanism results in a bipolar switching mechanism with the typical pinched hysteresis loop in the *I-V* plane. It is worth noticing that, before SET/RESET operations, the device usually undergoes an electroforming process to initially establish the conductive filament.

An example of an ECM cell based on a single NW is reported in Fig. 3B, where a single crystalline ZnO NWs is contacted by an Ag and a Pt electrode [28]. In the pristine

state, the electron flow in these devices is regulated by the ZnO NW resistance in combination with two back-to-back Schottky diodes created at the metal-semiconductor interface [29]. When electrical stimulated, the single-NW-based resistive switching device is characterized by an analog resistive switching behavior, where the LRS of the device can be tuned depending on the externally imposed compliance current as reported in Fig. 3C. This analog behavior can be exploited for multilevel storage. More importantly, the device represents a suitable model system for localizing the switching mechanism. Indeed, while the conductive filament is usually buried in the sandwich structure of conventional devices based on thin films, in this case, the conductive path composed of Ag nanoclusters is highly localized on the crystalline surface of the ZnO NW as can be observed in Fig. 3D–G by means of SEM and TEM analyses. The high localization of the conductive filament is here related to the higher mobility of Ag ions on the surface compared to the bulk and to the absence of spatial constrictions. Also, these devices can be exploited as model systems for the comprehension of the effect of competing redox processes (such as the effect of humidity) that influence not only the electronic conduction mechanism [30] but also memristive device functionalities [31]. Note also that resistive switching phenomena based on the migration of Ag ions was observed by considering Na, Ga, and Sb-doped single ZnO NWs [32,33]. Similarly, ECM mechanism in single-crystalline ZnO NWs was reported by considering a Cu active electrode, resulting in the formation of a Cu-based conductive filament [34–36]. The migration of Cu ions from the metal electrode was reported also in NW arrays, where ZnO NWs coated by a polyacrylic acid was reported to exhibit multiple resistance states, thanks to changes in surface states induced by redox reactions at NW surfaces [37].

2.4 VCM mechanism

The valence change mechanism (VCM) is related to the migration of oxygen-related ionic defects in metal-oxide insulator matrices [38]. The VCM switching mechanism described in the following is schematized in Fig. 4A. A positive voltage is applied to a metal electrode results in the creation of an abundance of oxygen vacancies according to the exchange reaction:

$$O_O \leftrightharpoons V_O^{\bullet\bullet} + 2e' + \frac{1}{2}O_2(g)$$

where, according to the Kröger-Vink notation, O_O are oxygen ions in regular lattice sites while $V_O^{\bullet\bullet}$ are oxygen vacancies. The oxygen vacancy defects migrate under the action of the applied electric field forming a conductive filament of reduced metal-oxide phase in correspondence of the SET process, turning the device to a low resistance state (LRS). Similarly, to ECM cells, also in this case an externally applied compliance, current is usually exploited to avoid a hard breakdown of the device during the SET operation. The initial high resistance state (HRS) can be then restored by applying an opposite polarity

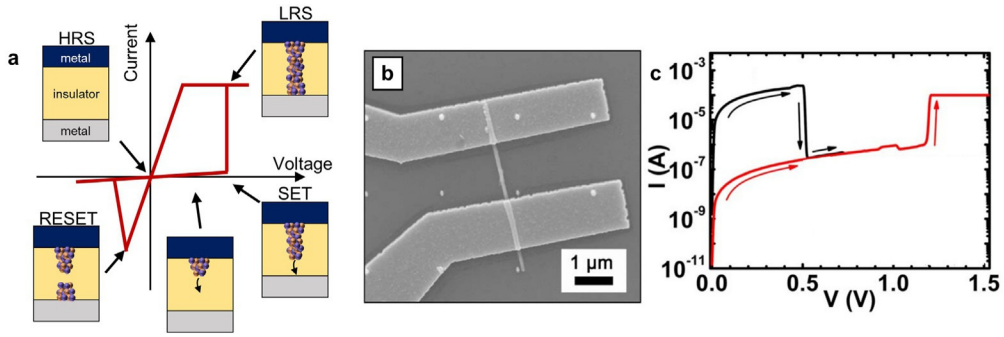


Fig. 4 Valence change switching mechanism (VCM) in single NWs. (A) Schematic representation of the working principle of an VCM cell, where the switching mechanism that relies on the migration of oxygen-related ionic defects in metal-oxide insulator matrices results in the pinched hysteresis loop in the *I-V* plane. (B) SEM image of a VCM cell based on single NWs, where a single NiO NW was contacted by Au/Ti electrodes and (C) its resistive switching behavior (SET process in *red*, RESET in *black*). *(Panels B and C adapted from S.I. Kim, J.H. Lee, Y.W. Chang, S.S. Hwang, K.-H. Yoo, Appl. Phys. Lett. 93 (2008) 033503. Copyright 2008, American Institute of Physics.)*

that induces a migration in the opposite direction of oxygen vacancies inducing a RESET process.

An example of a VCM memristive cell based on single NWs is reported in Fig. 4B, where a single NiO nanowire is contacted by means of Au/Ti electrodes [39]. In this case, the switching from the HRS to a LRS was reported to be induced by applying the same polarity to the electrodes (unipolar switching mechanism) and was suggested to be related to the NW microstructure with the presence of grain boundaries along the NW and/or Ni defects. Similarly, to ECM cells, NWs have been reported to be suitable model systems for the investigation of the VCM switching mechanism. Indeed, the high localization of the switching events allows direct in situ analysis of morphological changes and local chemical composition during resistive switching [40,41]. In this framework, the peculiar planar structure of VCM single NW cells have been exploited for investigating the intrinsic mechanism of switching including (i) the influence of chemical reaction with surroundings by means of atmosphere controlled measurements, (ii) the identification of the carrier type of the conductive filament thanks to field effect measurements and (iii) the analysis of switching location through multiprobe measurements [42,43]. Also, note that a VCM type of resistive switching was observed also in case of memristive devices based on NW arrays, where in case of ZnO NWs the switching was reported to be located in a ZnO thin layer in between the growth substrate acting as a bottom electrode and vertically aligned NWs [44,45]. It is worth noticing also that, by taking into advantage of the high surface-to-volume ratio of NWs, the switching mechanism of both single NWs and NW arrays has been reported to be modified and/or enhanced by specific surface treatments/functionalization [46–49] and by light stimulation [50,51].

3. Single nanowire memristor as artificial synapse

The physical mechanism of switching described in previous Sections 2.3 and 2.4 can be exploited for the realization of artificial synapses for neuromorphic systems by taking into advantage of the analog behavior of memristive devices [52–55]. The synaptic functionalities shown to be emulated in memristive devices include potentiation and depression, long-term and short-term plasticity (STP and LTP), spike rate-dependent plasticity (SRDP), and spike time-dependent plasticity (STDP) [55].

Thanks to the possibility of programming the device in a wide range of resistance states in an analog fashion, single TiO_x NW devices have been exploited for the emulation of synaptic plasticity processes such as STDP and Hebb's rule [56]. In this case, the synaptic weight (i.e., the capacity of a synapse to transmit a signal) is emulated by the single NW device conductance that was modulated by means of voltage pulses applied to presynaptic and postsynaptic metal electrodes. Similar devices have been shown to be programmable with heterogeneous stimuli since the conductance can be modulated not only by electrical stimuli but also through optical pulses, allowing the implementation of associative memory effects as reported by O'Kelly et al. [57].

By exploiting ECM type of switching, single ZnO NWs have been considered for mimicking STP effects typical of biological synapses [28,58]. In biological synapses, STP effects are regulated by Ca^{2+} dynamics that can be emulated by exploiting Ag^{+} dynamics on the crystalline NW surface, as depicted in Fig. 5A and B. In brief, when electrical stimuli reach the presynaptic terminal of a biological synapse lead the opening of voltage-dependent Ca^{2+} channels leading to an influx of Ca^{2+} ions inside the neuron terminal. This causes an enhanced synaptic transmission as a consequence of the release of neurotransmitters in the synaptic cleft. Subsequently, after the end of the stimulus, the basal intracellular concentration is restored by extrusion of Ca^{2+} ions. In the single NW artificial synapse, an electrical stimulation leads to the migration of Ag^{+} ions along the NW surface that, under proper stimulation conditions, is responsible for a gradual increase of the device conductivity (synaptic weight) as a consequence of the progressive formation, rearrangement, and reinforcement of the conductive filament bridging the electrodes. The extrusion process after the end of stimulation is instead emulated by the spontaneous dissolution over time of the previously formed Ag conductive filament due to interfacial energy minimization effects and electromotive forces. The gradual enhancement of the NW-based device conductance under stimulation and subsequent spontaneous relaxation to the ground state is reported in Fig. 5C, where the time constant of relaxation is observed to be ~23 ms. Ionic dynamics in these devices can be exploited for the emulation of Paired Pulse Facilitation (PPF), a feature of STP in biological synapses related to the progressive enhancement of the synaptic transmission under temporally correlated stimuli. The emulation of PPF in single NWs is reported in Fig. 5D, where the device conductance (current) can be observed to gradually increase over pulses. Due to the interplay between conductive path formation and spontaneous relaxation, the percentage of the conductance change (Δw) can be increased by reducing the

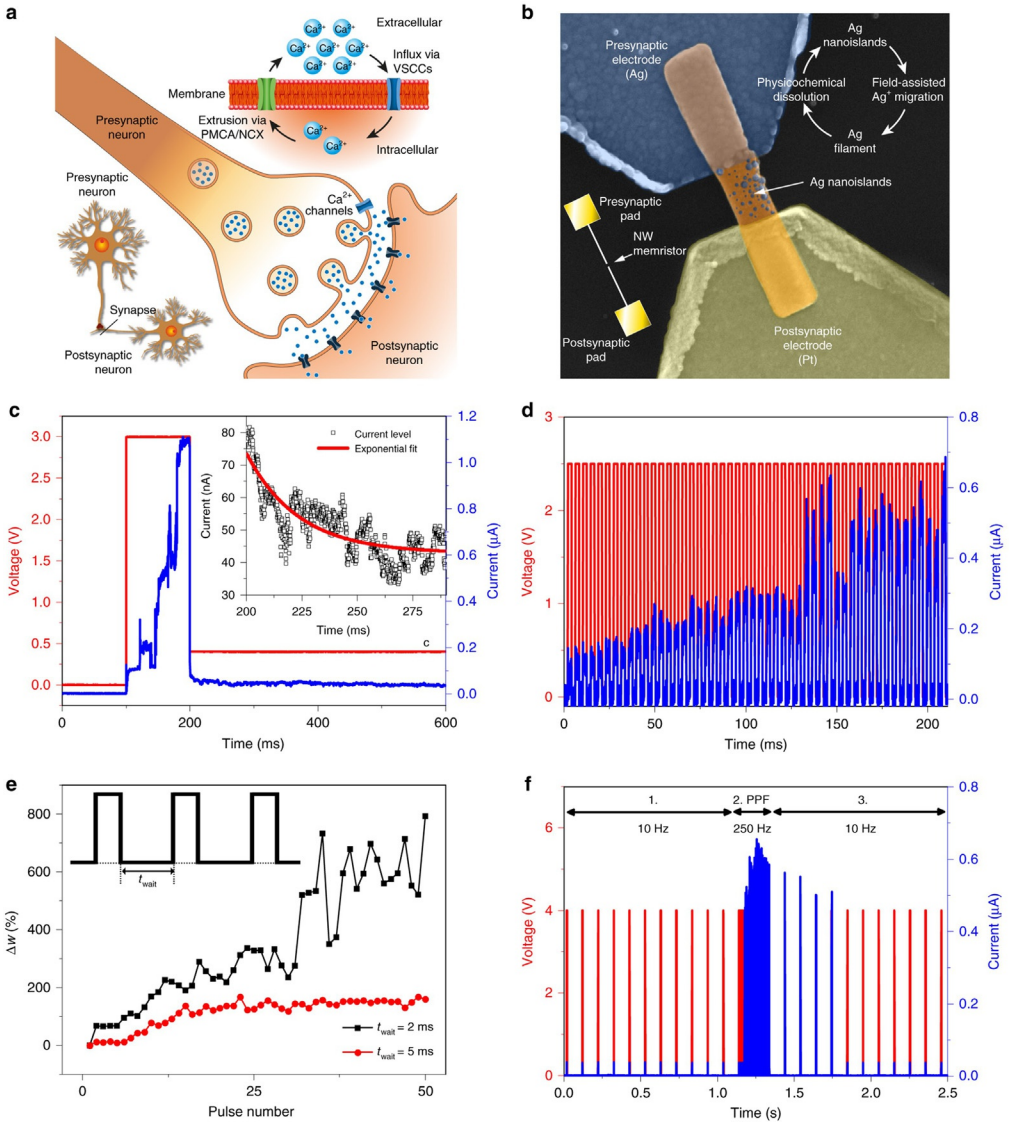


Fig. 5 Single NW-based memristive device as artificial synapse. Analogy between (A) a biological synapse and (B) an artificial synapse by exploiting similarities in between Ca^{2+} dynamics and Ag^{+} dynamics. (C) Short-term plasticity effect in a single ZnO NW device, where electrical stimulation results in a gradual increase of device conductivity (current), while a spontaneous relaxation to the ground state occurs after stimulation (detail of relaxation as inset). (D) PPF in a single ZnO NW memristive device and (E) the dependence on the change of conductivity (Δw) on the time in between pulses. (F) Dependence of PPF on the pulse frequency. PPF occurs when the device is stimulated with temporally correlated stimuli (high-pulse frequencies), with subsequent spontaneous relaxation to the ground state when the stimulation frequency is reduced. *(All panels adapted from G. Milano, M. Luebben, Z. Ma, R. Dunin-Borkowski, L. Boarino, C. F. Pirri, R. Waser, C. Ricciardi, I. Valov, Nat. Commun. 9 (2018) 5151 under the terms of Creative Commons Attribution 4.0 License.)*

time in between pulses (t_{wait}), as reported in Fig. 5E. In this framework, note that only highly temporally correlated pulses (high-pulse frequencies) lead to PPF, since the stimulation with low-pulse frequencies results in negligible variation of device conductivity as reported in Fig. 5F. Here, the device spontaneously relaxes back to the ground state when the pulse frequency is reduced after inducing PPF with high-frequency voltage pulses.

4. Nanowire random networks as artificial neural networks

Memristive devices realized with a top-down approach and organized in large crossbar arrays have been exploited for the realization of artificial neural networks (ANNs) able to implement supervised and unsupervised learning paradigms [10]. However, the topology and adaptability of these grid-like crossbar architectures, where implementation of learning schemes requires the addressing of single network elements, strongly differ from biological neural networks. Indeed, in biological neural networks, an emergent behavior arises from complexity and is intrinsically related to the interplay between structure and function, both governed by the principle of self-organization [59]. Indeed, human cognitive functions that allow us to receive, store, transform, and process information we receive from external stimuli rely on the human brain complex network where an emergent behavior arises from the interaction of 10^{11} neurons connected by means of 10^{14} synapses [59]. In this scenario, biologically plausible artificial neural networks based on self-organized NW networks have been proposed as a platforms for neuromorphic type of data processing and bio-inspired computing.

The realization of ANNs based on self-organized NWs is in agreement with the concept of "nanoarchitectonics" as a new paradigm of materials science and technology at the nanoscale, as conceived by M. Aono and K. Ariga where [60]:

i. nanomaterials and nanosystems are realized by organizing nanoscale structures (nanoparts) even with some unavoidable unreliability;
ii. main players are not the individual nanoparts but their interactions, which cause a new functionality to emerge; and
iii. unexpected emergent functionalities can result from assembling or organizing a huge number of nanoparts.

In the following, the realization of neuromorphic architectures based on self-organizing NWs is discussed by analyzing the NW network topology and the emergent behavior arising from complexity, discussing how these nanoarchitectures can be exploited for the emulation of synaptic plasticity effects typical of biological neuronal circuits and for the implementation of unconventional computing paradigms.

4.1 Nanowire network topology

With the aim of emulating biological neuronal circuits (Fig. 6A) where the principle of self-organization governs both structure and functions, metallic NW networks have been

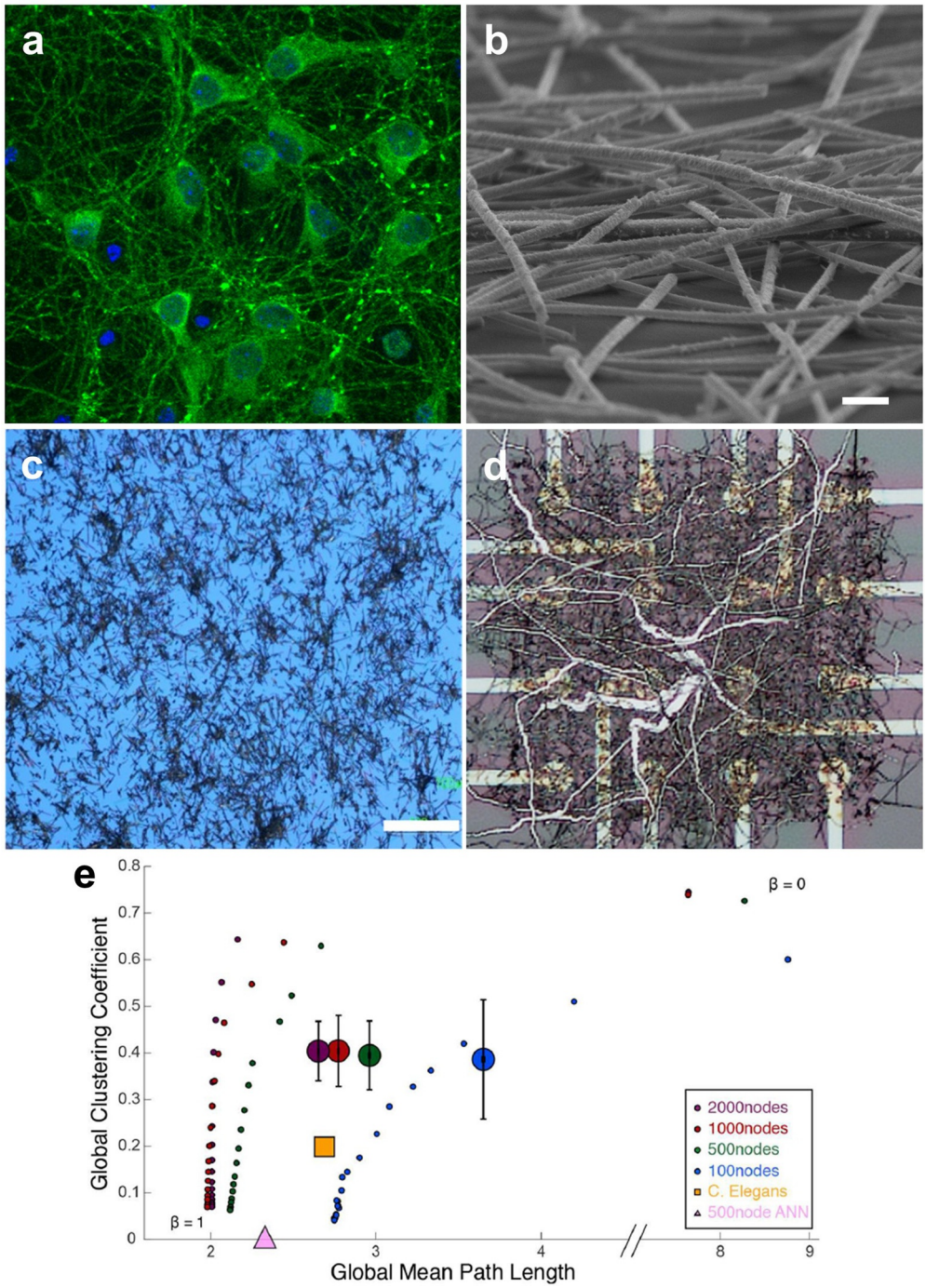


Fig. 6 See legend on next page

(Continued)

created by exploiting self-assembling techniques. NW random networks can be fabricated by means of a drop-casting techniques through the dispersion of metallic core-insulating shell NWs in solution on a substrate [61,62], obtaining highly interconnected NW networks of metal-insulator-metal junctions as shown in Fig. 6B and C. In this case, the NW density of the network can be controlled by appropriately tuning the NW concentration in the drop-casted solution. Importantly, this represents a simple and low-cost fabrication technique that does not require cleanroom facilities and nanolithography. Alternatively, NW networks can be fabricated by combination of top-down standard CMOS photolithography with bottom-up self-assembly through electroless deposition to realize complex networks as shown in Fig. 1D [63–65]. In this case, self-assembly of Ag NW structures proceeds from copper seeds that have to be priorly patterned. In this case, a subsequent sulfurization process with a carrier gas have to be exploited for the realization of Ag-Ag_2S-Ag metal-insulator-metal junctions. Acting as a complex network of atomic switches, this kind of network is usually referred as "Atomic Switch Networks."

The NW network can be considered as a complex system where the topology can be analyzed through the graph theory approach. In their work, A. Loeffer et al. [66] simulated the self-assembling of NWs by randomly placing 1D objects within a 2D plane. The system connectivity was mapped onto a graph adjacent matrix where nodes correspond to NWs, while edges corresponds to NW junctions. Then, they applied cartographic approaches to compare neuromorphic NW network structural topology with random Watt-Strogatz (WS) networks, grid-like Watt-Strogatz networks, and

Fig. 6, cont'd Nanowire network topology. (A) Biological neural network (primary mouse hippocampal neurons) where a multitude of synaptic contacts (bright fluorescent boutons at the contact points) provide high connectivity in between neurons and (B) SEM image of a biologically inspired NW network realized by drop-casting Ag NWs on an insulating substrate (scale bar, 500 nm). (C) Optical image of a nanowire networks after drop-cast deposition (scale bar, 100 μm). (D) Atomic switch network of highly interconnected Ag nanowires realized by self-assembling through electroless deposition. (E) Watts and Strogatz cartographic plane showing the relation between the global clustering coefficient and global mean path length of NW networks. *Large dots* represents 39 NW networks of different parameters, where *colors* represents the network size (number of NWs). *Small dots* are Watts-Strogatz networks, rewired from completely random ($\beta = 1$) to grid-like ($\beta = 0$) where β values decrease from *bottom* to *top*. The *large yellow square* represents the *C. elegans* network, while the *large pink triangle* represents a 500-node fully connected ANN. *((A) and (B) Adapted from G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, C. Ricciardi, Adv. Intell. Syst. 2 (2020) 2000096 under the terms of Creative Commons Attribution 4.0 License. (C) Adapted from A. Diaz-Alvarez, R. Higuchi, P. Sanz-Leon, I. Marcus, Y. Shingaya, A.Z. Stieg, J.K. Gimzewski, Z. Kuncic, T. Nakayama, Sci. Rep. 9 (2019) 14920 under the terms of Commons Attribution 4.0 License. (D) Adapted from H.O. Sillin, R. Aguilera, H.-H. Shieh, A.V. Avizienis, M. Aono, A.Z. Stieg, J.K. Gimzewski, Nanotechnology 24 (2013) 384004, Copyright 2013, IOP Publishing. (E) Adapted from A. Loeffler, R. Zhu, J. Hochstetter, M. Li, K. Fu, A. Diaz-Alvarez, T. Nakayama, J.M. Shine, Z. Kuncic, Front. Neurosci. 14 (2020) https://doi.org/10.3389/fnins.2020.00184 under the terms of Creative Commons Attribution 4.0 License.)*

the *C. elegans* biological system. This comparison was performed by considering graph metrics such as the mean path length and the clustering coefficient, as reported in Fig. 6E. While the path length measures the average number of edges along the shortest paths for all possible pairs of network nodes, the clustering coefficient measures how much nodes in a graph tend to cluster together. NW networks are characterized by higher clustering and longer mean path length than WS random networks while they are less clustered and characterized by shorter path lengths than grid-like WS networks. Large NW networks have similar mean path length respect to the *C. elegans* but higher clustering. More importantly, NW networks possess relatively low path length and high clustering, falling in the "small-world" category similarly to biological neural networks. In this case, the majority of edges are involved in forming small and densely connected clusters whereas the remaining edges are involved in maintaining connections between these clusters. Empirical evidences support the argument that the "small world" architecture contributes to the robustness and efficiency of information transfer in complex networks, including our brain [67,68].

4.2 Memristive behavior of single network elements

The neuromorphic emergent behavior of the NW networks relies on the memristive mechanisms occurring in NW network elements. Indeed, the change of conductance in single memristive elements is responsible for redistribution of current/voltage across the network giving rise to collective phenomena. In this framework, the neuromorphic emergent behavior of the network has been reported to be related to resistive switching events occurring in single NW junctions ("reweighting" effect) and single NWs ("rewiring" effect), as discussed in the following.

4.2.1 "Reweighting" effect in single NW junctions

The reweighting effect relies on the change of conductivity of NW junctions when a voltage difference is present across intersecting NWs. This mechanism can be investigated by considering single NW junction devices as reported in Fig. 7A [61]. Depending on the NW core and shell materials, this memristive behavior at the nanowire intersection can be ascribed to VCM or ECM resistive switching mechanisms. Cross-point junctions realized with intersecting NWs with an electrochemically inert metal core and surrounded by a metal-oxide layer exhibit a resistive switching mechanism that is related to the migration of oxygen-related species in the metal-oxide insulating layer to form a substochiometric conductive pathway connecting the metallic cores according to the VCM mechanism. This type of switching mechanism related to the formation/rupture of an oxygen-related conductive filament was reported in case of cross-point junctions of Ni-NiO core-shell NWs [69–71]. In case of NWs with an electrochemically active core, the switching mechanism occurs according to the ECM mechanism. In this case, a voltage difference in between metallic cores of intersecting NWs is responsible for the

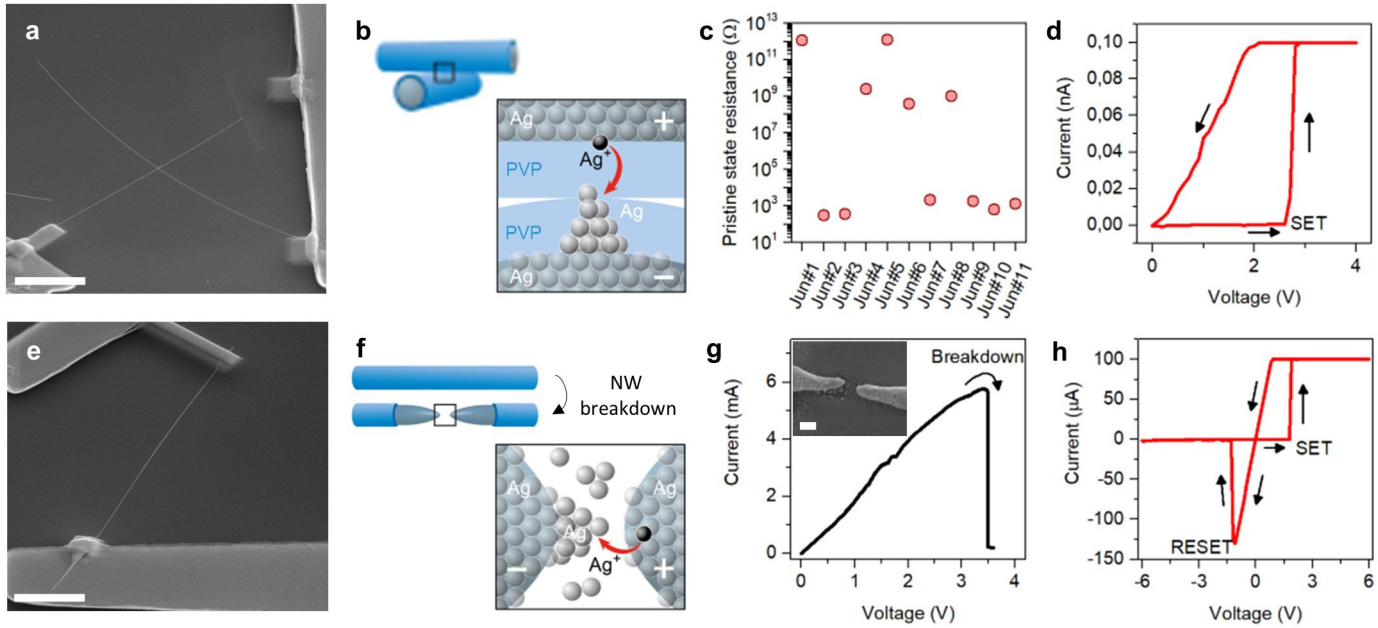


Fig. 7 Memristive behavior of NW network elements. (A) Single Ag NW cross-point junction device (SEM image, scale bar of 10 μm) and (B) schematic representation of the memristive mechanism of reweighting at the intersection in between NWs. (C) Pristine state of different single NW junction devices where the mechanical stochasticity of the contact is responsible for the high resistance variability and (D) resistive switching behavior under voltage sweep stimulation. (E) Single NW device (SEM image, scale bar of 10 μm) and (F) schematic representation of the memristive mechanism of rewiring in the NW gap after breakdown. (G) Breakdown of a single NW device showing a drop in current due to the creation of a nanogap (SEM image of a created nanogap as inset, scale 100 nm) and (H) resistive switching behavior of the breakdown-induced nanogap in a single NW. *(All panels adapted from G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, C. Ricciardi, Adv. Intell. Syst. 2 (2020) 2000096 under the terms of Creative Commons Attribution 4.0 License.)*

dissolution of the metallic core to form metal ions that starts to migrate across the insulating shell layer at the NW junction to form a metallic conductive filament bridging the cores of intersecting NWs, as schematized in Fig. 7B in case of Ag NWs surrounded by a PVP shell layer. The formation/rupture of this metallic filament under electrical stimulation is responsible for the memristive behavior of these cross-point junctions. An important aspect is that experimental results on Ag NWs revealed that cross-point junction devices exhibit a wide range of pristine state resistances as a consequence of the mechanical stochasticity of the contact in between NWs, as reported in Fig. 7C [61]. However, these cross-point devices show resistive switching characteristics as reported in Fig. 7D [61]. By considering similar Ag NW junctions, Manning et al. [72] showed that the conductivity of NW junctions can be programmed to different conductance states depending on the imposed compliance current, showing the analog behavior of these cross-point devices. It is also important to remark that the symmetric structure of the junction results in the possibility of forming a conductive filament by applying both voltage polarities to inner cores of intersecting NWs [61].

4.2.2 "Rewiring" effects in single NWs

The rewiring effect rely on the rupture and rewiring that can occur in single NWs composing the network under electrical stimulation. This mechanism can be observed by considering single NW devices as reported in Fig. 7E [61]. The electrically induced rupture of NWs is related to Joule heating and electromigration-driven electrical breakdown events that are responsible for the creation of a needle-like nanogap. Then, rewiring can occur by regenerating the electrical connection through electromigration phenomena involving the formation of a conductive pathways across the nanogap. The rupture/rewiring phenomena is schematized in Fig. 7F in case of Ag NWs surrounded by a PVP shell layer. The corresponding electrical breakdown characteristic is reported in Fig. 7G where it is possible to observe a sudden drop of current related to the creation of a nanogap across a single Ag NW. After the nanogap formation, resistive switching behavior related to electrically induced rupture/rewiring of the nanogap can be observed as reported in Fig. 7H [61].

4.3 Emergent dynamics

Reweighting and rewiring effects related to resistive switching occurring in single NW junctions and single NWs at the nanoscale and the mutual interaction in between these memristive elements give rise to an emergent behavior of the network at the mascroscale. Indeed, when a voltage difference is applied to any two connected areas of the network, the current is distributed according to the Kirchhoff's laws and is regulated by the conductivity of each memristive element of the network. Changes in the conductivity of memristive elements upon electrical stimulation are responsible for a redistribution of current and voltage across other network elements with a consequent cascade of

conductance changes of connected memristive elements. These cascade effects are responsible for the emergence of a nonlinear memristive response of the network, as shown in Fig. 8A that reports the memristive hysteretic loop in the *I-V* plane under electrical stimulation in between two areas of an Ag NW network [61].

The emergent memristive dynamics of the networks are related to an avalanche of conductance changes in network elements that are responsible of spatially correlated structures of network activity. A direct experimental visualization of the emergence of a conductive pathway in Ag NW networks under electrical stimulation was performed by Manning et al. [72] by means of passive voltage contrast. By using this technique, NWs that are participating to the conductive pathway appear to be darker than NWs that are less relevant for the network conductivity. As can be observed from Fig. 8, unperturbed NWs (not electrically stimulated) do not show significant contrast in between NWs. However, by stimulating the network with a current compliance of 50 nA, it is possible to observe the emergence of a conductive pathway composed of darker NWs that bridge the two electrodes. This conductive pathway results to be reinforced by increasing the flowing current. As discussed by Manning et al. [72], electrical stimulation of random networks is responsible for the establishment of "winner-takes-all" conductive pathways that represent the lowest possible energy connectivity pathway across the network. In this case, the memory state is encoded in specific connectivity pathways similarly to biological systems. It is important to remark that in the limit of sufficiently low input currents (before the establishment of the "winner-takes-all" regime), O'Callaghan et al. [73] reported that the system behaves as a leakage capacitive network where NW junctions behave like capacitors.

Direct visualization of the formation of a conductive pathway by means of the lock-in thermography technique was reported also by Q. Li et al. [74] in case of Ag NWs decorated with TiO_2 nanoparticles. In this case, the square-wave-modulated input voltage used to stimulate the network was also used as a reference for synchronization with the IR camera to provide lock-in operation of the system. Fig. 8C reports the optical image of the device (topography), the corresponding lock-in amplitude, and the phase images that were acquired simultaneously. As can be observed from the lock-in amplitude image, a trace can be observed in between stimulating electrodes as a consequence of the IR radiation of NWs and NW junctions due to Joule heating effects that occur mainly in correspondence of the main current pathway. The phase image allows to distinguish in between different processes contributing to the thermal image. Indeed, radiation emission related to direct power dissipation due to Joule heating is responsible for a radiation that is in phase with the current, while a delay in the emission is expected in case of diffusive processes. As can be observed, a nearly zero phase region can be observed in correspondence of the main trace observed by lock-in amplitude, meaning that radiation is here emitted locally and directly within the current transmission pathway.

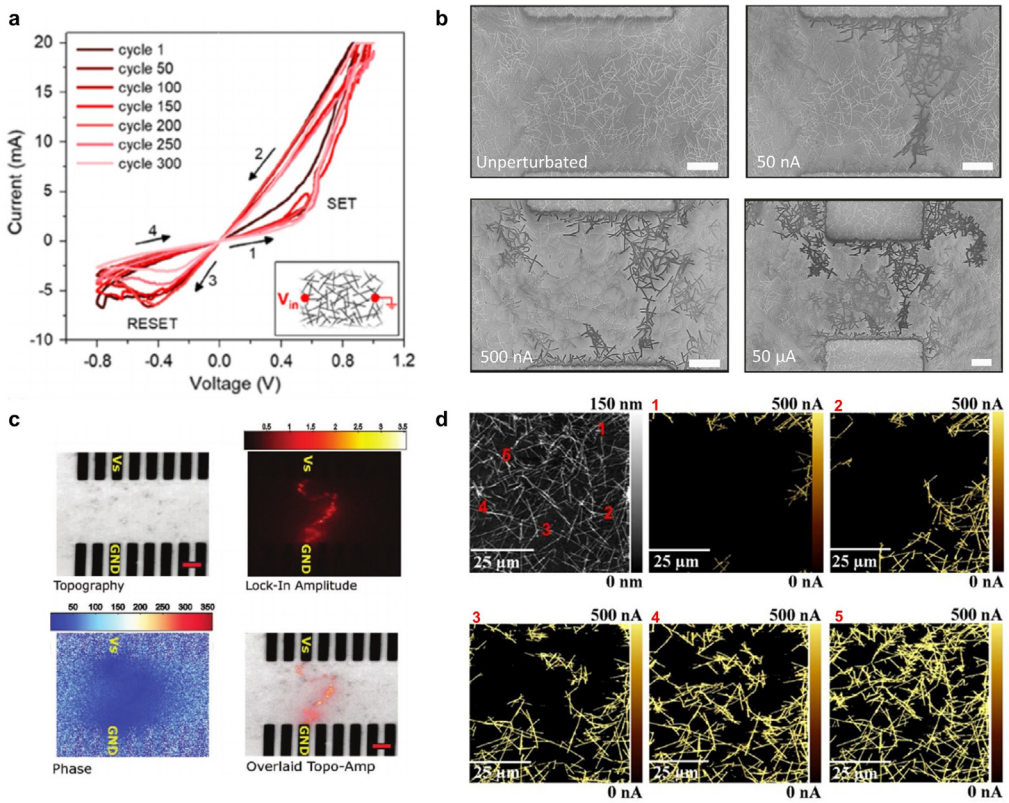


Fig. 8 Emergent dynamics of NW networks. (A) Emergent memristive behavior of an Ag NW network in two-terminal configuration (as schematized in the inset) under voltage sweep stimulation. (B) Direct visualization of the conductive pathway across an Ag NW network by passive voltage contrast SEM images. Images were acquired on the same network that was stimulated by means of *I-V* sweeps with different limiting current compliances in between the two metal contacts. (C) Direct visualization of the conductive pathway during electrical stimulation by means of lock-in thermography in Ag-TiO_2 core-shell NW networks. (D) Visualization of conductive pathways and local stimulation of an Ag NW network through C-AFM. Current maps are the results of applying voltage pulses in between selected regions marked 1–5 in the topographic map and an electrode located at the top of the image. *((A) Adapted from G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, C. Ricciardi, Adv. Intell. Syst. 2 (2020) 2000096 under the terms of Creative Commons Attribution 4.0 License; (B) Adapted from H.G. Manning, F. Niosi, C.G. da Rocha, A.T. Bellew, C. O'Callaghan, S. Biswas, P.F. Flowers, B.J. Wiley, J.D. Holmes, M.S. Ferreira, J.J. Boland, Nat. Commun. 9 (2018) 3219 under the terms of Creative Commons Attribution 4.0 License; (C) Adapted with permission from Q. Li, A. Diaz-Alvarez, R. Iguchi, J. Hochstetter, A. Loeffler, R. Zhu, Y. Shingaya, Z. Kuncic, K. Uchida, T. Nakayama, Adv. Funct. Mater. 30 (2020) 2003679. Copyright 2020, Wiley-VCH; (D) Adapted with permission from P.N. Nirmalraj, A.T. Bellew, A.P. Bell, J.A. Fairfield, E.K. McCarthy, C. O'Kelly, L.F.C. Pereira, S. Sorel, D. Morosan, J.N. Coleman, M.S. Ferreira, J.J. Boland, Nano Lett. 12 (2012) 5966. Copyright 2012, American Chemical Society.)*

The emergent behavior of the network can be explored also by manipulating the connectivity at the nanoscale by means of conducting atomic force microscopy (C-AFM) as reported by Nirmalraj et al. [75]. For this purpose, authors employed a metal-coated AFM probe as a mobile nanoscale electrical contact to acquire simultaneously topographic and conductance maps, with the possibility also to locally stimulate the NW network. As reported in Fig. 8D, it is possible to observe changes in the network conductance map after electrical stimulation of five different selected regions indicated in the topographic map as a consequence of the local activation of conductive pathways after stimulation. Additional experimental techniques that can be explored for spatial mapping variations of conductivity under electrical stimulation include electrical resistance tomography (ERT) [76,77].

It is worth noticing also that the dynamics of these networks can be theoretically explored through information-theoretic metrics such as the transfer entropy (TE) and the active information storage (AIS) that allow to investigate the dynamical information flow across the network abstracted as a graph [78].

4.4 Synaptic plasticity in NW networks

The emergent adaptive behavior of memristive nanowire networks under external stimulation has been exploited for the emulation of plasticity effects typical of biological systems including short-term and long-term synaptic plasticity, structural plasticity, and homosynaptic/heterosynaptic plasticity as discussed in the following.

4.4.1 Short-term and long-term synaptic plasticity

Short-term synaptic plasticity (STP) has been reported to be emulated in Ag NW networks by exploiting the volatile behavior of the conductive path due to the spontaneous dissolution of Ag conductive filaments previously formed in memristive elements of the network [61,62]. In this framework, the short-term dynamics were observed by applying a voltage pulse stimulation to the network and subsequently recording the time trace of network conductance over time. As can be observed from Fig. 9A, the stimulation with a voltage pulse in between two pads (that emulate presynaptic and postsynaptic neuron terminals) is responsible for a gradual increase of the network conductance (facilitation). After stimulation, the synaptic weight gradually relaxes back to the initial ground state exhibiting a volatile behavior. These short-term plasticity dynamics can be controlled by modulating the stimulation amplitude, since a higher voltage pulse amplitude is responsible for a more intense network potentiation and longer relaxation over time. These short-term plasticity dynamics related to the formation and progressive dissolution of a conductive pathway in between two network electrodes can be exploited for the emulation of synaptic functionalities such as paired-pulse facilitation (PPF). As reported in Fig. 8B, network stimulation by means of voltage pulses applied within short time intervals leads to a gradual enhancement of the network conductivity as a function of the number of pulses while, after stimulation, the network spontaneously relaxes back

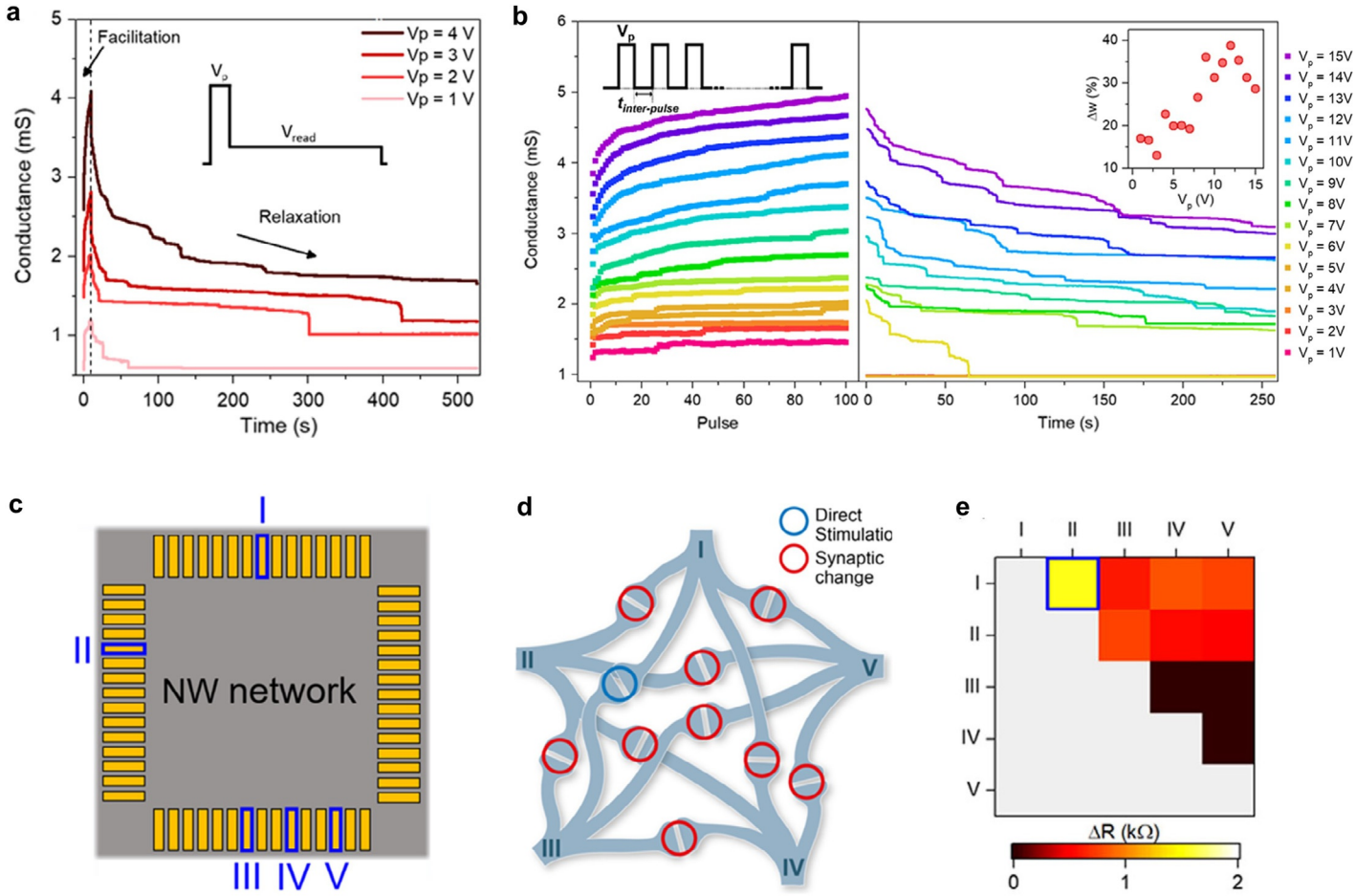


Fig. 9 Synaptic plasticity effects in NW networks. (A) Short-term synaptic plasticity effect of Ag NW networks measured in two-terminal configuration showing a gradual increase in conductance (synaptic facilitation) after voltage pulse stimulation and subsequent conductance relaxation. A higher facilitation was observed by stimulating the network with voltage pulses with higher amplitude. (B) Paired-pulse facilitation (PPF) in Ag NW networks *(left panel)* and subsequent spontaneous relaxation *(right panel)*. The inset shows the change in the synaptic weight (Δw) after 100 pulses as a function of the applied pulse amplitude. (C) Multiterminal configuration of an Ag NW network where electrodes represent neuron terminals, (D) corresponding biological representation of the device where synaptic interactions lead to heterosynaptic plasticity effects with synaptic changes also in nondirectly stimulated synapses and (E) correlation maps of resistance variation (ΔR) in synaptic pathways after stimulation of synaptic pathway I–II (highlighted in *blue*) experimentally showing heterosynaptic plasticity effects. The resistance variation was calculated as $\Delta R = -(R_{post_stimulus} - R_{pre_stimulus})$, thus the variation of ΔR has to be intended as a decrease of the resistance of the synaptic pathway after stimulation. *(All panels adapted from G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, C. Ricciardi, Adv. Intell. Syst. 2 (2020) 2000096 under the terms of Creative Commons Attribution 4.0 License.)*

to its initial ground state. The synaptic weight change (Δw) after PPF and subsequent relaxation time can be modulated by properly tuning the voltage pulse amplitude. Also, it was demonstrated that networks can preserve information from previously stimulated pathways through increased local junction connectivity, reshaping and altering the formation of other conductive pathways when the network is subsequently stimulated in different locations [74]. This leads to an interaction of short-term and long-lasting effects (long-term synaptic plasticity) [74].

4.4.2 Structural plasticity

In addition to weight plasticity that rely on the weight change of connections in between already existing connections of the neural network, the connectivity of the system can be altered by means of the formation and elimination of new connections. This form of plasticity that includes synapse formation/elimination is referred as wiring or structural plasticity, a plasticity effect that can affect encoding of learned information in biological systems [54,79]. This synaptic behavior can be emulated in NW networks by exploiting the rewiring effect occurring in single NWs as previously discussed. While reweighting effects occurring in NW junctions are responsible for a change of the synaptic connectivity of network junction, the rewiring effect can be exploited for the elimination/formation of new connections thus emulating wiring plasticity as demonstrated in case of Ag NWs [61]. It is important to remark that this type of plasticity it is hardly implementable in conventional memristive architectures realized with a top-down approach that is characterized by fixed connections.

4.4.3 Homosynaptic and heterosynaptic plasticity

While homosynaptic plasticity is input-specific and depends on the direct stimulation of neuron terminals of a specific synapse, when a direct stimulation of a synapse leads to the variation of the synaptic strength also in other nondirectly stimulated synapses, it is referred as heterosynaptic plasticity. In biological systems, this form of plasticity has been reported to contribute to the homeostasis and stability of the network [80]. By exploiting the functional connectivity of the system, heterosynaptic plasticity was demonstrated in NW networks in multiterminal configuration [61] as depicted in Fig. 9C, where each pad represents a neuron terminal. The corresponding biological representation of the system is reported in Fig. 9D, where the connection in between any pair of neuron terminal pads represents a synaptic pathway. In this configuration, the intrinsic heterosynaptic plasticity is related to the collective response of the network to external stimulation. As an example, Fig. 9E shows the correlation map of resistance variations (ΔR) (that represents the change of the synaptic connection) across the network after direct stimulation of neuron terminal pads I and II (synapse I–II) of the network reported in Fig. 9A. As can be observed, the direct potentiation of synapse I–II results not only in a variation of the synaptic strength of synapse I–II but also of other nonstimulated synapses. More in details,

larger changes were observed in synaptic pathways directly connected to directly stimulated terminals I and II (synapse I–III, I–IV, I–V, II–III, II–IV, II–V) while the strength of spatially distant synaptic pathways (synapse III–IV, III–V, IV–V) remained almost unaltered. This shows that the resistance of synaptic pathways results to be poorly influenced by peripheral changes in the network connectivity, showing a dependence of heterosynaptic behavior on the spatial location of the primary plasticity effect.

4.5 Criticality and avalanches effects

The human brain complex network has been proposed to operate at a self-organized critical point to optimize the information transfer, information processing, and memory [81,82]. At the critical point, complex networks of coupled nonlinear elements are characterized by scale-invariant avalanches effects where each event is likely to trigger subsequent events as reported in a wide range of physical systems from magnetization to earthquakes [83]. Near criticality, complex networks embracing complexity in the network dynamics have been argued to maximize their computational performance [84].

Emergent criticality was reported by A. Stieg et al. [85] in Ag-NW-based atomic switch networks by observing the current response of the complex nanonetwork under voltage pulse stimulation. In particular, the current response was observed to fluctuate through a wide range of metastable conductance states that were associated to distinct and discrete network configurations. These discrete configurations were classified by considering the residence time of given resistance states (where considered metastable all the conductance states with persistence time that exceeds the measurement bandwidth). The observed fluctuations of conductivity under stimulation can be attributed to internal network dynamics. The probability of state duration and its likelihood have been reported to be related by a power law distribution. Together with the observation of a spatially distributed network activity, the temporal correlation of metastable states with power law scaling for probability has been reported as fingerprints of critical dynamics of the system. Pike et al. [86], in case of self-organized nanoparticle networks, reported that criticality emerges in a parameter range where the network self-tunes to a state with an optimal number of pathways through the network. In this framework, Mallison et al. [87] reported that, in nanoparticle networks, sizes and durations of avalanches of switching events are power law distributed with power law exponents in accordance with criticality when fabricated at a percolating phase transition. All these works suggest that signals in self-organized networks, if appropriately fabricated, are qualitatively and quantitatively in accordance with signals measured in the cortex. In view of the implementation of computing paradigms, it is worth noticing that Hochstetter et al. [88] reported in case of NW networks that NW junctions switch collectively as avalanches and networks can be tuned between ordered and chaotic dynamical regimes, where the edge-of-chaos regime was observed to optimize information processing.

5. Computing with nanowire random networks

Traditional computing architectures based on the von Neumann paradigm are at the base of modern processors, where the computing capability has been largely increased during last years by developing architectures with multiple cores and large memory caches. Also, computing efficiency have been increased by reducing the memory access times and by parallelizing computations. However, advances in computing capabilities continually require device scaling, where the reduction in size is also related to the improvement of costs and power consumption. In this framework, a huge amount of resources have been invested in sustaining the scaling trend of transistors to sustain the ever-growing demand of electronics and computing capabilities. However, increasing costs and inherent fundamental physical limits represents nowadays a challenge for further scaling of the transistor technology, making difficult to fulfill the Moore's prediction of doubling the number of transistor every 2 years [89]. In this context, new computing paradigms and new electronic devices are demanded to fulfill the ever-growing computing demand.

Among emerging computing paradigms, bioinspired computing aims to realize intelligent systems able to emulate the efficiency and efficacy of the human brain. In this scenario, nanowire-based nanoarchitectures have attracted growing attention as a platform for the implementation unconventional computing paradigms such as reservoir computing. Differently from conventional computing architecture that suffers the so-called "von-Neumann" bottleneck associated to the high costs of data movement between the memory and the processing unit (Fig. 10A), complex nanowire networks have been proposed as promising platforms for simultaneous processing of multiple inputs where the output of the system can be then analyzed by a readout function (Fig. 10B). The concept of computing in a classical computing architecture and in a complex network reservoir are depicted in Fig. 10C and D, respectively. While in classical computing calculation proceed sequentially, complex networks of interacting nano objects can combine and process simultaneously various inputs to generate a new state of the system through a complex transformation in accordance with the principle of "reservoir computing."

Reservoir computing represents a unified computational framework derived from Recurrent Neural Network (RNN) models such as liquid state machines (LSMs) [90] and echo state networks (ESNs) [91]. Among Artificial Neural Networks (ANNs), while Feedforward Neural Networks (FNNs) are mainly exploited for static (nontemporal) data processing, RNNs can be exploited for dynamic (temporal) data processing by exploiting their recurrent connectivity with feedback connections. A reservoir computing system can be divided in two unit sections, the first called "*reservoir*" and the second named "*readout*." [92] In the reservoir, temporal input signals are transformed into spatiotemporal patterns in a new space that represents a nonlinear transformation of the input. This projection of the temporal input signal is related to the dynamical evolution of the reservoir state. Then, reservoir states are analyzed by the readout that is exploited to generate the desired output of the system. Importantly, only the weights of the readout

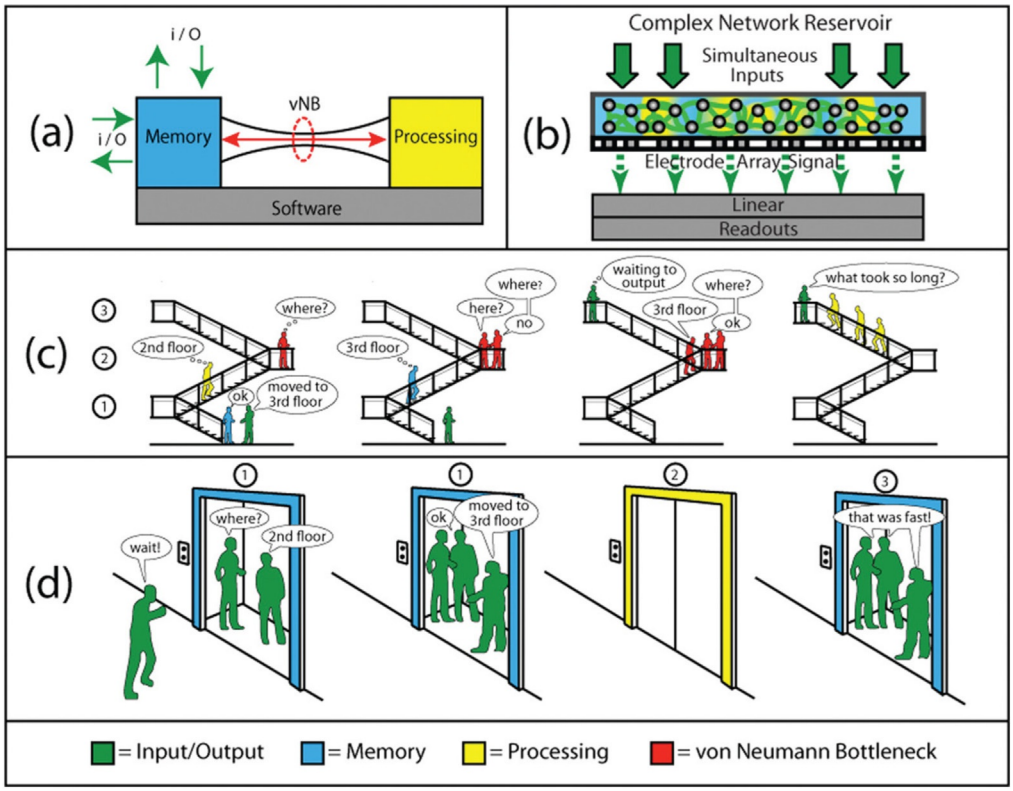


Fig. 10 Comparison of computation in a classical computing architecture and in a complex network reservoir. (A) In the conventional computation, data should be passed forward and backward in between memory and processing units. This architecture intrinsically suffers from the so-called von Neumann bottleneck. (B) Computation in a complex network reservoir where simultaneous inputs are processed into a higher dimensional space that is then analyzed by means of a readout function that perform classification. (C) Schematization of the conventional computing paradigm. Here, calculation proceed sequentially, and each new input *(blue figure)* is delivered to memory *(green figure)*. Due to outdated instructions, earlier processes are unable to produce the desired output and are inactive in the von Neumann bottleneck *(red figures)*. Only when new instructions arrive from memory, calculations can proceed toward the output *(green figure on third floor)*. (D) Schematization of computing in a complex network reservoir. Here, various inputs are combined simultaneously to fill the waiting elevator. Due to mutual interactions, these inputs undergo a complex transformation to generate a new state of the system when arrived at the third floor (output). *(Adapted with permission from A.Z. Stieg, A.V. Avizienis, H.O. Sillin, C. Martin-Olmos, M. Aono, J.K. Gimzewski, Adv. Mater. 24 (2012) 286. Copyright 2020, Wiley-VCH.)*

have to be trained while no training is required for weights of the recurrent connections within the reservoir. For this reason, the main advantage of reservoir computing systems is related to the significant reduction of the computational cost of learning [92,93]. Besides low training cost and fast learning, another advantage is represented by the high versatility of these system. Indeed, since training occurs only at the readout, the same

reservoir can be in principle exploited for solving different tasks because of no interference occurs among tasks during training [93].

Besides nonlinear dynamics, the main characteristic required for the reservoir unit is the so-called echo-state property. This property refers the requirement of the reservoir state to be expressed as a function of the previous input sequence only [93].

Acting as a physical reservoir, self-organized nanowire networks have been proposed as dynamical systems for a physical implementation of reservoir computing taking into advantage of the possibility of processing multiple spatiotemporal inputs in form of electrical stimuli [61]. Here, the reservoir state is represented by the collective state of the NW network elements. The high nonlinear dynamics of the reservoir can be provided by the memristive response of the network to input stimuli, as previously discussed in Section 4.3. Also, the short-term synaptic plasticity effects discussed in Section 4.4 can be exploited to meet the requirements of echo-state property, allowing the reservoir state to depend mainly on recent-past inputs. A schematization of implementation of reservoir computing in NW networks acting as physical reservoir is reported in Fig. 11.

In this context, Sillin et al. [64] demonstrated that an atomic switch network based on NWs can be exploited as a pattern generating kernel in reservoir computing, where the network response can be optimized by adjusting the network connectivity and input gain. Following these principles, simulations have shown that reservoir computing implemented in NW networks can be exploited for solving computing tasks including pattern recognition and time series prediction [94,95]. In the framework of reservoir computing, the computing capability was shown to be intrinsically related to the NW topology. Indeed, simulations by Loeffer et al. [96] show that, while random networks

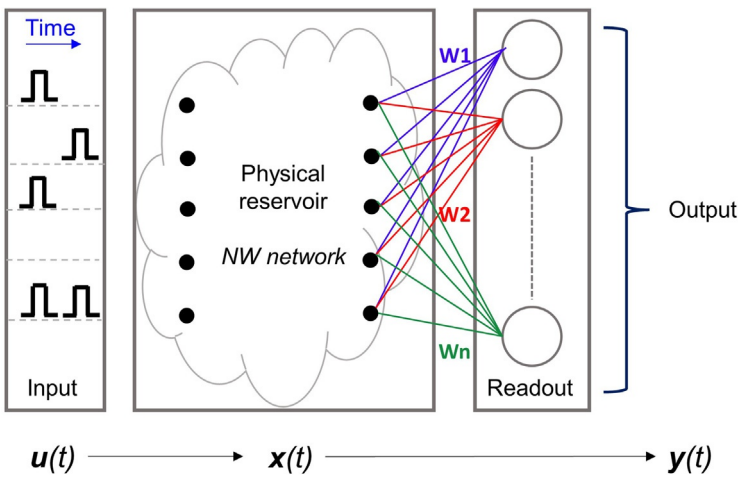


Fig. 11 Basic principle of the implementation of reservoir computing in NW networks acting as physical reservoirs. The physical reservoir maps an input ***u****(t)* in form of pulse streams applied to the NW network to a feature space ***x****(t)* that can be then analyzed by a readout function to obtain the desired output ***y****(t)*. In this computing paradigm, only the readout function needs to be trained.

achieve higher performances on independent tasks, highly modular NW networks achieve best performances in case of simultaneous tasks (multitasking capability).

Diaz-Alvarez et al. [97] showed also that NW networks can be exploited for the realization of a neuromorphic device for supervised learning through associative memory, as shown in a NW network-based multiterminal device. Here, training was performed by taking into advantage of the network memory of preferred conductive pathways in between stimulated electrodes. During following testing, it was shown that the network can correctly retrieve the stored patterns even by testing with incomplete or faulty input patterns thanks to the ability of the network to group correlated input information as shared conductive pathways across the network.

Finally, a fully memristive architecture based on self-organized NW networks in multiterminal configuration coupled with ReRAM devices was demonstrated for *in materia* implementation of reservoir computing for recognition of spatiotemporal patterns and time-series prediction [98], paving the way for the realization of neuromorphic system that combines different hardware technologies.

6. Conclusions

In summary, the current status of memristive devices based on NWs has been discussed from the fundamentals to the realization of NW-based nanoarchitectures for the implementation of unconventional computing paradigms. The NW synthesis has been discussed together with different strategies adopted for the realization of NW-based memristive devices. While single NW devices represent suitable platforms for the investigation of the physical mechanism of switching thanks to the high localization of the switching events and can find application as artificial synapses, NW networks represent low-cost nanoarchitectures where emerging functionalities that arise from the interaction of a multitude of NWs and their junctions can be exploited for the implementation of computing paradigms. Future challenges are represented by tailoring NW material properties to meet the desired memristive functionalities and by integrating NW-based nanoarchitectures with conventional electronics toward the realization of intelligent systems.

References

[1] K. Berggren, Q. Xia, K.K. Likharev, D.B. Strukov, H. Jiang, T. Mikolajick, D. Querlioz, M. Salinga, J.R. Erickson, S. Pi, F. Xiong, P. Lin, C. Li, Y. Chen, S. Xiong, B.D. Hoskins, M.W. Daniels, A. Madhavan, J.A. Liddle, J.J. McClelland, Y. Yang, J. Rupp, S.S. Nonnenmann, K.-T. Cheng, N. Gong, M.A. Lastras-Montaño, A.A. Talin, A. Salleo, B.J. Shastri, T.F. de Lima, P. Prucnal, A.N. Tait, Y. Shen, H. Meng, C. Roques-Carmes, Z. Cheng, H. Bhaskaran, D. Jariwala, H. Wang, J.M. Shainline, K. Segall, J.J. Yang, K. Roy, S. Datta, A. Raychowdhury, Nanotechnology 32 (2021), 012002.

[2] M.A. Zidan, J.P. Strachan, W.D. Lu, Nat. Electron. 1 (2018) 22.

[3] R. Waser (Ed.), Nanoelectronics and Information Technology: Materials, Processes, Devices, John Wiley & Sons, 2012.
[4] L. Chua, IEEE Trans. Circuit Theory 18 (1971) 507.
[5] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, Nature 453 (2008) 80.
[6] Z. Wang, H. Wu, G.W. Burr, C.S. Hwang, K.L. Wang, Q. Xia, J.J. Yang, Nat. Rev. Mater. 5 (2020) 173.
[7] J.J. Yang, M.D. Pickett, X. Li, D.A.A. Ohlberg, D.R. Stewart, R.S. Williams, Nat. Nanotechnol. 3 (2008) 429.
[8] R. Waser, M. Aono, Nat. Mater. 6 (2007) 833.
[9] D. Ielmini, R. Waser, Resistive Switching - From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications, Wiley-VCH, Weinheim, 2016.
[10] Q. Xia, J.J. Yang, Nat. Mater. 18 (2019) 309.
[11] Y. Li, Z. Wang, R. Midya, Q. Xia, J.J. Yang, J. Phys. D. Appl. Phys. 51 (2018), 503002.
[12] D.V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. Le Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, S. Menzel, I. Valov, G. Milano, C. Ricciardi, S.-J. Liang, F. Miao, M. Lanza, T.J. Quill, S.T. Keene, A. Salleo, J. Grollier, D. Marković, A. Mizrahi, P. Yao, J.J. Yang, G. Indiveri, J.P. Strachan, S. Datta, E. Vianello, A. Valentian, J. Feldmann, X. Li, W.H.P. Pernice, H. Bhaskaran, E. Neftci, S. Ramaswamy, J. Tapson, F. Scherr, W. Maass, P. Panda, Y. Kim, G. Tanaka, S. Thorpe, C. Bartolozzi, T.A. Cleland, C. Posch, S.-C. Liu, A.N. Mazumder, M. Hosseini, T. Mohsenin, E. Donati, S. Tolu, R. Galeazzi, M.E. Christensen, S. Holm, D. Ielmini, N. Pryds, 2022 roadmap on neuromorphic computing and engineering, *Neuromorph. Comput. Eng.* 2 (2022) 022501.
[13] J.J. Yang, D.B. Strukov, D.R. Stewart, Nat. Nanotechnol. 8 (2013) 13.
[14] D. Ielmini, Z. Wang, Y. Liu, APL Mater. 9 (2021), 050702.
[15] D. Marković, A. Mizrahi, D. Querlioz, J. Grollier, Nat. Rev. Phys. 2 (2020) 499.
[16] V.K. Sangwan, M.C. Hersam, Nat. Nanotechnol. 15 (2020) 517.
[17] G. Milano, S. Porro, I. Valov, C. Ricciardi, Adv. Electron. Mater. 5 (2019) 1800909.
[18] W. Lu, C.M. Lieber, Nat. Mater. 6 (2007) 841.
[19] D. Ielmini, C. Cagli, F. Nardi, Y. Zhang, J. Phys. D. Appl. Phys. 46 (2013), 074006.
[20] S. Spiga, T. Yanagida, T. Kawai, Resistive Switching, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2016, pp. 661–694.
[21] H.J. Fan, P. Werner, M. Zacharias, Small 2 (2006) 700.
[22] A.L. Tiano, C. Koenigsmann, A.C. Santulli, S.S. Wong, Chem. Commun. 46 (2010) 8093.
[23] N. Wang, Y. Cai, R.Q. Zhang, Mater. Sci. Eng. R Rep. 60 (2008) 1.
[24] L. Güniat, P. Caroff, A.F.i. Morral, Chem. Rev. 119 (2019) 8958.
[25] M. Lanza, H.-S.P. Wong, E. Pop, D. Ielmini, D. Strukov, B.C. Regan, L. Larcher, M.A. Villena, J.J. Yang, L. Goux, A. Belmonte, Y. Yang, F.M. Puglisi, J. Kang, B. Magyari-Köpe, E. Yalon, A. Kenyon, M. Buckwell, A. Mehonic, A. Shluger, H. Li, T.-H. Hou, B. Hudec, D. Akinwande, R. Ge, S. Ambrogio, J.B. Roldan, E. Miranda, J. Suñe, K.L. Pey, X. Wu, N. Raghavan, E. Wu, W.D. Lu, G. Navarro, W. Zhang, H. Wu, R. Li, A. Holleitner, U. Wurstbauer, M.C. Lemme, M. Liu, S. Long, Q. Liu, H. Lv, A. Padovani, P. Pavan, I. Valov, X. Jing, T. Han, K. Zhu, S. Chen, F. Hui, Y. Shi, Adv. Electron. Mater. 1800143 (2018) 1800143.
[26] I. Valov, R. Waser, J.R. Jameson, M.N. Kozicki, Nanotechnology 22 (2011), 289502.
[27] I. Valov, W.D. Lu, Nanoscale 8 (2016) 13828.
[28] G. Milano, M. Luebben, Z. Ma, R. Dunin-Borkowski, L. Boarino, C.F. Pirri, R. Waser, C. Ricciardi, I. Valov, Nat. Commun. 9 (2018) 5151.
[29] E. Miranda, G. Milano, C. Ricciardi, IEEE Trans. Nanotechnol. (2020) 1.
[30] G. Milano, M. Luebben, M. Laurenti, S. Porro, K. Bejtka, S. Bianco, U. Breuer, L. Boarino, I. Valov, C. Ricciardi, Adv. Mater. Interfaces 6 (2019) 1900803.
[31] G. Milano, F. Raffone, M. Luebben, L. Boarino, G. Cicero, I. Valov, C. Ricciardi, A.C.S. Appl, Mater. Interfaces 12 (2020) 48773.
[32] J. Qi, J. Huang, D. Paul, J. Ren, S. Chu, J. Liu, Nanoscale 5 (2013) 2651.

[33] B. Wang, T. Ren, S. Chen, B. Zhang, R. Zhang, J. Qi, S. Chu, J. Huang, J. Liu, J. Mater. Chem. C 3 (2015) 11881.
[34] G. Milano, L. Boarino, C. Ricciardi, Nanotechnology 30 (2019), 244001.
[35] Y. Yang, X. Zhang, M. Gao, F. Zeng, W. Zhou, S. Xie, F. Pan, Nanoscale 2011 (1917) 3.
[36] F. Raffone, F. Risplendi, G. Cicero, Nano Lett. 16 (2016) 2543.
[37] S. Porro, F. Risplendi, G. Cicero, K. Bejtka, G. Milano, P. Rivolo, A. Jasmin, A. Chiolerio, C.F. Pirri, C. Ricciardi, J. Mater. Chem. C 5 (2017) 10517.
[38] R. Waser, R. Dittmann, G. Staikov, K. Szot, Adv. Mater. 21 (2009) 2632.
[39] S.I. Kim, J.H. Lee, Y.W. Chang, S.S. Hwang, K.-H. Yoo, Appl. Phys. Lett. 93 (2008), 033503.
[40] Y.-T. Huang, S.-Y. Yu, C.-L. Hsin, C.-W. Huang, C.-F. Kang, F.-H. Chu, J.-Y. Chen, J.-C. Hu, L.-T. Chen, J.-H. He, W.-W. Wu, Anal. Chem. 85 (2013) 3955.
[41] Z. Fan, X. Fan, A. Li, L. Dong, Nanoscale 5 (2013) 12310.
[42] K. Oka, T. Yanagida, K. Nagashima, T. Kawai, J. Kim, B.H. Park, J. Am. Chem. Soc. 132 (2010) 6634.
[43] K. Nagashima, T. Yanagida, K. Oka, M. Kanai, A. Klamchuen, J.-S. Kim, B.H. Park, T. Kawai, Nano Lett. 11 (2011) 2114.
[44] G. Milano, S. Porro, M.Y. Ali, K. Bejtka, S. Bianco, F. Beccaria, A. Chiolerio, C.F. Pirri, C. Ricciardi, J. Phys. Chem. C 122 (2018) 866.
[45] V. Fra, M. Beccaria, G. Milano, S. Guastella, S. Bianco, S. Porro, M. Laurenti, S. Stassi, C. Ricciardi, Nanotechnology 31 (2020), 374001.
[46] Y. Lai, P. Xin, S. Cheng, J. Yu, Q. Zheng, Appl. Phys. Lett. 106 (2015), 031603.
[47] S. Lee, J. Park, M. Lee, J.J. Boland, AIP Adv. 6 (2016), 125010.
[48] Y. Sun, X. Yan, X. Zheng, Y. Liu, Y. Zhao, Y. Shen, Q. Liao, Y. Zhang, A.C.S. Appl, Mater. Interfaces 7 (2015) 7382.
[49] A. Younis, D. Chu, X. Lin, J. Yi, F. Dang, S. Li, A.C.S. Appl, Mater. Interfaces 5 (2013) 2249.
[50] J. Park, S. Lee, K. Yong, Nanotechnology 23 (2012), 385707.
[51] R. Zhang, W. Pang, Q. Zhang, Y. Chen, X. Chen, Z. Feng, J. Yang, D. Zhang, Nanotechnology 27 (2016), 315203.
[52] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nano Lett. 10 (2010) 1297.
[53] Z. Wang, S. Joshi, S.E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J.P. Strachan, Z. Li, Q. Wu, M. Barnell, G.-L. Li, H.L. Xin, R.S. Williams, Q. Xia, J.J. Yang, Nat. Mater. 16 (2017) 101.
[54] J. Tang, F. Yuan, X. Shen, Z. Wang, M. Rao, Y. He, Y. Sun, X. Li, W. Zhang, Y. Li, B. Gao, H. Qian, G. Bi, S. Song, J.J. Yang, H. Wu, Adv. Mater. 31 (2019) 1902761.
[55] S.G. Kim, J.S. Han, H. Kim, S.Y. Kim, H.W. Jang, Adv. Mater. Technol. 1800457 (2018) 1800457.
[56] D.S. Hong, Y.S. Chen, J.R. Sun, B.G. Shen, Adv. Electron. Mater. 2 (2016) 1500359.
[57] C.J. O'Kelly, J.A. Fairfield, D. McCloskey, H.G. Manning, J.F. Donegan, J.J. Boland, Adv. Electron. Mater. 2 (2016) 1500458.
[58] E. Miranda, G. Milano, C. Ricciardi, IEEE Trans. Nanotechnol. 19 (2020) 609.
[59] E.R. Kandel, J.H. Schwartz, T.M. Jessell, S.A. Siegelbaum, A.J. Hudspeth, Principles of Neural Science, fifth ed., McGraw-Hill Professional, 2013.
[60] M. Aono, K. Ariga, Adv. Mater. 28 (2016) 989.
[61] G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, C. Ricciardi, Adv. Intell. Syst. 2 (2020) 2000096.
[62] A. Diaz-Alvarez, R. Higuchi, P. Sanz-Leon, I. Marcus, Y. Shingaya, A.Z. Stieg, J.K. Gimzewski, Z. Kuncic, T. Nakayama, Sci. Rep. 9 (2019) 14920.
[63] E.C. Demis, R. Aguilera, H.O. Sillin, K. Scharnhorst, E.J. Sandouk, M. Aono, A.Z. Stieg, J.K. Gimzewski, Atomic switch networks—nanoarchitectonic design of a complex system for natural computing, Nanotechnology 26 (20) (2015) 204003.
[64] H.O. Sillin, R. Aguilera, H.-H. Shieh, A.V. Avizienis, M. Aono, A.Z. Stieg, J.K. Gimzewski, Nanotechnology 24 (2013), 384004.
[65] A.V. Avizienis, H.O. Sillin, C. Martin-Olmos, H.H. Shieh, M. Aono, A.Z. Stieg, J.K. Gimzewski, PLoS One 7 (2012), e42772.
[66] A. Loeffler, R. Zhu, J. Hochstetter, M. Li, K. Fu, A. Diaz-Alvarez, T. Nakayama, J.M. Shine, Z. Kuncic, Front. Neurosci. (2020) 14, https://doi.org/10.3389/fnins.2020.00184.

[67] K. Takagi, Front. Comput. Neurosci. 12 (2018) 1.
[68] M.P. van den Heuvel, O. Sporns, J. Neurosci. 31 (2011) 15775.
[69] C. Cagli, F. Nardi, B. Harteneck, Z. Tan, Y. Zhang, D. Ielmini, Small 7 (2011) 2899.
[70] S.M. Oliver, J.A. Fairfield, A.T. Bellew, S. Lee, J.G. Champlain, L.B. Ruppalt, J.J. Boland, P.M. Vora, Appl. Phys. Lett. 109 (2016), 203101.
[71] Y. Ting, J. Chen, C. Huang, T. Huang, C. Hsieh, W. Wu, Small 14 (2018) 1703153.
[72] H.G. Manning, F. Niosi, C.G. da Rocha, A.T. Bellew, C. O'Callaghan, S. Biswas, P.F. Flowers, B.J. Wiley, J.D. Holmes, M.S. Ferreira, J.J. Boland, Nat. Commun. 9 (2018) 3219.
[73] C. O'Callaghan, C.G. Rocha, F. Niosi, H.G. Manning, J.J. Boland, M.S. Ferreira, J. Appl. Phys. 124 (2018), 152118.
[74] Q. Li, A. Diaz-Alvarez, R. Iguchi, J. Hochstetter, A. Loeffler, R. Zhu, Y. Shingaya, Z. Kuncic, K. Uchida, T. Nakayama, Adv. Funct. Mater. 30 (2020) 2003679.
[75] P.N. Nirmalraj, A.T. Bellew, A.P. Bell, J.A. Fairfield, E.K. McCarthy, C. O'Kelly, L.F.C. Pereira, S. Sorel, D. Morosan, J.N. Coleman, M.S. Ferreira, J.J. Boland, Nano Lett. 12 (2012) 5966.
[76] G. Milano, A. Cultrera, K. Bejtka, N. De Leo, L. Callegaro, C. Ricciardi, L. Boarino, A.C.S. Appl, Nano Mater. (2020), acsanm.0c02204.
[77] A. Cultrera, G. Milano, N. De Leo, C. Ricciardi, L. Boarino, L. Callegaro, Sci. Rep. 11 (2021) 13167.
[78] R. Zhu, J. Hochstetter, A. Loeffler, A. Diaz-Alvarez, T. Nakayama, J.T. Lizier, Z. Kuncic, Sci. Rep. 11 (2021) 13047.
[79] D.B. Chklovskii, B.W. Mel, K. Svoboda, Nature 431 (2004) 782.
[80] M. Chistiakova, N.M. Bannon, M. Bazhenov, M. Volgushev, Neuroscience 20 (2014) 483.
[81] D.R. Chialvo, Nat. Phys. 6 (2010) 744.
[82] M. Zare, P. Grigolini, Chaos, Solitons Fractals 55 (2013) 80.
[83] J.P. Sethna, Statistical Mechanics: Entropy, Order Parameters and Complexity, second ed., Oxford Scholarship Online, 2021, https://doi.org/10.1093/oso/9780198865247.001.0001.
[84] N. Srinivasa, N.D. Stepp, J. Cruz-Albrecht, Front. Neurosci. 9 (2015) 1.
[85] A.Z. Stieg, A.V. Avizienis, H.O. Sillin, C. Martin-Olmos, M. Aono, J.K. Gimzewski, Adv. Mater. 24 (2012) 286.
[86] M.D. Pike, S.K. Bose, J.B. Mallinson, S.K. Acharya, S. Shirai, E. Galli, S.J. Weddell, P.J. Bones, M.D. Arnold, S.A. Brown, Nano Lett. 20 (2020) 3935.
[87] J.B. Mallinson, S. Shirai, S.K. Acharya, S.K. Bose, E. Galli, S.A. Brown, Sci. Adv. 5 (2019) eaaw8438.
[88] J. Hochstetter, R. Zhu, A. Loeffler, A. Diaz-Alvarez, T. Nakayama, Z. Kuncic, Nat. Commun. 12 (2021) 4008.
[89] S. Salahuddin, K. Ni, S. Datta, Nat. Electron. 1 (2018) 442.
[90] W. Maass, T. Natschläger, H. Markram, Neural Comput. 14 (2002) 2531.
[91] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Ger. Inf. Technol. Rep 148 (34) (2001) 13.
[92] G. Tanaka, T. Yamane, J.B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, A. Hirose, Neural Netw. 115 (2019) 100.
[93] K. Nakajima, Jpn. J. Appl. Phys. 59 (2020), 060501.
[94] K. Fu, R. Zhu, A. Loeffler, J. Hochstetter, A. Diaz-Alvarez, A. Stieg, J. Gimzewski, T. Nakayama, Z. Kuncic, in 2020 Int. Jt. Conf. Neural Networks, IEEE, 2020, pp. 1–8.
[95] Z. Kuncic, O. Kavehei, R. Zhu, A. Loeffler, K. Fu, J. Hochstetter, M. Li, J.M. Shine, A. Diaz-Alvarez, A. Stieg, J. Gimzewski, T. Nakayama, in 2020 IEEE Int. Symp. Circuits Syst., IEEE, 2020, pp. 1–5.
[96] A. Loeffler, R. Zhu, J. Hochstetter, A. Diaz-Alvarez, T. Nakayama, J.M. Shine, Z. Kuncic, Neuromorphic, Comput. Eng. 1 (2021), 014003.
[97] A. Diaz-Alvarez, R. Higuchi, Q. Li, Y. Shingaya, T. Nakayama, AIP Adv. 10 (2020), 025134.
[98] G. Milano, G. Pedretti, K. Montano, S. Ricci, S. Hashemkhani, L. Boarino, D. Ielmini, C. Ricciardi, Nat. Mater. (2021), https://doi.org/10.1038/s41563-021-01099-9.

CHAPTER 9

Artificial intelligence accelerator using photonic computing

Xing Lin, Sheng Gao, Yi Gao, and Zhengyang Duan
Department of Electronic Engineering, Tsinghua University, Beijing, China

1. Introduction

Artificial neural networks (ANNs), a vital form of artificial intelligence inspired by the signal processing in animal brains, have radically facilitated rapid progress in a multitude of realms in our society, including finance, telecom, medical industry, etc., owing to its high computing capability in processing the large-scale data. However, as the footprint of electronic transistors is approaching its physical limit and the efficacy of Moore's law is gradually waning, augmentation in the performance of traditional electronic computing hardware has been unable to meet the exponential ever-growing computing demands of state-of-the-art ANNs. Therefore, pursuing a new generation of advanced computing technology that can circumvent electronic bottlenecks has attracted a significant amount of attention. Photonic computing, using photons instead of electrons as the information carrier and computing medium, has showcased its potential for building the next-generation advanced intelligent processor due to its remarkable advantages of ultrafast computing speed, high computing parallelism, low energy consumption, and high compatibility with the existing mainstream electronic computing architectures.

However, achieving practical applications of photonic intelligent computing systems, e.g. photonic or optical neural networks [1–8], still confronts the following challenges: (i) Limited computing scale. The number of optical parameters in most existing integrated photonic computing systems is limited to several hundred, which is far left behind the parameters in its electronic neural network counterparts. (ii) Simplicity of the model. The model complexity of existing optical neural networks is relatively simple, and the model performance is far from that of electronic artificial neural networks. (iii) Difficulty in implementing optical nonlinearity. The strong nonlinear neuron activation function in weak light that meets the high-speed optical information processing is difficult to realize all-optically. (iv) Limited system programmability and reconfigurability. Programming the photonic computing system parameters typically depends on complex optical effects, such as laser reading and writing, thermo-optical effect, etc. But it is difficult to high-speed program large-scale parameters accurately in practical applications. Besides, most of the existing photonic intelligence processors cannot be reconfigured for different

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00009-3
Copyright © 2023 Elsevier Inc.
All rights reserved.

types of neural network architectures, resulting in an inability of the processor to perform diverse functions for different applications. (v) Difficulty of system error correction. The system error is positively correlated with the system complexity. Therefore, efficient error correction algorithms are indispensable for the optical computing of practical problems. But the high-accurate universal error correction algorithm is still lacking so far. (vi) Optical data memory. The data conversions between the optical and electronic domains inherently limit the optical data processing speed and energy efficiency. The existing optical data memory technologies still confront the difficulty of simultaneously achieving high integration density, high write and read-out speed, and high energy efficiency.

In this chapter, we summarize the recent research advances in photonic computing for addressing these challenges, following the bottom-up construction of artificial neural network architectures. The related research works are organized into five parts, including the optical weighted interconnections, optical neuron activation functions, optoelectronic devices and AI systems, designing optical neural network architectures, and in-situ optical backpropagation training methods. We also discuss the technical developing tendency and outlook on the future application areas of photonic computing-based AI systems.

2. Optical weighted interconnections

The artificial neural networks involve the fan-in and fan-out of massively parallel signals, which makes it inefficient for electronic computing and impractical for using the electronic metal wire to implement the massive interconnections between layers of neurons. With the breakthrough of corresponding technology in recent years, photonic computing has shown its huge advantages of large-scale parallel computing and extremely low power consumption, which is very suitable for the ultra-high-performance physical realization of neural networks. Moreover, the attenuation and heating of the optical signal transmitted in the waveguide are lower, and there is no inductance and skin effect. By using light wave propagation to implement weighted interconnection and independently control each connection and weight, photonic computing can support massively parallel, reconfigurable, and high-speed connections between distributed photonic processing elements. The basic principles of existing photonic matrix operations for implementing neuron interconnections include interference, resonance, phase change,and diffraction. The research work on optical weighted interconnections can be divided into two categories as follows, i.e., the free-space and integrated implementations.

2.1 Free-space optical weighted interconnection

Diffraction is the fundamental property of light waves, which can be utilized to construct diffractive interconnections to execute optical operations. The basic computing unit of the photonic neural network and all-optical weighted interconnection constructed based on the diffraction principle is a phase modulator at the optical wavelength scale or the

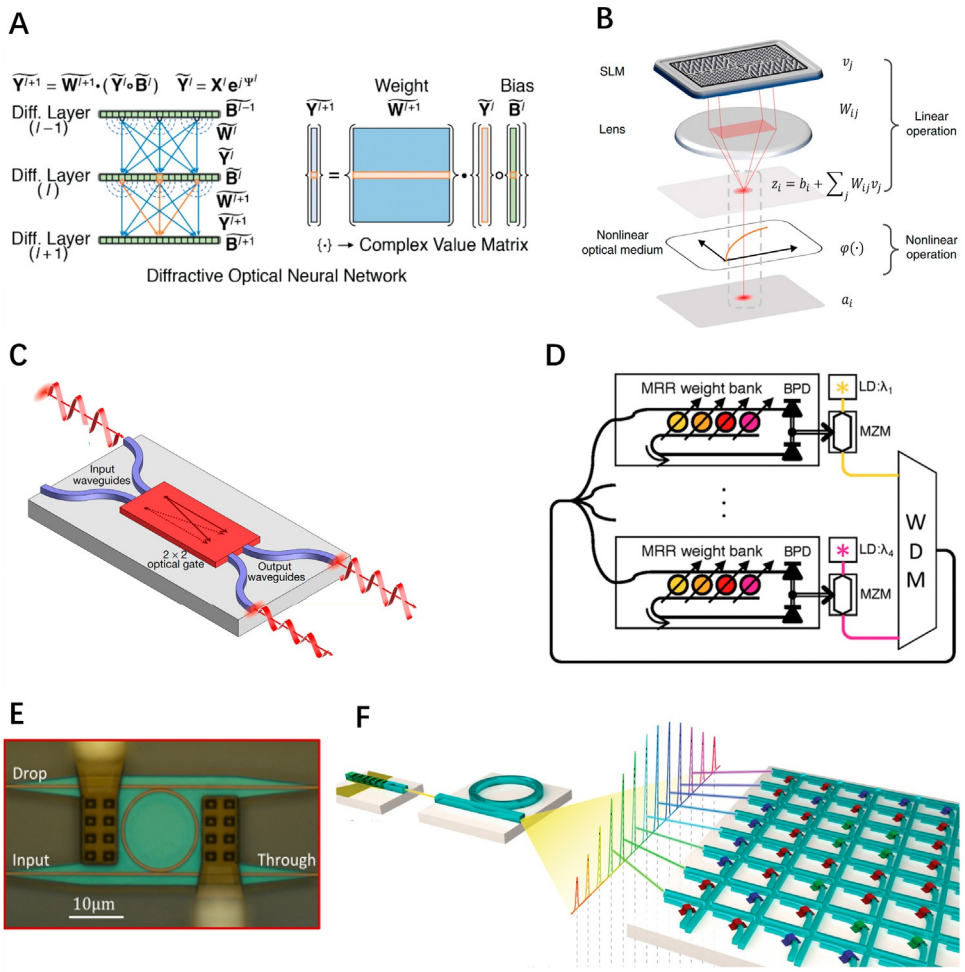


Fig. 1 Optical weighted interconnections. (A) The all-optical diffractive neural network uses the optical diffractive phenomenon and the transmission or reflection coefficient of the diffractive layer to realize weighted interconnection. (B) Schematic illustration of the experimental implementation of linear and nonlinear operations of optical neurons. Linear operation is achieved by combining programmable SLMs and Fourier lenses. (C) In the photonic-integrated circuit, the most basic block 2×2 analog optical gate in MZI realizes weighted interconnections. (D) The broadcast-and-weight protocol uses the MRR weight bank to achieve tunable spectral filtering for WDM signals. (E) Schematic diagram of the structure of MRR in an add-drop configuration. (F) The nonvolatile PCM unit is used to store the analog value of the convolution kernel in-situ and achieve optical in-memory computing.

subwavelength metasurface structure scale. Based on the Huygens-Fresnel principle, an energy-efficient phased array with multi-layer diffraction modulation can be established, which realizes the controllable modulation of diffractive optical field propagation and wavefront spherical sub-wave point wave source [1]. The all-optical diffractive deep

neural network (D^2NN) uses multiple diffractive layers to achieve the controllable light-matter interactions during the light wave propagation, where each point on the diffractive layer can be regarded as an artificial neuron. The complex transmissive or reflective coefficient on the layer's surface modulates the received wave and connects to the following layer through optical diffraction (as shown in Fig. 1A).

Based on Huygens' theorem, each neuron on the diffractive layer generates a spherical secondary wave after receiving the incident wave. The amplitude and phase of the secondary waves are modulated by the neuron's complex transmission or reflection coefficient (which can be regarded as a multiplicative bias term). And then, it is connected to each neuron in the next layer through optical diffraction (which can be regarded as a weighted item), and the neuron sums the waves in all diffractive directions. According to the Rayleigh-Sommerfeld diffraction formula:

$$U(P_0) = \frac{1}{j\lambda}\iint_{\Sigma} U(P_1)\frac{e^{jkr}}{r}\cos\theta ds,$$

where $U(P_1)$ is the complex field of the secondary wave at P_1; P_0 is any point in space; r is the distance between P_0 and P_1 θ is the angle between the unit wave surface normal and P_0P_1 direction. Therefore, we can design

$$h(P_1P_0) = \frac{1}{j\lambda}\frac{e^{jkr}}{r}\cos\theta,$$

where the physical meaning of $h(P_1, P_0)$ is the diffractive weight at P_0 of the secondary wave generated by the unit area ds at P_1, which is also called the impulse response. Therefore, we can consider that the optical mode of the secondary wave generated by the neurons in each layer of D^2NN in the free-space diffraction is as follows:

$$w_i^l(x, y, z) = (z - z_i)/r^2\left(\frac{1}{2\pi r} + \frac{1}{j\lambda}\right)\exp\left(\frac{j2\pi r}{\lambda}\right),$$

where l represents the l-th layer of the network; i represents the i-th neuron of the l-th layer; the position coordinates are (x_i, y_i, z_i) and (x, y, z) are the coordinates of the next layer neurons. Therefore, $w_i^l(x, y, z)$ represents the diffractive weight between two neurons. In addition, the amplitude and phase of the secondary wave are modulated by the complex transmission or reflection coefficient t_i^l on the surface of the neuron, which can be regarded as the multiplicative bias term of the neural network. From the above, we can get the output wave from the i-th neuron of the l-th layer to any neuron of the $l+1$-th layer:

$$n_i^l(x, y, z) = w_i^l(x, y, z)\cdot t_i^l(x_i, y_i, z_i)\cdot\sum_k n_i^{l-1}(x_i, y_i, z_i) = w_i^l(x, y, z)\cdot|A|\cdot e^{j\Delta\theta},$$

where $\sum_k n_i^{l-1}(x_i, y_i, z_i)$ is the input wave of the neuron; $|A|$ corresponds to the amplitude of the secondary wave generated by the neuron; and $\Delta\theta$ corresponds to the phase of the secondary wave. These coherent secondary waves propagate from layer to layer and

interfere with each other to form a complex optical field at the front end of neurons in the next layer. In this way, optical weighted interconnection in free space is realized. But D^2NN can only realize the fully connected weighted interconnections, and the diffractive weights cannot be adjusted after the fabrication.

In addition to the optical realization of the fully connected layer discussed above, the convolutional layer can also be implemented by Fourier optics. Miscuglio et al. [2] proposed an amplitude-only (AO) Fourier convolutional layer based on the Fourier optics 4*f* system and high-resolution digital micromirror device (DMD), which can perform classification tasks on two-megapixel images at a refresh rate of 10 kHz. The DMD and the 4*f* system together implement the optical weighted interconnection. A reconfigurable all-optical neural network is proposed to realize both linear and nonlinear optical computing operations [3]. In this optical design (shown in Fig. 1B), the programmable linear operation is implemented based on the spatial light modulation (SLM) and Fourier lens, while the nonlinear operation is implemented based on the cold atom system with electromagnetically induced transparency (EIT) effect.

2.2 Integrated optical weighted interconnection

With the gradual maturity of silicon-photonic-integrated circuits (PIC), it is possible to use integrated photonic devices to realize on-chip optical weighted interconnection. The basic computing unit to realize all-optical weighted interconnection based on the interference principle is the on-chip integrated Mach-Zehnder Interferometer (MZI). As shown in Fig. 1C, each MZI controls the power distribution and phase delay of the optical field through a waveguide beam splitter and a phase shifter, causing the optical fields in the strip waveguide to interfere with each other, and the photonic matrix operation is realized by cascading MZI mesh [4]. The basic computing unit to realize the all-optical weighted interconnection based on the resonance principle is the Micro-ring Resonator (MRR, shown in Fig. 1E), which combines the wavelength division multiplexing technology to encode the input signal and construct the neuron MRR weight bank. By configuring the resonance state of the MRR and broadcasting the wavelength division multiplexed signal to modulate the optical field intensity, the network weight adjustment and weighted interconnection are realized.Tait et al. [5] proposed a broadcast-and-weight protocol as shown in Fig. 1D. The broadcast-and-weight protocol consists of a set of neurons sharing the same input WDM signal, each with an MRR weight bank at the front end.

The basic computing unit to realize all-optical weighted interconnection based on the phase change principle is the Structured Phase Change Material (S-PCM). Phase-change materials can be optically written to switch between crystalline and amorphous states, thereby modulating the intensity of the optical field. Because PCM has high optical contrast and long-term stability, it is often used in rewritable optical disk memory technology [6]. Many recent works combine PCM with PIC to realize in-memory optical computing, which can bring many advantages, such as high bandwidth, wavelength division multiplexing, and low cross talk. Feldman et al. [7] demonstrated a wafer-level-integrated

photonic tensor core to complete parallel convolution processing, which is capable of operating at the computing speed of 10^{12} MAC operations per second. The integrated photonic tensor core uses PCM to store the convolution kernel matrix on-chip and implements the parallelizing convolution operations by combining the on-chip frequency comb and WDM technology (shown in Fig. 1F). Compared with the thermo-optic or electro-optic effects used in MZIs or MRRs, PCM can achieve the nonvolatile storage and programming of weight and does not require the energy source to maintain the weighted interconnections.

The basic computing unit on-chip interferometers, micro-ring resonators, and structured phase change materials that the above three types of principles rely on need to occupy a space of tens to hundreds of micrometers in size, resulting in the low integration of computing units, limiting the scalability of the scale of on-chip photonic matrix operation, and is difficult to apply to the weighted interconnection between large-scale photonic neurons. Since the size of the diffractive phase modulator can be reduced to wavelength and sub-wavelength scales, it can be applied to the diffractive weighted interconnection of large-scale photonic neurons. However, the current diffractive photonic neural network is mainly based on the spatial light calculation realized by free-space light diffraction. Although some research work has begun to explore its on-chip integration [43], high-integration on-chip diffractive optical field propagation and wavefront modulation accurate models have not yet been established, and it is difficult to apply to high-precision all-optical weighted interconnects and diffractive photonic neural network chips design. Therefore, it is necessary to establish a set of high-precision chip design theories and methods for on-chip diffractive photonic computing.

3. Optical neuron activation functions

Since a wide range of problems in the real world can fundamentally come down to solve all kinds of mathematical relations, artificial neural networks should ideally be able to generalize to fit arbitrary mathematical functions. However, it is impossible to approximate arbitrary mathematical functions by merely exploiting optical weighted interconnection, as it can merely perform linearity. Therefore, optical neuron activation functions, which can provide nonlinearity, are introduced into optical neural networks for large-scale AI computations at high complexity. Researchers have realized optical neuron activation functions on the basis of different optical effects, including photorefractive effect, saturable absorption, optical Kerr effect, structural phase transition, optical resonance, photoelectronic effect, etc.

3.1 Photorefractive effect

The photorefractive effect originates in the variance among spatial optical intensity, which induces the photoionization in space and hence generates large amounts of free charge carriers, changing the distribution of the local electric field. As a result, the refractive index of the

medium is altered with respect to the electric field, leading to an ultimate nonlinear phase modulation to the incident light. For example, SBN:60, a kind of ferroelectric crystal, is capable of performing nonlinearity based on the photorefractive effect. The nonlinear relation between phase transition Δn and illumination intensity I can be described as follows:

$$\Delta n = \frac{k E_{app} \langle I \rangle}{1 + \langle I \rangle},$$

where $\langle I \rangle$ is the fluctuation of illumination intensity against a spatially uniform distributed background intensity $\langle I_0 \rangle$; E_{app} is the electric field along c axis; and κ is a constant relevant to the electro-optic coefficient.

Based on the nonlinearity property of SBN:60, Yan et al. [8] conducted their research on Fourier-space diffractive deep neural network (F-D^2NN), where SBN:60 was placed as nonlinear layers in the shape of a ferroelectric thin film (Fig. 2A). To create sufficient nonlinearity for phase modulation, Yan et al. [8] set the thickness of SBN:60 to be 1 mm and adjusted the voltage on it to be 972 V, making the relation between phase and intensity to be

$$\Delta \Phi = \frac{\pi \langle I \rangle}{1 + \langle I \rangle},$$

where phase variation ranges from 0 to π. Owing to the nonlinearity performed by SBN:60, the classification accuracy of F-D^2NN based on the MNIST dataset increases significantly, reflecting an augmentation of its capability in high-level representations for target scenes.

In retrospect, the advantage of the photorefractive effect is its low requirement for the intensity of the input beam. The minimal illumination for the operation of SBN:60 is merely at the scale of $\sim 0.1\,\mathrm{mW/mm^2}$. However, the response time of SBN:60 is highly relevant to the intensity of illumination, and there is a certain amount of energy consumption owing to absorption, reflection, and dark current of SBN:60.

3.2 Saturable absorption

Saturable absorption denotes the phenomenon where the absorption of photons of a medium (e.g., graphene) began to decline as the intensity of the input signal exceeded a specific threshold, hence producing nonlinearity. The nonlinearity of saturable absorption is realized by Guo et al. [9], who deployed a strong pump in an atomic vapor. Based on the phenomenon of saturable absorption, the pump output is in a nonlinear relation with its input, formulated as follows (see Fig. 2B):

$$E_{P,out} = g(E_{P,in}) = \exp\left(-\frac{\alpha_0/2}{1 + E_{P,in}^2}\right) E_{P,in},$$

where α_0 indicates resonant optical depth. The saturable absorption based on graphene is now implemented in a wide range of photonic devices owing to the high gain property of graphene.

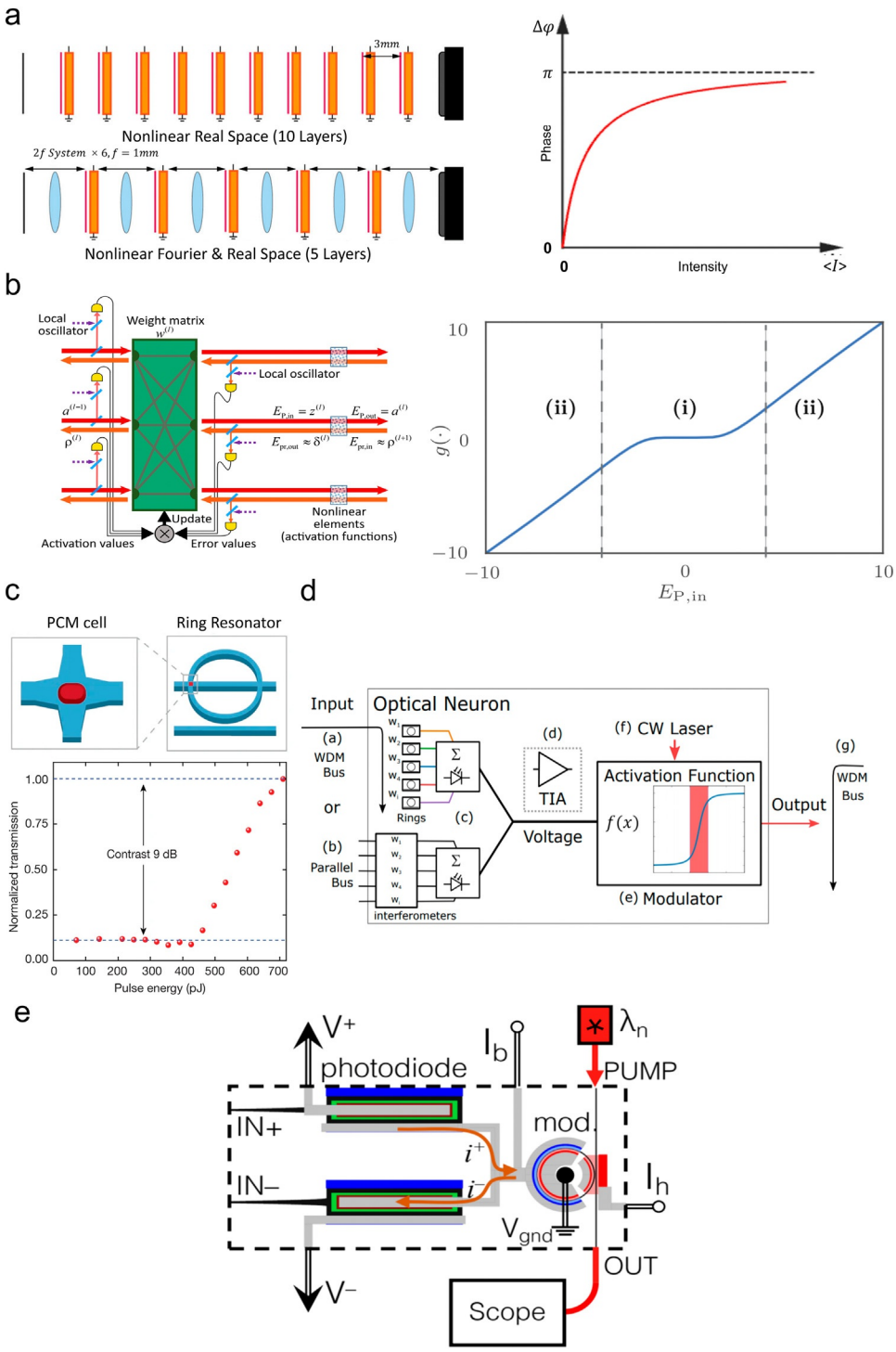


Fig. 2 Optical neural activation functions. (A) The nonlinear response of SBN:60 Ferroelectric thin film used in the diffractive deep neural networks. (B) Saturable absorption nonlinear effect under different resonant optical depths. (C) ReLu function obtained by structural phase transition effect. (D) The paradigm for implementing electro-optical activation function. (E) The nonlinear microring modulator based on the optical resonance effect.

3.3 Optical Kerr effect

The optical Kerr effect phenomenon occurs as the intense input signal exerts different extents of changes between the refractive index of horizontal polarized direction and perpendicular polarized direction. It causes a variation of refractive index Δn, which is proportional to the intensity of illumination in the medium. Therefore, a nonlinear relation between the refractive index of the medium n and input illumination I is generated. Based on a theoretical prove given by Sui et al. [10], the optical Kerr effect can be described as follows:

$$n = n_0 + n_2 I,$$

where n indicates current refractive index under an illumination intensity I; n_0 represents original refractive index of the medium; and $n_2$is a nonlinear coefficient.

Application of the nonlinearity based on the Kerr effect in artificial neural networks was proposed by Skinner et al. [11], who managed to perform both linear operation (weighted connections) and nonlinear activation function simultaneously in thin layers of materials. Based on the optical Kerr effect, the approach increases the processing speed of the network drastically. In addition, the simple physical construction of the network also guarantees its mass production potentiality in the future.

3.4 Structural phase transition

Structural phase transition denotes an exceptional property of phase change material (PCM), whose extent of light absorption is disparate between its two states: amorphous state and crystalline state. In the amorphous state, a large amount of light can be transmitted through the PCM cell; in the crystalline state, the PCM cell absorbs most of the light passing through it. Based on such property, Feldmann et al. [12] coupled PCM with an optical ring resonator to perform nonlinearity, as shown in Fig. 2C. Since the difference in the energy of pulses can drive PCM to different extent of amorphous state, the actual state condition of the PCM is a combination of amorphous and crystalline states. Owing to a disproportional absorption of light between these two phases, the light transmitted through PCM is therefore in a nonlinear relation with the input illumination.

In addition, by fixing the input signal at a specific wavelength, the input signal becomes in resonance with the ring resonator, and a sharp contrast between the maximum and minimum output of the activation function is hence realized. For example, when the input signal has a resonance wavelength of 1553.4 nm with an intensity below 430 pJ, the light transmitted through the ring resonator is zero. When the input intensity transcends that threshold, the output intensity increases significantly. Therefore, a nonlinear activation function assuming the shape of ReLu is created, with a sharp contrast of 9 dB between its maximum and minimum output. In fact, the proposed device has extraordinarily low energy consumption, marking a big step forward in optical nonlinearity.

3.5 Optical resonance

The principle of nonlinearity based on optical resonance is to take advantage of the resonance dip produced by microring (MRR). The resonance dip is formulated as the light transmitted through the microring (MRR) modulator decreases significantly when the input wavelength is in resonance with MRR, shaping a nonlinear function between output transmission of MRR and input wavelength. In addition, since current biasing applied on MRR can affect its spectra of resonance dip, a reconfigurable activation function can hence be realized.

Based on this phenomenon, an electro-optic architecture was proposed by George et al. [13] (as shown in Fig. 2D) and instantiated by Tait et al. [14] (as shown in Fig. 2E). In their architecture, input is weighted by rings from a WDM. Then, photodiodes convert illumination into photocurrents, which is the reflection of the summed inputs. After that, the photocurrents are biased with the modulator current I_b, which can affect the depth and quality of the MRR modulator resonance peak. Simultaneously, the photocurrents are also biased with a heater current I_h, which is capable of shifting the absolute wavelength with an efficiency of 0.24 nm/mW. Finally, these three currents altogether drive an electro-optic modulator (MRR), performing nonlinearity on a brand-new CW laser as an output. In this way, the output and input are decoupled in the process of conversion between illumination and photocurrents. In particular, the shape of the activation function produced by the MRR modulator can be reconfigured by adjusting the heater current bias I_h, which changes the resonance frequency of MRR. As a result, sigmoid can be realized by biasing the current bias I_h at the maximum slope point; ReLu can be obtained by biasing the current bias I_h slightly off resonance, namely, either below or above the wavelength.

3.6 Photoelectronic effect

The principle of nonlinearity based on electro-optic is to convert a portion of the input optical signal into electric currents, which thence intensity-modulate the original optical signal by shifting its phase. Based on this idea, Williamson et al. [15] proposed an electro-optic-based nonlinear activation function method with a significant performance in both multiinput exclusive-OR (XOR) logic function and MNIST dataset for classifying handwritten numbers. In their architecture, as shown in Fig. 2F, the input light with a total power of $|z|^2$ firstly enters into a directional coupler, where a portion of illumination $\alpha|z|^2$ was sent into photodiodes (PD). The optical signal is then transferred into an electrical current $I_{pd} = R\alpha|z|^2$, where R is the responsivity of a photodetector. And after transmitting through an amplifier gain G with a nonlinear transform function $H(\cdot)$ and a static bias voltage V_b, a phase shift

$$\Delta\phi = \frac{\pi}{V_\pi}\left[V_b + H\left(GR\alpha|z|^2\right)\right]$$

was induced for the illumination light, where V_π indicates the amount of voltage required to induce a phase change of π. By combining the transmission formula of MZI, we obtain

$$t_{MZI} = j\exp\left(-j\frac{\Delta\phi}{2}\right)\cos\left(\frac{\Delta\phi}{2}\right).$$

The activation function of incident light can be explicitly described as:

$$f(z) = j\sqrt{1-\alpha}\exp\left(-j\frac{1}{2}\left[\phi_b + \pi\frac{H\left(GR\alpha|z|^2\right)}{V_\pi}\right]\right)\cos\left(\frac{1}{2}\left[\phi_b + \pi\frac{H\left(GR\alpha|z|^2\right)}{V_\pi}\right]\right)z,$$

where $\phi_b = \pi\frac{V_b}{V_\pi}$. In the light of this, a reconfigurable activation function based on electro-optic conversion can be physically realized by setting different voltage V_b.

In summary, the ideal realization of neuron activation functions in optical domain holds great fortes in two aspects: ultrafast speed and approximately free consumption of energy. Activation functions can be instantaneously performed as the illumination transmitted from the nonlinearity devices at the speed of light, and no power supply is required for the sustainability of optical nonlinearity devices. However, the incident light is required to be highly intense in many cases (including optical Kerr effect) at present stage. To respond to this challenge, a plentiful of new optical materials and mechanisms are emerging to overcome the difficulty of realizing all-optical strong activation functions in low light condition for high-speed information processing, thence accelerating the advent of photonic computing processor in the near future.

4. Designing photonic neural network architectures

With the development of on-chip photonic devices and components, the architecture of photonic neural networks has become more and more complex, resulting in a substantial increase in performance. The photonic neural network can directly convert the paradigm of the electronic neural network to achieve photonic fully connected neural networks, photonic convolutional neural networks, and photonic recurrent neural networks. In view of the unique characteristics of photonics and optics, some brand-new network architectures such as photonic reservoir neural networks and photonic spiking neural networks can be implemented. Based on various photonic neural networks, computing

tasks including image recognition, audio classification, and video recognition have been efficiently implemented.

4.1 Photonic fully connected neural networks

Free space optics provides a medium for massively parallel free-space interconnections and data information processing at the speed of light. The physical diffractive layer provides a method for storing and realizing the weight matrix. All-optical diffractive deep neural network (D^2NN) in terahertz spectrum (0.4 THz) has been shown to be able to conduct classification tasks at the speed of light, which is composed of multiple 3D-printed diffractive layers [1]. By designing the architecture of the 3D-printed layer (shown in Fig. 3A), the functions of image classification and imaging lens can be implemented, respectively. Yan et al. [8] proposed a Fourier-space diffractive deep neural network (F-D^2NN), which possesses extremely compact diffractive layers and introduces nonlinear films, and finally demonstrated that its performance is superior to D^2NN on object classification and visual saliency detection tasks. However, the aforementioned 3D-printed diffractive layers cannot be reconfigured, thus limiting their applications.

The advancement of on-chip optical devices and components has led to substantial progress in the research of photonic-integrated circuits (PIC). Shen et al. [16] proposed a programmable integrated optical neural network consisting of 56 MZI cascaded arrays, which can perform vowel recognition tasks at the speed of light (shown in Fig. 3B). This ONN has two layers, and each layer contains two optical interference units (OIU) to realize arbitrary optical matrix multiplication. The programmable ONN architecture can passively and optically perform calculations, which is theoretically two orders of magnitude faster than electronic neural networks. However, it is necessary to further explore the large-scale integrated optical interconnection with high integration density photonic computing unit. Besides, the nonlinear activation function plays an important role in ONN, so an electro-optic hardware platform is designed for OIU to realize the optical-to-optical nonlinear activation function [15].

Compared with 2D integration, the 3D integration with a volumetric nanophotonic neural medium (NNM) has a higher degree of freedom to implement the fully connected neuron interconnections [17]. NNM is composed of a host material containing a large number of inclusions, where the host material is silicon dioxide, and the inclusions can be any material with a reflection index different from that of the host material. Although the proposed NNM neural network can only achieve 79% handwritten digit recognition accuracy due to the limited computational scale, it has the advantages of ultra-high computing density, stronger expression capability, and no gradient disappearance.

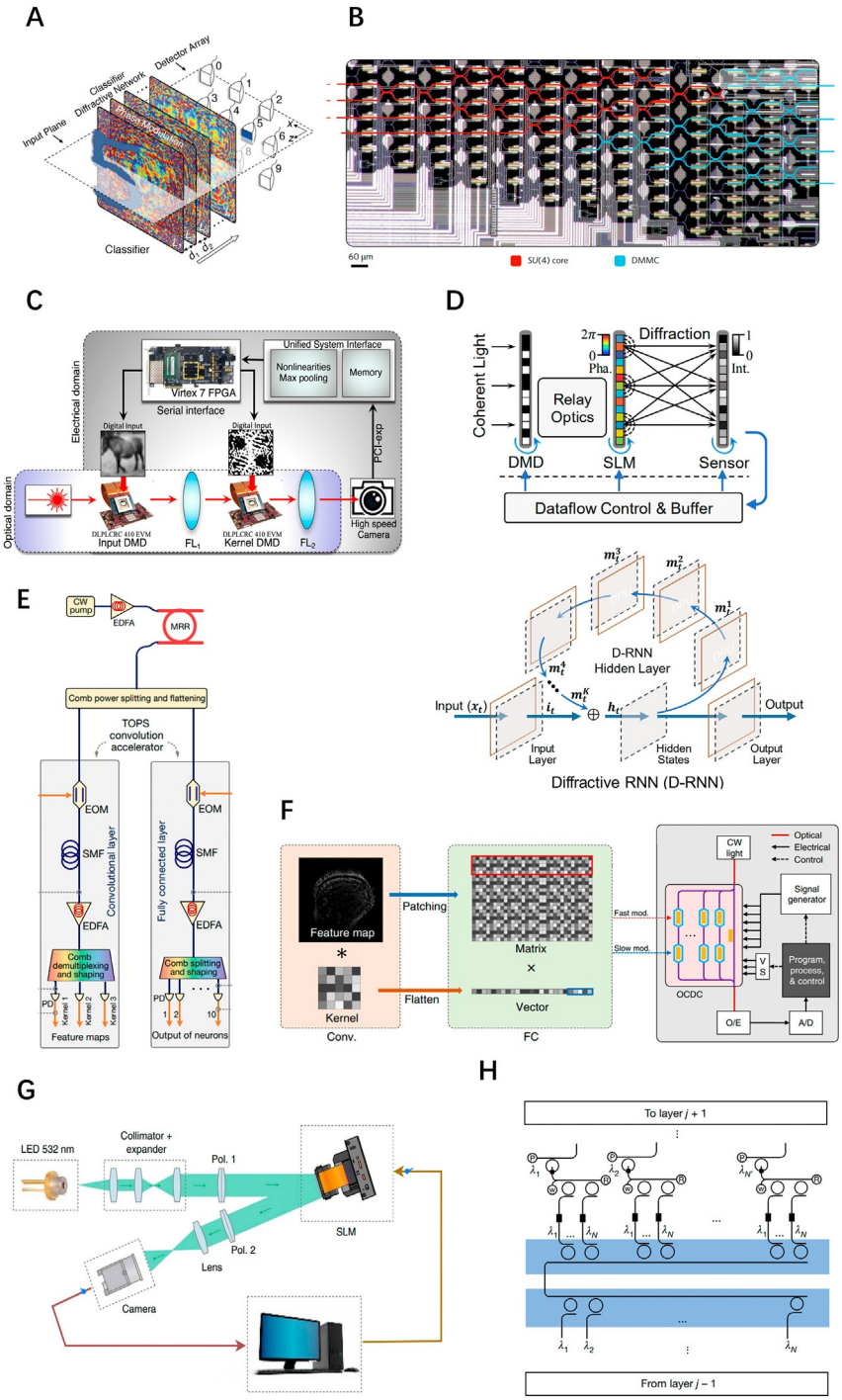


Fig. 3 Photonic neural network architectures. (A) D^2NN is composed of five 3D printed physical diffractive layers, which can be used for the classification of handwritten digits and fashion products. (B) Optical micrograph illustration of the photonic-integrated circuit OIU. (C) System architecture of the reconfigurable amplitude-only optoelectronic convolution processor. (D) Optoelectronic implemented DPU and constructed a recurrent neural network. (E) Photonic convolutional accelerator based on the optical microcomb and optoelectronic devices. (F) The schematic for implementing convolution operations based on OCDC. (G) Schematic diagram of the optical reservoir neural network. (H) Scaling architecture for all-optical on-chip spiking neural network.

4.2 Photonic convolutional neural networks

In electronic neural networks, the convolutional layer consumes the majority of computing resources and causes significant computational latency. Photonic convolutional neural networks implement the convolution operations optically that can improve the computational efficiency of CNN. Miscuglio et al. [2] proposed a reconfigurable amplitude-only (AO) optoelectronic convolution processor (see Fig. 3C) that uses a reprogrammable high-resolution DMD to perform classification tasks with a large two-megapixel matrix at a rate of 10 kHz. An efficient neural vision system has also been proposed, which can realize the energy-free convolutional operation at the optical front end by encoding the imaging aperture, which is suitable for incoherent light processing in natural scenes [18]. Through experimental verification of the neural vision system, character and gesture recognition tasks can be achieved optically with extremely low lantency and close to state-of-the-art classification accuracy performance. Another study [19] showed that the energy consumption and latency of the hybrid photonic-electronic architectures outperformed conventional electronic platforms for large-sized input images and convolution kernels.

In addition to utilizing the optical 4*f* system to implement the optical convolutional layer, PCM-based optical in-memory computing also possesses the capability to efficiently conduct parallel convolutional processing. Based on the on-chip optical frequency comb and nonvolatile PCM memory array, an integrated photonic tensor core that can implement parallel photonic in-memory convolution computing is proposed [7]. The photonic tensor core utilizes WDM technology to achieve high-computational bandwidth and can process multiple image patches or even complete images in a single time step. A phase-change metasurface mode converter (PMMC) based on the PCM ($Ge_2Sb_2Te_5$) is proposed [20], which can achieve up to 64 levels of phase change mode contrast corresponding to a 6-bit weight precision. By building a 2×2 PMMC array, programmable convolution kernels can be implemented and successfully applied to edge detection and pattern recognition tasks.

Different from the PCM in-memory computing, a scalable and reconfigurable photonic convolutional accelerator is proposed which utilizes integrated microcombs, electro-optic modulators, and single-mode fibers to achieve simultaneous time, wavelength, and space multiplexing [21]. The photonic convolutional accelerator (shown in Fig. 3E) can process 250,000-pixel input images, with a computing speed of up to 11 TOPS, and successfully implemented the task of classifying handwritten digits. Moreover, this paradigm is universal and fully compatible with other electronic or photonic interfaces. A silicon-based optical coherent dot-product chip (OCDC) is proposed [22], which can perform matrix multiplication and convolutional operations in the real-valued optical domain based on push-pull modulation. Based on OCDC (shown in Fig. 3F), a single-layer fully connected layer and a convolutional layer can be implemented, and a

cutting-edge neural network AUTOMAP is further constructed. In the above-mentioned optical CNN, input data patching is a necessary process. An optical patching scheme based on WDM and optical delay lines is proposed [23], which can effectively reduce the scale of the input modulator array.

There are many other methods to implement photonic convolutional neural networks, such as based on 2D or 3D waveguide architecture design. A new all-optical on-chip CNN is proposed [24], based on cascaded Y-waveguides with side-coupled silicon waveguide segments, which can implement transcendental equation solvers and a variety of mathematical operations. Different from the traditional implementation of 2D photonic neural networks using photolithography, Moughames et al. [25] proposed a 3D photonic waveguide architecture based on fractal topology, which can achieve an extremely large number of inputs and outputs in a very small footprint interconnection.

4.3 Photonic recurrent neural networks

Recurrent neural networks possess recurrently connected hidden layers and nodes. These nodes extract and store the information of inputs from previous time instances and pass it to the next time step, so it is suitable for processing temporal/sequential data. Zhou et al. [26] proposed a reconfigurable diffractive processing unit (DPU), which can construct various complex optoelectronic neural networks, and use in-situ adaptive training to circumvent accumulated system errors, and finally achieve experimental performance that exceeds the cutting-edge electronic computing platforms. As an optoelectronic neuromorphic processor with large-scale diffractive neurons, DPU is composed of an input layer, the diffractive interconnection, and an output layer (shown in Fig. 3D). By controlling and buffering massive parallel optoelectronic data streams, DPU can construct a large-scale diffraction recurrent neural network (D-RNN). Through large-scale neuromorphic optoelectronic computing, DPU achieves eight times higher calculation speed than Tesla V100 GPU and more than an order of magnitude system energy efficiency.

The optical reservoir neural network is a special recurrent neural network that can be applied to wireless channel equalization, spoken digital recognition [27], and prediction of the future evolution of chaotic time series. In the optical reservoir neural network, all weights of input nodes and recurrent nodes are randomly distributed, and specific machine learning tasks are implemented by training the weights of output nodes. Therefore, it has the advantages of easy training and smaller training datasets. A typical discrete-time optical reservoir neural network includes N internal variables $x_{i\in 0\ldots N-1}$, and the evolution of the discrete-time at the next moment can be expressed as

$$x_i(n+1) = f_{NL}\left(\sum_{j=0}^{N-1} W_{ij}x_j(n) + \sum_{j=0}^{K-1} B_{ij}u_j(n)\right),$$

where f_{NL} is a nonlinear function; W_{ij} is the reservoir adjacency matrix; B_{ij} is the input mask; and $u_j(n)$ is the input time series. The dynamics of the reservoir are determined by W_{ij} and B_{ij}, and both matrices are independent of time and taken from a random distribution with zero means. The output node of the reservoir neural network is obtained by the linear combination of internal variables:

$$y_l(n) = \sum_{j=0}^{N-1} W_{lj}^{out}x_j(n),$$

where W_{lj}^{out} is the trainable output weight. An optoelectronic reservoir neural network is proposed in [28], which can accommodate up to 16,384 reservoir nodes and efficiently realize human action recognition in video processing tasks. The experimental setup of the reservoir neural network is composed of free-space optical components and an electronic computer (shown in Fig. 3G). Compared with the electronic neural network, the optoelectronic reservoir neural network possesses obvious characteristics of shorter training time and comparable classification performance.

4.4 Photonic spiking neural networks

The development of artificial intelligence is limited by the traditional memory-computing separated computing architecture. But neuromorphic computing methods effectively overcome these limitations by emulating brain-like neurons and synapses in hardware, such as photonic spiking neural networks. Feldmann et al. [12] proposed an integrated all-optical spiking neural network, which is based on the fundamental phase-change all-optical neurons implemented through PCM and WDM technology, and demonstrated the ability to perform supervised and unsupervised learning. The phase-change all-optical neuron is composed of a ring resonator integrated with the nonvolatile PCM. All-optical spiking neural networks are composed of basic phase-change all-optical neurons, which are scalable and reconfigurable, enabling more complex deep learning tasks. The envisioned spiking neural network contains an input layer, an output layer, and multiple hidden layers (as shown in Fig. 3H).

In summary, there are many types of photonic neural network architectures that have been implemented and existing differences in implementation methods, neuron models, training methods, and network architectures. Since the scale of photonic integration is limited, and the construction of on-chip intelligent photonic computing processor for optoelectronic fusion is limited by a large number of optical-electro and electro-optical conversion devices, so the existing photonic neural network architecture is simple, and

it is difficult to apply to intelligent photonic computing tasks of complex scenes and objects. Advanced on-chip photonic neural network architectures supporting large-scale photonic integration need to be explored for more complex intelligent photonic computing tasks.

5. Optoelectronic devices and AI systems

The pursuit of all-optical AI chips and systems is impeded by a lack of fundamental on-chip optical devices, including assemblers, compilers, logic gates, analog-digital-analog (ADA) converters, on-chip light generators, and memory, which are widely used in electronic processors. In response to this challenge, a plentiful of research regarding the physical realizations of these optical devices are carried out. However, given that these photonic devices are still functionally scattered and unable to merge into an end-to-end computing system, the state-the-of-art photonic AI system still needs to take advantage of electronic devices. Therefore, in this chapter, we will discuss optoelectronic devices and AI systems in four major aspects: on-chip light sources, photonic memory, photonic modulator, and optoelectronic AI accelerator systems.

5.1 On-chip light source

The inability to generate light by silicon gives rise to two major approaches to getting light onto silicon chips: (i) using an external optic fiber (off-chip method); (ii) using a non-silicon die (on-chip method). However, the off-chip method loses its allure owing to its relatively high package expense and large coupling losses. On the other hand, on-chip methods, with their tremendous potentiality in obtaining higher integration density, emerge as the most popular research orientation at present.

The method of searching for opportune non-silicon material for on-chip applications was proposed by Zhou et al. [29], indicating four criteria: (i) emission wavelength of 1310 or 1550 nm for fiber connection; (ii) laser under electrical pumping for compact size; (iii) high power efficiency for limited energy cost; and (iv) integration of silicon-based on CMOS for mass manufacturing. Under these criteria, III–V material emerges as the most popular material in realizing the on-chip method currently. However, the integration of III–V material onto silicon chip is partly impeded by its mismatch with silicon in terms of the crystal lattice and thermal expansion coefficients. To address this issue, three major integration approaches are meticulously studied: heterogeneous integration, hybrid integration, and heteroepitaxial growth.

Heterogeneous integration refers to the bonding between a patterned III–V die onto patterned silicon circuits, enabling the light generated in the epitaxial layers of III–V to vertically couple into silicon circuits by evanescent coupling. The advantage of heterogeneous integration, which is its essential superiority over hybrid integration, is its low requirements for precision positioning, as all structures are lithographically aligned for wafer-scale processing. However, the convenience in precision positioning of

heterogeneous integration is tempered by its spontaneous inability to pretest each light source before integration, indicating a radical increase in the requirements for the yield of the light source.

Hybrid integration denotes integrating individual laser diodes one at a time on silicon substrates enabling a pretest of laser diodes before integration. However, this operation requires meticulous alignment for each laser diode, consuming a considerable amount of time and assembly cost. Fortunately, this drawback is ameliorated by Rodlin et al. [30] in their groundbreaking research (as shown in Fig. 4B), where a three-dimensional (3D) photonic wire bonds (PWB) with an ability to allow for adaptions to exact positions was implemented to eliminate the demand of precision alignment. The laser sources in their research are placed side by side with the silicon platform, and the coupling losses hence plummet to merely 0.4 dB, making photonic wire bonding a general approach for photonic multichip assemblies based on hybrid integration.

Heteroepitaxial growth is the epitaxial growth of a deposit on a substrate of different materials, which requires some degree of mobility of the atoms and nuclei on the surface. However, the epitaxial growth of III–V material on silicon is constrained by a phenomenon called high-density threading dislocations (TDs), which is a ramification of the disparity in thermal expansion coefficients between silicon and III–V material. Traditionally, the effective technique for eradicating TDs is called epitaxial lateral overgrowth (ELOG), where a buffer layer with adequate thermal expansion coefficient and lattice constant is deposited between silicon and III–V material, enabling a defect-free top surface layer to be formed. In fact, high-quality quantum wells (QW) grown on ELOG InP can obtain a light emission intensity that is comparable to those grown on a planar InP. On the other hand, a new approach to constraining TDs, the quantum dots (QD) technology, has surpassed the QW technique in the last few years. The advantages of QDs are major three aspects: (i) limited threshold current density (J_{th}); (ii) low-temperature susceptibility; and (iii) tolerance against defects owing to discrete distribution (as shown in Fig. 4C). For example, Chen et al. [31] proposed an InAs/GaAs QD laser with a threshold density of merely 62.5 A/cm^2 and can operate efficiently at a temperature up to 120°C.

5.2 Photonic memory

In the process of photonic computing, the preservation of a large quantity of intermediate arithmetical results for subsequent computations necessitates high-speed memory for storage. Therefore, photonic memory with fast access times, increased bandwidth, and scaled footprint are pressing in recent years. Naturally, the most likely approach for realizing photonic memories is to replicate the mechanism of its electronic memory counterparts, which are widely used in the existing processors. However, the inherent nature of photons and their neutral charge property hampers their potentiality of emulating electronic memory based on the capacitor. To circumvent this problem, researchers exploited optical bistability to realize both optical volatile and nonvolatile memory.

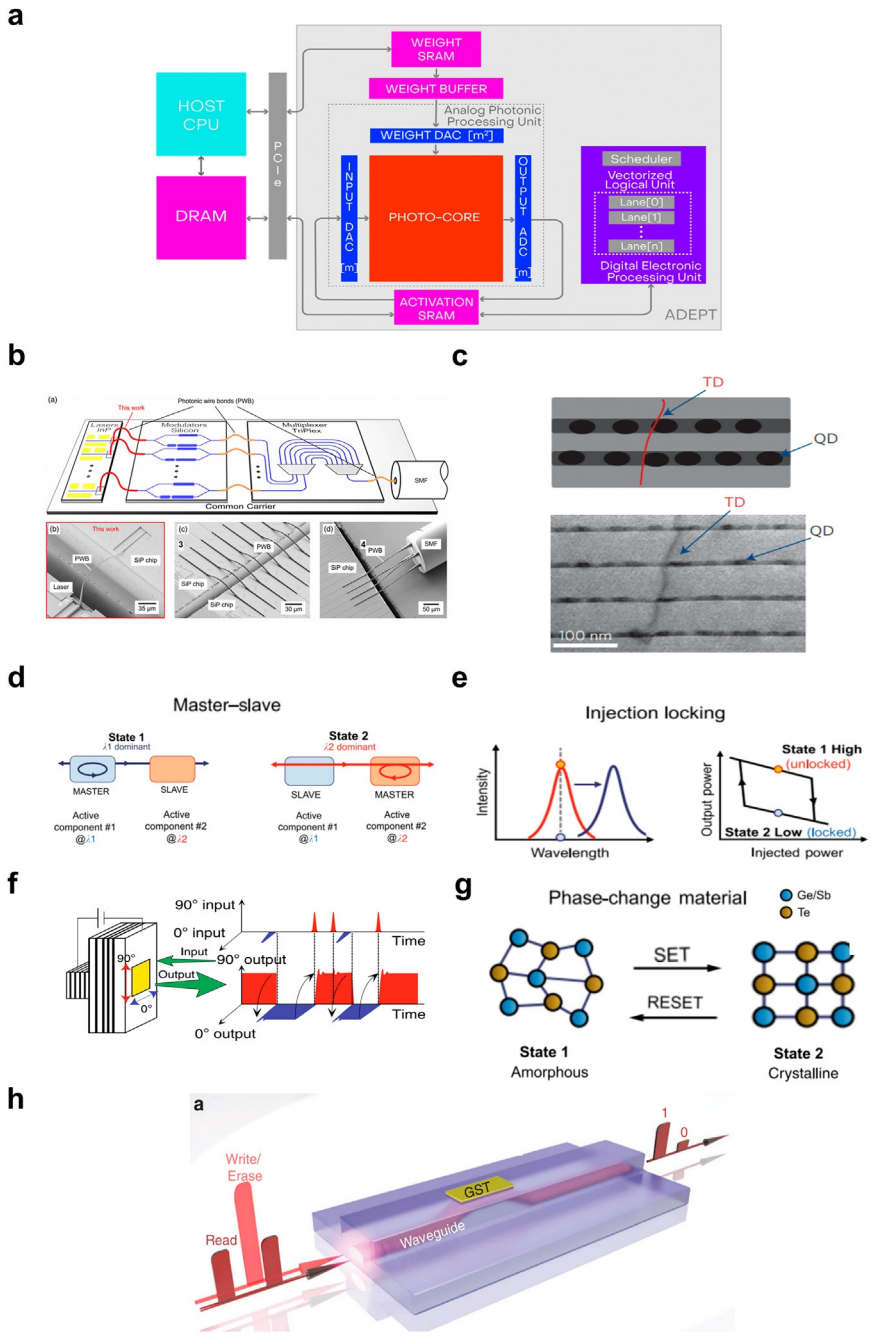


Fig. 4 Optoelectronic devices and AI systems. (A) The architecture of a photonic computing-based electro-photonic system for accelerating AI computations. (B) 3D photonic wire bonds (PWB) are applied in hybrid integration. (C) Quantum dots (QD) applied in minimizing TDs in heteroepitaxial growth. (D) Master-slave layout in photonic SRAM memory. (E) An SOA-MZI photonic SRAM memory based on master-slave layout. (F) Injection locking layout in photonic SRAM memory. (G) A VCSEL photonic SRAM memory based on injection locking layout. (H) Phase transition material (PCM) layout in photonic nonvolatile memory. (I) A nonvolatile all-optical multibit memory based on a type of PCM material, i.e., GST.

Optical volatile memory can be realized through two major disability layouts: (a) master-slave layout and (b) injection locking layout, as put forward by Alexoudi et al. [32]. Master-slave layout comprises two active cavities, as shown in Fig. 4D, where bistability is realized by storing light of two different wavelengths in the respective cavity to represent two distinctive states: 0 and 1. The transition of states can be realized by externally injecting an opportune amount of beam into the "master" to suppress its operation, waiting for the quantity of light in the "slave" to increase, hence interchanging the previous status of two cavities. For example, Pitris et al. [33] proposed a semiconductor optical amplifier (SOA)-based bistable photonic memory, whose binary states are represented by two continuous-wave (CW) input signals. The advantage of SOA layouts is their theoretically operating speed of 40 Gb/s. However, the energy consumption of state transition (~3 pJ, ~0.5 pJ) and current biasing (~120 and ~180 pJ) are of large quantity, and the footprint of SOA is at a tremendous size of $6 \times 2\,\text{mm}^2$.

Injection locking layout, on the other hand, represents optical memory bistability based on free-running mode (unlocked) and injection-controlled mode (locked), as shown in Fig. 4E. The general principle of injection locking is to force certain characteristics (wavelength, polarization state) of the original laser to follow the corresponding characteristics of the externally injected optical signal. For example, injection locking based on polarization can be realized through vertical-cavity surface-emitting lasers (VCSELs), as shown in Fig. 4F. The externally injected signal comprises two choices of polarized direction: vertical or orthogonal, and the polarized direction of the original signal is derived from being in accordance with that of the injected signal. Therefore, bistability can be obtained by controlling the polarized direction of the injected signal. The advantages of the VCSEL-based injected locking layout are: (i) high operating speed up to 40 Gb/s and (ii) low energy consumption of 105 fJ. However, the footprint of VCSEL is of ~36 μm.

As for nonvolatile photonic memory, phase change material (PCMs) can indicate two distinct states by its inherent binary phase (as shown in Fig. 4G): crystalline state and amorphous state. The transition between the binary states can be subnanosecond with a slight amount of energy consumption. Given that the amorphous state and crystalline state absorb light transmitted through them to different extents, information written in the PCM in the form of crystalline or amorphous state can be encoded into the light beam instantly by simply propagating the light through the PCM. In fact, a nonvolatile PCM memory based on $Ge_2Sb_2Te_5$ (GST) was realized by Ríos et al. [34] with a capacity of 5 bits storage within an area of merely $4 \times 1.3\ \mu\text{m}^2$, and operates at a frequency of 1 GHz powered by merely 13.4 pJ, as shown in Fig. 4H.

5.3 Optoelectronic modulator

In the process of transmitting, transferring, and receiving optical signals, an optical modulator plays an indispensable role in encoding information onto the amplitude or phase of

the beam, and there are a plethora of approaches for modulating incident light, including spatial light modulation (SLM) and Mach-Zehnder modulator (MZI). Bueno et al. [35] designed a large-scale recurrent neural network consisting of up to 2025 diffractively coupled photonic nodes. The modulator of this model is realized through a spatial light modulator (SLM) and a digital micromirror device (DMD), which denotes the binary states and creates a spatially modulated image of the state in SLM, respectively. The output of the model is hence calculated via superimposing the modulated intensities. In the light of this, the fully parallel, passive weight layout of the network maximizes energy efficiency and bandwidth, realizing its outstanding performance and potential widespread applications. Shen et al. [16] experimentally put forward a programmable nanophotonic processor with a cascaded array of 56 programmable Mach-Zehnder interferometers (MZI) on a silicon-photonic-integrated circuit. The modulator of this architecture is supported by waveguides and directional couplers with phase shifters acting as a unitary transformer. In fact, the weight matrix can be further transformed by the singular value decomposition method, where $M = U\Sigma V$, among which unitary matrices U and V are implemented experimentally by MZIs and diagonal matrix Σ is implemented by Mach-Zehnder modulator (MZM). The performance of the proposed processor was tested on a vowel recognition problem with an accuracy comparable to the conventional 64-bit computer using a fully convolutional neural network (CNN) algorithm.

5.4 Optoelectronic AI systems

To merge all these devices into an organic intelligent processor, Demirkiran et al. [36] proposed a remarkable optoelectronic AI accelerator (as shown in Fig. 4A), with outstanding performance in all metrics regarding throughput, power efficiency, and power-area efficiency, marking the tremendous potential of photonics in accelerating AI networks in the future.

Within their architecture, all electrical devices are constructed around an analog photonic core, which is specialized in performing general matrix-matrix multiplication (GEMM), as GEMM constitutes over 90% of a total number of floating-point operations during the AI computation. As for the left 10% non-GEMM operations, a traditional digital electronic ASIC is applied. With the ultrafast processing speed of the photonic core, the accelerator was tested on ResNet-50, BERT-large, and RNN-T networks with a better power efficiency of 6.35×, 4.27×, and 10.96×, better power-area efficiency of 5.85×, 3.92×, and 10.08× than systolic arrays, circumventing the power wall that impedes electronic accelerators from higher performances currently.

In summary, the state-of-the-art optoelectronic AI accelerators have exhibited much more outstanding performance than their electrical counterparts in metrics including power efficiency and power-area efficiency, which proves the tremendous performance of photonic computing. However, the realization of an all-optical AI accelerator is still restricted

by a lack of cornerstones, including optical memory and an on-chip light generator. Fortunately, a multitude of research has proposed several applicable methods for these fundamental devices, propelling the progress of realizing all-optical AI accelerators in the future.

6. In-situ optical backpropagation training methods

Traditional deep learning relies on a large amount of labeled data and computing resources for network training, uses the backpropagation algorithm to update the network weights, and finally obtains a network architecture that can complete advanced machine learning tasks. To date, the training process of most ONNs has also been performed entirely digitally on a computer, and the optimized parameters are then used to deploy the optical device. Such an in-silico training method requires accurate simulation of wave-matter interactions, so it has high-computational complexity and long training time and does not take full advantage of photonic computing. In addition, due to various imperfect factors of the optical system, the deviation of the pretraining model and the actual calculation error are accumulated layer by layer. In contrast, the in-situ optical backpropagation training method of ONNs can physically implement the training process directly in the optical system, thereby greatly improving the time and energy efficiency of network training. More importantly, in-situ training can reduce the impact of different error sources on the performance of the pre-trained network in the experimental implementation. A variety of optical in-situ training methods have been successfully implemented. A brute force in-situ training method for MZI-integrated circuits is proposed [16]. Reinforcement learning can also be used for in-situ training of photonic recurrent neural networks [35]. Research hotspots in recent years have mainly focused on the in-situ optical backpropagation training method based on gradient descent, which can significantly improve the performance of ONNs in machine learning tasks [37]. It is mainly divided into two categories: gradient calculation based on optical backpropagation and gradient calculation based on electronic backpropagation.

6.1 Gradient calculation based on optical backpropagation

Since the gradient calculation in the backpropagation algorithm is a computationally expensive process, many scholars have studied how to utilize the parallelism and energy efficiency of optical computing to achieve in-situ gradient calculation in optical neural networks. An in-situ optical backpropagation training method for diffractive optical neural networks is proposed [38]. Based on the principle of optical reciprocity and phase conjunction, the weight gradient of the diffractive layer can be calculated fully optically. The diffractive ONN architecture is implemented by the cascaded SLM (shown in Fig. 5A),

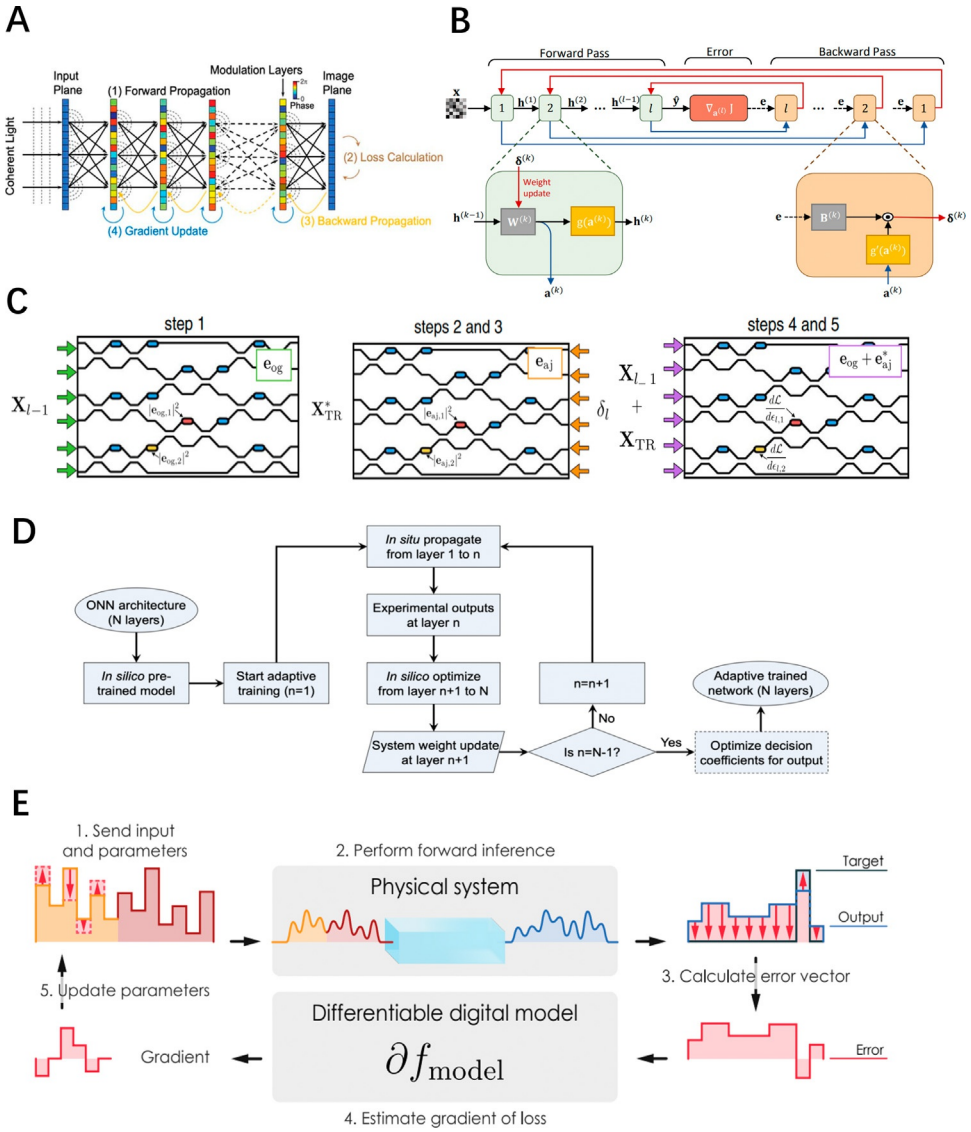


Fig. 5 In-situ optical training methods. (A) The in-situ optical training process of diffractive ONN is realized by cascading SLM. (B) Scheme for the direct feedback alignment algorithm. (C) Schematic diagram of the MZI phase shifter gradient experimental measurement based on the adjoint variable method. (D) Flowchart of the proposed adaptive training method based on hierarchical in-situ training. (E) Physics-aware training implemented by hybrid physical-digital structure.

which can update the diffractive weights in real-time. The in-situ backpropagation training method includes four steps: forward propagation, loss calculation, backward propagation, and gradient update. Let $L(O, T)$ represent the loss function of ONN, which measures the difference between the network output O and the ground truth label T, then the gradient of the loss function relative to the diffractive weight ϕ_k of the k-th layer can be expressed as:

$$\frac{\partial L}{\partial \phi_k} = 2\,\mathrm{Re}\left\{ j\left(\left(\prod_{i=k}^{1} M_i W_i\right) U_0\right) \odot \left(\left(\prod_{i=k+1}^{1} W_i^T M_i\right) W_{N+1}^T E\right)\right\}^T$$
$$= 2\,\mathrm{Re}\left\{ j P_k^f \odot P_k^b \right\}^T,$$

where $M_k = \mathrm{diag}(e^{j\phi_k})$ is the diffractive modulation coefficient controlled by SLM; W_k is the diffractive propagation matrix; and U_0 is the optical field at the input layer. Therefore, $P_k^f = (\prod_{i=k}^{1} M_i W_i) U_0$ represents the output optical field at the k-th layer by forwarding propagation. According to the optical reciprocity, W_k^T represents the diffractive propagation matrix from the k-th layer to $(k-1)$-th by backward propagation. E is the error optical field obtained by the network output and ground truth. Therefore, $P_k^b = (\prod_{i=k+1}^{N} W_i^T M_i) W_{N+1}^T E$ represents the error optical filed of the k-th layer by backward propagation. By measuring P_k^f and P_k^b, the gradient of the loss function can be calculated exactly and fully optically. The proposed in-situ optical backpropagation training method is not only superior to electronic training in training speed and computational energy efficiency but also can adapt to diffractive layer misalignment and other system imperfections, indicating the realization of large-scale reconfigurable diffractive ONN.

Different from the diffractive ONN, the on-chip integrated ONN implemented by the cascaded MZIs array requires calculating the gradient of each phase shifter to update network parameters. An optical backpropagation gradient calculation method based on the adjoint variable method (AVM) is proposed [39], which can perform efficient in-situ training of the ONN implemented by the MZI mesh. The key to the AVM method is to update the weight of each phase shifter, which can be implemented by in-situ intensity measurement of the forward propagation optical field and error backpropagation adjoint optical field (shown in Fig. 5C). Compared with the brute force search method, this training method has better parallelism and scalability.

Optical backpropagation gradient computation can take full advantage of photonic computation, but it is difficult to establish optical field backpropagation for photonic neural network architectures, and the accuracy of gradient computation is limited. The fusion of optical forward-propagation and electronic backpropagation gradient computation can improve its programmability and computational accuracy.

6.2 Gradient calculation based on electronic backpropagation

A promising in situ training approch is to combine experimental optical system measurements with computer-simulated training to achieve higher model inference accuracy.

Zhou et al. developed an adaptive training method that iteratively fine-tuned the model after in-silico pretraining to avoid system errors and achieved experimental accuracy that closes to the state-of-the-art electronic neural networks [26]. As shown in Fig. 5D, the adaptive training method first performs in-silico training on a computer to obtain a pre-trained model, then uses the in-situ experimental measurement output of each layer of ONN as the input of the next layer, and retrains the parameters of all subsequent layers on the computer. Compared with the above-mentioned in-situ optical backpropagation training method, this adaptive training method can effectively solve the model deviation caused by different error sources and improve the robustness and generalization of the optoelectronic neural network system.

Physical neural network (PNN) refers to the hierarchical physical computing architectures that can be trained. Wright et al. proposed a physics-aware training (PAT) method based on in situ forwarding propagation of PNN using the experimental system and in-silico backpropagation with gradient calculation using computer [40]. The proposed PAT approach is verified by performing image classification on three different physical systems. One of the physical systems is the ultrafast optical second harmonic generation (SHG), which is based on quadratic nonlinear optical media. An ultrafast nonlinear ONN can be realized by reusing multiple SHGs, and the classification result can be obtained by measuring the largest spectrum bin. The implemented PAT is a hybrid physical-digital architecture, as shown in Fig. 5E. PAT takes advantage of the in situ measured system outputs during the neural network training so that the trained model is resilient to noise and system imperfections.

As an alternative to the classic backpropagation training algorithm, the direct feedback alignment (DFA) algorithm can effectively perform supervised training of conventional deep neural networks [41]. Different from the backpropagation algorithm, DFA uses a fixed random feedback connection to directly propagate network errors from the output layer to each hidden layer, which can realize the gradient update of multiple hidden layers at the same time (shown in Fig. 5B). Using DFA to calculate the gradient of each hidden layer can be expressed as

$$\delta^k = B^k e \odot g'\left(a^k\right),$$

where e is the gradient of the error loss function; B^k is a fixed random weight matrix of the k-th hidden layer; $g'(\cdot)$ is the derivative of the activation function; and a^k is the weighted input signals of the k-th hidde layer. Then the weight of each hidden layer can be updated iteratively by using the calculated gradient δ^k. Filipovich et al. proposed a CMOS-compatible silicon photonic architecture that can perform in-situ DFA training of ONNs in parallel and efficiently [42]. According to theoretical analysis, the computing speed is as high as 24 TOPS, and the energy consumption per MAC operation is less than 1 pJ. And through experimental demonstration, it is verified that the performance of in-situ DFA training can be comparable to the backpropagation algorithm on the task of handwritten digit classification. The implemented optoelectronic architecture can perform ultrafast

neural network in-situ DFA training, demonstrating the great potential of photons in artificial intelligence applications.

In summary, although the all-optical backpropagation gradient calculation can fully utilize the parallelism, extremely low energy loss, and time cost of optical computing, there is still a problem that the system is too complicated in actual implementation. Therefore, the compromise method is to use the physical ONN to achieve in-situ forward propagation and some matrix-vector multiplication operations, and the gradient calculation is digitally implemented in a compatible electronic CMOS processor. In this way, the advantages of photonic computing and electronic computing can be complemented, and the prediction accuracy and system robustness of ONN can be greatly improved.

7. Discussion and outlook

With the growth of electronic computing encountering a bottleneck and cannot meet the needs of large-scale artificial intelligence tasks, photonic computing has become the main driving force for the development of the next generation of technology due to its extremely high parallelism and energy efficiency. The basic operation of weighted interconnection of neural networks can be efficiently implemented with free-space optical systems and photonic-integrated circuits: (1) In a free-space system, the weighted interconnection between large-scale neurons can be achieved through 3D printing diffractive layers, where the programmable SLM and DMD can also be used to achieve phase and amplitude modulation. (2) In photonic-integrated circuits, light can be modulated through cascading MZI and MRR array as well as PCM to realize photonic in-memory computing. The physical realization of the ideal all-optical neuron activation function has two significant advantages: ultrafast processing speed and almost free device energy consumption. At present, optical nonlinearity has been achieved based on a variety of optical effects, including photorefractive effect, saturable absorption, optical Kerr effect, structural phase change, optical resonance, optoelectronics, etc. The development of basic optoelectronic devices is the cornerstone of implementing an efficient optoelectronic computing platform. At present, photonic I/O that emits light on a chip is mainly realized through three integration methods: heterogeneous integration, hybrid integration, and heteroepitaxial growth; photonic memory is mainly based on bistable SRAM implemented by the state layout and nonvolatile memory based on PCM. In view of the performance degradation of the existing ONN system due to its imperfection, a variety of in-situ optical backpropagation training methods have been proposed, which are divided into optical gradient calculation and digital gradient calculation. There are many types of ONN architectures, including photonic fully connected networks, photonic convolutional neural networks, photonic recurrent neural networks, and photonic spiking neural networks.

Free-space and integrated optoelectronic AI accelerators have demonstrated performance comparable to traditional electronic neural networks in a variety of machine learning tasks. Including image classification, visual saliency detection, high-speed video

image classification, voice classification, optical character, gesture recognition, edge detection, image reconstruction, and other artificial intelligence tasks. The latest research [26] shows that the optoelectronic AI accelerator can achieve up to 99.0% classification accuracy on the MNIST classification task and 90.2% accuracy on the Fashion-MNIST dataset, which is comparable to the classification performance of the electronic neural network LeNet-5. On the Weizmann and KTH human action datasets, the optoelectronic AI accelerator can achieve up to 100% and 96.3% human action recognition accuracy, which surpasses theadvanced electronic computing approaches. Moreover, as the most advanced computing processor, the optoelectronic AI accelerator has overwhelming performance than electronic components in terms of power efficiency and computing speed, thus verifying the huge potential of using photons as a computing medium in the next generation of global processors.

We expect that the final intelligent photonic integrated circuits can realize real-time processing of massive information carried by light, which will greatly promote its application in terminal computing, edge computing, and data center, accelerating the development of autonomous driving and robotics. Among them, the miniaturized free-space intelligent optoelectronic computing system can be applied to free-space optical communication. It can also be integrated with the visual imaging system to realize real-time processing of visual information, including high-dimensional spectral visual information and light-field visual information. At the same time, the integrated optoelectronic intelligent computing system can also be integrated with optical fiber communication to realize the efficient processing of large-scale optical information, which can be applied to the fields of wireless communication, cloud computing, and data center, thus leading to the transformation of the new generation of intelligent industry and information technology.

References

[1] X. Lin, et al., All-optical machine learning using diffractive deep neural networks, Science 361 (2018) 1004–1008.
[2] M. Miscuglio, et al., Massively parallel amplitude-only Fourier neural network, Optica 7 (2020) 1812–1819.
[3] Y. Zuo, et al., All-optical neural network with nonlinear activation functions, Optica 6 (2019) 1132–1137.
[4] W. Bogaerts, et al., Programmable photonic circuits, Nature 586 (2020) 207–216.
[5] A.N. Tait, et al., Broadcast and weight: an integrated network for scalable photonic spike processing, J. Lightwave Technol. 31 (2014) 4029–4041.
[6] X. Li, et al., Fast and reliable storage using a 5 bit, nonvolatile photonic memory cell, Optica 6 (2019) 1–6.
[7] J. Feldmann, et al., Parallel convolutional processing using an integrated photonic tensor core, Nature 589 (2021) 52–58.
[8] T. Yan, et al., Fourier-space diffractive deep neural network, Phys. Rev. Lett. 123 (2019) 2.
[9] X. Guo, et al., Backpropagation through nonlinear units for all-optical training of neural networks, 2019.
[10] X. Sui, et al., A review of optical neural networks, IEEE Access 8 (2020) 70773–70783.
[11] S.R. Skinner, J.E. Steck, E.C. Behrman, Optical neural network using Kerr-type nonlinear materials, in: International Conference on Microelectronics for Neural Networks & Fuzzy Systems IEEE, 1994.

[12] J. Feldmann, et al., All-optical spiking neurosynaptic networks with self-learning capabilities, Nature 569 (2019) 208–214.
[13] J. George, et al., Neuromorphic photonics with electro-absorption modulators, Opt. Express 27 (2019) 5181–5191.
[14] A.N. Tait, et al., Silicon photonic modulator neuron, Phys. Rev. Appl. 11 (2019), 064043.
[15] I. Williamson, et al., Reprogrammable electro-optic nonlinear activation functions for optical neural networks, IEEE J. Sel. Top. Quantum Electron. 1 (2019) 99.
[16] Y. Shen, et al., Deep learning with coherent nanophotonic circuits, Nat. Photonics 11 (2017) 441–446.
[17] E. Khoram, et al., Nanophotonic media for artificial neural inference, Photon. Res. 7 (2019) 823–827.
[18] P. Pad, et al., Efficient neural vision systems based on convolutional image acquisition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12285–12294.
[19] S. Colburn, et al., Optical frontend for a convolutional neural network, Appl. Opt. 58 (2019) 3179–3186.
[20] C. Wu, et al., Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network, Nat. Commun. 12 (2021) 96.
[21] X. Xu, et al., 11 TOPS photonic convolutional accelerator for optical neural networks, Nature 589 (2021) 44–51.
[22] S. Xu, et al., Optical coherent dot-product chip for sophisticated deep learning regression, Light Sci. Appl. 10 (2021) 221.
[23] S. Xu, et al., Optical patching scheme for optical convolutional neural networks based on wavelength-division multiplexing and optical delay lines, Opt. Lett. 45 (2020) 3689–3692.
[24] K. Liao, et al., All-optical computing based on convolutional neural networks, Opto-Electron. Adv. 4 (2021), 200060.
[25] J. Moughames, et al., Three-dimensional waveguide interconnects for scalable integration of photonic neural networks, Optica 7 (2020) 640–646.
[26] T. Zhou, et al., Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit, Nat. Photonics 15 (2021) 367–373.
[27] K. Vandoorne, et al., Experimental demonstration of reservoir computing on a silicon photonics chip, Nat. Commun. 5 (2014) 3541.
[28] P. Antonik, et al., Human action recognition with a large-scale brain-inspired photonic computer, Nat. Mach. Intell. 1 (2019) 530–537.
[29] Z. Zhou, B. Yin, J. Michel, On-chip light sources for silicon photonics, Light Sci. Appl. 4 (11) (2015), e358.
[30] B.M. Rodlin, et al., Hybrid integration of silicon photonics circuits and InP lasers by photonic wire bonding, Optica 5 (7) (2018) 876.
[31] S. Chen, et al., Electrically pumped continuous-wave III-V quantum dot lasers monolithically grown on silicon, in: 25th International Semiconductor Laser Conference (ISLC), 12–15 September 2016, Kobe, Japan, 2016.
[32] T. Alexoudi, G.T. Kanellos, N. Pleros, Optical RAM and integrated optical memories: a survey, Light Sci. Appl. 9 (1) (2020), 91.
[33] S. Pitris, et al., WDM-enabled optical RAM at 5 Gb/s using a monolithic InP flip-flop Chip, IEEE Photon. J. 8 (2) (2016) 1–7.
[34] C. Ríos, et al., Integrated all-photonic non-volatile multi-level memory, Nat. Photonics 9 (2015) 725–732.
[35] J. Bueno, et al., Reinforcement learning in a large-scale photonic recurrent neural network, Optica 5 (2018) 756–760.
[36] C. Demirkiran, et al., An Electro-photonic System for Accelerating Deep Neural Networks, arXiv, 2021.
[37] J. Wu, et al., Analog optical computing for artificial intelligence, Engineering 10 (2022) 133–145.
[38] T. Zhou, et al., In situ optical backpropagation training of diffractive optical neural networks, Photon. Res. 8 (2020) 940–953.
[39] T. Hughes, et al., Training of photonic neural networks through in situ backpropagation and gradient measurement, Optica 5 (2018) 864–871.

[40] L.G. Wright, Deep physical neural networks trained with backpropagation, Nature 601 (2022) 549–555.
[41] J. Launay, et al., Direct feedback alignment scales to modern deep learning tasks and architectures, Proc. Adv. Neural Inf. Process. Syst. 33 (2020) 9346–9360.
[42] M.J. Filipovich, et al., Monolithic Silicon Photonic Architecture for Training Deep Neural Networks with Direct Feedback Alignment, arXiv, 2021. 2111.06862v1.
[43] S. Zarei, Integrated photonic neural network based on silicon metalines, Opt. Express 28 (24) (2020).

CHAPTER 10

Machine learning in nanomaterial electron microscopy data analysis

Lehan Yao[a] **and Qian Chen**[a,b,c,d]

[a]Department of Materials Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States
[b]Department of Chemistry, University of Illinois at Urbana-Champaign, Urbana, IL, United States
[c]Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL, United States
[d]Materials Research Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, United States

1. Introduction

1.1 High-throughput analysis of nanomaterial microscopy data

The morphology of nanomaterials plays important roles in determining their macroscopic properties and functions. While traditional characterization techniques such as small-angle X-ray scattering probes the ensemble-averaged material feature length scale [1], they fail to provide information on structural heterogeneity. The real-space imaging techniques including scanning/transmission electron microscopy (S/TEM), scanning electron microscopy (SEM), and atomic force microscopy (AFM) can directly visualize the nanomaterial structures, providing the most intuitive information. However, real-space images focus on a local area instead of characterizing the whole sample, and thus usually work as a qualitative and supplementary technique in nanomaterial research. To comprehensively characterize the whole sample, one should take multiple microscope images from different locations and extract the information of interest in a quantitative and efficient way. The recently developed real-time imaging techniques, for example, liquid-phase TEM (LP-TEM), enable the ability to study nanomorphology change over-time [2,3] and nanoparticle diffusion and self-assembly [4,5]. Such technique generates videos, or stacks of images, which usually contains hundreds of frames and thus also requires efficient methods to locate and extract the features. Unlike the spectroscopy data, a real-space image is high dimensional, and the features inside are variant to translation, rotation, and scaling, which makes it inapplicable to treat images as a feature vector, retrieving information from fixed positions or fitting them into simple statistical algorithms. The quantitative analysis of nanomaterial microscope images usually starts from feature recognition, either by hand annotation or image processing algorithms (e.g., filtering and thresholding). While hand annotation by expert researchers is believed to be accurate, it only works on a small volume of data due to its low efficiency. On the other hand, most conventional image processing algorithms are less "smart," which struggles to find various features of interest buried under the complicated contrast profile

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00010-X
Copyright © 2023 Elsevier Inc.
All rights reserved.

and inevitable noises in the images. Machine learning (ML) is a fast-developing area in computer science in the past decade and a lot of ML models are proven to be capable for classification, feature detection, segmentation, and trajectory tracking of real-life pictures. Efforts have also been put into applying, adapting, and designing ML models for scientific image data and show excellent performance in analyzing biomedical and optical microscope images. In this chapter, the recent progresses on ML applications in processing and analyzing microscopy data of nanomaterials will be reviewed with a wide range of different imaging techniques, data type (images, videos, and tomographs), and ML models covered.

1.2 Machine learning models relevant to microscopy data analysis

A rough classification of ML models includes supervised and unsupervised learning, generative learning, and reinforcement learning (RL). The major difference between supervised and unsupervised learning is the requirement of training dataset. Training dataset is a set of exampled input and output pairs, based on which the ML models can optimize its weights in order to make predictions on new inputs. Supervised learning has its wide applications in image processing and analysis, including image denoising [6], deconvolution, segmentation, object detection, classification, and tomography reconstructions, which will be mostly covered in this section. The unsupervised learning exploits the inner distribution of the existing data itself to make the predictions on its clustering behaviors, and thus does not require additional training dataset. In material characterization, the unsupervised learning can automatically sort or group morphologies [7], lattices structures [8], spectra [9], and diffraction patterns [10–12]. Unlike the supervised and unsupervised learning (also regarded as predictive learning), the generative learning generates previously nonexisting data which captures the important features and visually resembles the existing dataset. Such learning techniques can potentially help in designing novel material structure with certain functions [13,14], predicting material morphology which accelerates physical property investigation [15], fabricating training datasets for supervised learning [16], and helping the tomography reconstruction [17]. Lastly, RL trains a decision maker—the agent to interact with the environment. RL algorithms are particularly good at interactive operations with information feedback, for examples, optimizing a chemical reaction [18], nanofabrication [19], controlled self-assembly [20], and navigation of self-propelled particles [21].

As we previously mentioned, images, if viewed as feature vectors, have very high dimensionality, strong spatial correlations, and are translational, rotational, and scaling variant. Those properties also cause difficulty in conventional ML techniques, for example, a fully connected neural network with the same size as the image can simply has too many neurons with high computational cost. As a result, the convolutional neural networks (CNNs) are usually chosen for processing image data in ML. During convolution,

small matrices with few weights, also known as kernel, slide through the input image to give the output to the next layer. The convolution can provide certain degrees of translational equivariance and save a lot of computational cost as compared to fully connected neural networks. Meanwhile, the pooling layers can also enable the CNNs to detect features at different length scales. CNN models such as LeNet [22], AlexNet [23], VGGNet [24], and ResNet [25], are designed for image classification, which means given an image input, the model will label the image with one or more logical or categorical labels. CNNs with symmetric encoder-decoder architectures such as fully convolutional network (FCN) [26] and U-Net [27] can make pixelwise prediction in an image-to-image manner, which is capable for noise reduction and semantic segmentation of one or more species. The limitation of semantic segmentation exists, in particular, in nanomaterials detections. Since the semantic segmentation directly labels the meaning of each pixel, it needs to be accompanied by postprocessing algorithms to identify the individual particles, which can be challenging with particles touching, overlaying, and aggregating with each other. Other CNNs can directly detect the objects by proposing and regressing the bounding boxes of the objects of interests in an image, examples of those CNNs can be region-based convolutional neural networks (R-CNN) [28], Fast R-CNN [29], Faster R-CNN [30], YOLO [31], and RetinaNet [32]. Such models localize the bounding boxes of objects without specifying their orientations and contours, which is segmentation-free but less applicable when researchers are interested in characterizing the morphology of nanomaterials. Combining semantic segmentation and object detection gives rise to more complicated models such as Mask R-CNN [33] which simultaneously gives the objects and their contours known as instance segmentation. Some aforementioned CNN architectures such as ResNet [34] and U-Net [35] can also be implemented in 3D to help the video recognition and tomography segmentation. Fully exploiting the temporal information from microscopy videos requires corelating the neighboring frames. Recurrent neural networks for example long short-term memory (LSTM) can selectively "memorize" or "forget" the information appears in the previous frames to deal with an entire sequence of data and has its application in particle trajectory classification [36] and object tracking [37].

Benefited from the real-space imaging capability of microscopies, a copious amount of individual microscopic object samples can be collected to understand the heterogeneity of nanomaterials with unknown morphologies. Then unsupervised ML models such as *k*-means, Gaussian mixture model (GMM), and density-based spatial clustering of applications with noise (DBSCAN) can classify them into groups, averaging and simplifying the morphologies with high similarity while highlighting heterogeneity across the whole sample [7,38,39]. Both *k*-means and GMM attempt to assign each data point (each sample) to a cluster center in the feature space. *K*-means minimizes the distance between all samples and its cluster center while GMM maximizes the likelihood of the samples belonging to its cluster center by assuming the overall datapoint distribution is a mixture

of several multivariant Gaussian distributions. DBSCAN is a density-based clustering algorithm to group the datapoints in high density regions together and regard the low-density regions as noises. Comparing with the *k*-means and GMM, DBSCAN does not require one to specify the number of clusters and is more applicable to distributions with arbitrary shapes. The inputs of such unsupervised ML techniques are usually certain feature descriptors (e.g., particle diameter, aspect ratio, or projection area) extracted from the microscopic characterizations. Since ML techniques always rely on a huge amount of data, collecting the shape descriptors from the raw microscopy images by automated and high-throughput algorithms are beneficial comparing with manual measurement. In some studies, the spectrum, diffraction pattern, or shape contours can also be directly used as the input of the unsupervised classification algorithms. One complication here is, different from the short feature vectors made up of few scalar descriptor values, those data type usually have high dimensionality (>10) and, due to the well-known "curse of dimensionality" [40,41], are hard to be classified. To solve this issue, dimension reduction techniques (which are also considered as unsupervised learnings) are applied prior to the classification. Commonly, top 2 or 3 dimensions after the reduction are either directly visualized to show the sample distribution or serve the as the input of the unsupervised classifications. Such dimension reduction algorithms include principal component analysis (PCA) [42], t-distributed stochastic neighbor embedding (t-SNE) [43], and uniform manifold approximation and projection (UMAP) [44].

Variational autoencoder (VAE) [45] and generative adversarial networks (GAN) [46] with its variations are two common models in the generative ML. The GAN consists of a generator network and a discriminator network. The generator network takes randomly initialized input vectors to make up images, while the discriminator network judges whether the images are synthesized by the generator or real images in the training set. In this special training process, the generator will be optimized to "fool" the discriminator and discriminator network will be trained to be smarter to detect the fake images. Eventually, with this adversarial training process, the generator can learn to synthesize previously nonexisting images that visually resemble the images in training set.

1.3 Training datasets in supervised learning of nanomaterial microscopy images

For supervised ML models, before making any reasonable predictions, they have to be trained on a labeled training dataset that covers the features one wants to predict, and the volume of the training dataset usually needs to be relatively large (at least hundreds of images) to cover a whole range of sample variations. Although training datasets of common life object photos and biomedical images with corresponding labels are widely accessible to the ML community [47–49], given the vast and fast-expanding ranges of different nanomaterials, it is still difficult to train models that are widely applicable to one's arbitrary and customized experimental images [50]. While manually labeling an

experimental image library is a conventional approach of crafting the training dataset, paradox exists in the meaning of such supervised ML in image analysis. Manually labeling a large number of images of specific materials compromises the efficiency, which is the initial reason of applying ML methods.

To fully exploit the efficiency of manual annotation, training image augmentation can be applied to expand the size of training datasets [51,52]. The augmentation allows one to sample and manually label a relatively small portion of the experimental images that are going to be predicted. Then random combinations of transformations including translation, rotation, shearing, zooming, flipping, brightness change, and so on are applied to both experimental images and the corresponding labels (if the labels are also images) to give previously nonexisting but reasonable training samples. The augmented training set increases the sample volume and variations by mimicking the random orientation, shape variation, magnification and brightness, contrast fluctuations that happen in the real microscopy imaging process, and consequently makes model robust against them. Another efficient way of generating the training set is through image simulation. By considering the contrast formation mechanism and sample randomness in the real microscopic images, image simulation algorithms produce training images that visually resemble the experimental images, which, in principle, can provide training datasets with unlimited sizes. Such simulation-based training has been applied to train neural networks that identify atoms and nanoparticles from scanning transmission electron microscopy (STEM) [53,54], TEM [55], and helium ion microscopy (HIM) [56] images. Good simulations reproduce the material pattern and contrast profile under the experimental condition while leaving randomness on the contours, orientations, positions, and other information needs to be extracted. Otherwise, if the simulation does not resemble the experimental images, the prediction performance would be limited when it comes to predicting the real images regardless how high is the accuracy on training dataset.

2. ML in 2D microscopy image analysis

2.1 Classification and regression of nanomaterial microscopy images

One of the relatively simple tasks in 2D image analysis is classification, which means given an image input, one interests in assigning the image with one or more logical or categorical labels. Through CNNs, a collection of microscopic images of nanomaterials can be automatically classified into subcategories with similar features in a high-throughput manner. To give an example, Luo et al. [57] combine the supervised CNN with unsupervised embedding algorithms to classify the TEM images of carbon nanostructures into several predefined morphologies (Fig. 1A), which can potentially provide information to help the risk assessment for the exposure to these carbon materials. In this work, the authors innovatively extract and cluster hypercolumns from layers in a VGG-16 model representing information at various length scales and compare their results with the

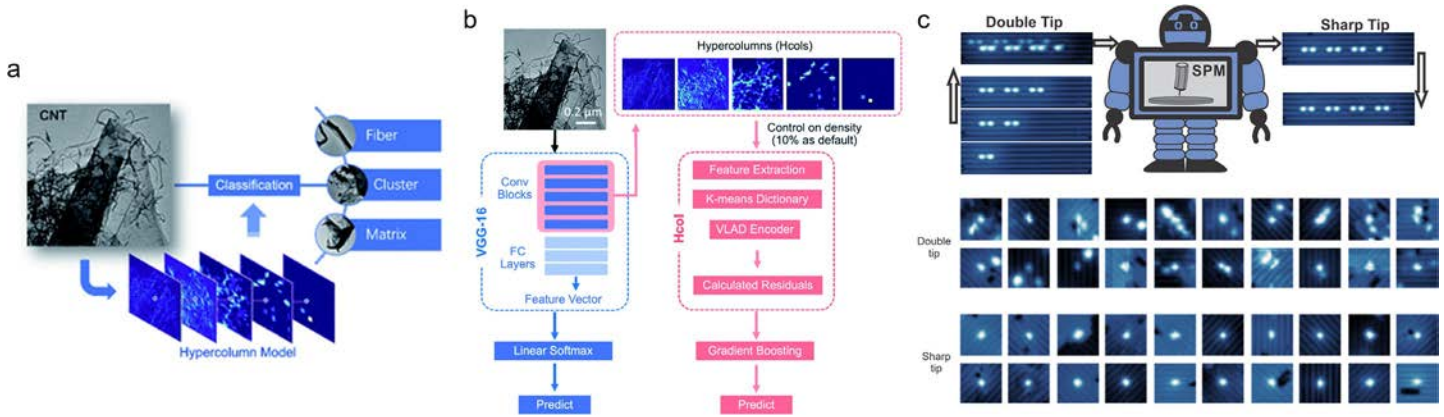


Fig. 1 Studies of using ML in image classification. (A) A schematic of the carbon nanostructure's image classification workflow. (B) Carbon nanostructure's image classification models: *blue arrows* indicate the conventional VGG-16 model classification; *red arrows* indicate the hypercolumn model. In the hypercolumn model, images were imported to the pretrained CNN architecture and followed by the hypercolumn feature extraction. A *k*-means library was then used as a visual feature dictionary and introduced to a vector of locally aggregated descriptor encoder to compute the residuals. Finally, the classifier was trained with a boosting cycle to make predictions. (C) A schematic showing the CNN-based SPM probe quality evaluating workflow and example training images of double tip and sharp tip. *((B) Adapted from Q. Luo, E.A. Holm, C. Wang, A transfer learning approach for improved classification of carbon nanomaterials from TEM images, Nanoscale Adv. 3 (2021) 206–13, https://doi.org/10.1039/D0NA00634C with permission from The Royal Society of Chemistry. (C) Adapted with permission from M. Rashidi, R.A. Wolkow, Autonomous scanning probe microscopy in situ tip conditioning through machine learning, ACS Nano 12 (2018) 5185–9, https://doi.org/10.1021/acsnano.8b02208. Copyright 2018 American Chemical Society.)*

performance of the original VGG-16 model (Fig. 1B). Their novel model yields 90.9% and 84.5% accuracies on 4- and 8-class classifications, respectively, which is higher than the conventional CNN model. Meanwhile, the authors also found that by applying data augmentation to the minority classes, the prediction accuracies in all categories can be improved and the accuracy variations among different classes can be reduced. This is because with the training data imbalance, the model will be less penalized from misclassifying the minority groups, and by rebalancing the training data classes by data augmentation, this issue can be solved. Such image classification tasks can also be exploited by being integrated in other processes. Rashidi et al. [58] utilize CNN to help the sharp tip restoration in scanning probe microscopy (SPM)-based nanofabrication. Trained on SPM images of manually labeled, isolated silicon dangling bonds, the CNN can keep making binary classifications on whether the SPM tip is sharp or doubled in the tip conditioning process until a sharp tip is detected (Fig. 1C). This method can serve as a module and automate the atomic-scale nanofabrication as well as be applied to other material systems and nanoscale imaging techniques.

Other than predicting logical or categorical values out from the image input, CNNs are also able to regress numerical values given the proper choice of the loss function. For example, Hu et al. [59] train model to predict the 3D orientations of the shape-anisotropic gold nanostars. The nanostars have low structural symmetry and thus exhibit different patterns at different 3D orientations under the differential interference contrast (DIC) microscopy. With taking the multispectral DIC patterns of nanostars at three wavelengths, their trained neural network can predict the in-plane and out-plane angles (φ and θ) of the nanostars with average prediction errors below 20°. Their results demonstrate the application of CNNs in fast and automated orientation tracking for studying 3D rotational dynamics of nanoparticles.

2.2 Segmentation of nanomaterial microscopy images

More quantitative analysis of nanomaterials such as the synthesis yield analysis, lattice structure characterization, nanomorphology extraction, and studies on nanoparticle diffusion and self-assembly requires one to quantitatively exploit the spatial information from microscopy images. The conventional approach of such studies usually starts from the semantic segmentation, which means to cut the microscopy image into several domains or areas with different meanings, for examples, different atoms, elements, lattice structures, nanoparticles, and the background. Then in the following analysis, the locations, contours, as well as particle counts can be easily extracted from the area boundaries. Non-ML methods typically employ thresholding, where the segmentation is achieved by setting intensity thresholds and assigning certain meanings to pixels within the threshold range. Although efforts in preprocessing including denoising, contrast enhancements, and creating sliding windows to implement the threshold locally can help the

performance, incorrect segmentations can often be obtained from thresholding methods because of the existence of high noise, blurriness, complicated material contrast profile, and unwanted impurity species.

In 2014 and 2015, the invention of FCN [26] and U-Net [27] has revolutionized the semantic segmentation process. Given an image as the input, such CNN models can make pixelwise prediction on the probability of each pixel in the original image belonging to different species (e.g., in the simplest case, foreground objects vs the background). This is benefitted from their special encoder-decoder architectures, where the encoder part allows the detection of features on multiple length scales and the decoder reconstructs the information back in the original spatial domain, and meanwhile, the direct connection between the layers in encoder and decoder can help to recover the full spatial resolution in the output [60]. Unlike the conventional threshold-based segmentation, the FCN and U-Net not only consider the absolute intensity values in the image during the prediction, but also their relative differences, their surrounding environments, and the model's previous experiences. These factors combined exactly mimic the decision-making process that happens in the minds of human experts during their manual image segmentation.

With U-Net first demonstrates its application in the segmentation of optical microscopy images of cells, CNNs with similar architectures have soon been widely applied to the microscopy-based nanomaterial characterizations. Ziatdinov et al. [53] train FCNs to predict the atomic positions as well as the dopants and defects in the STEM images of 2D lattices of graphene and $MoSe_2$ (Fig. 2A and B). In their study, several interesting phenomena have been observed. The models trained on graphene lattice can predict the atomic positions in the $MoSe_2$ lattice as well. The similar lattice structures and drastically different atomic compositions in those two materials indicate that the FCN has the potential to learn from the lattice structures instead of the exact intensity profile of individual atoms. Another important finding is that by being trained on simulated STEM images of lattices with single vacancies only, the model can spot more complicated vacancies associated with more than one missing atoms. Such observations validate the generalizability of the training process and the versatility of the trained models—one does not have to explicitly include every possible physical scenario during the training data selection or simulation, and the CNN models have the ability to learn the general features and extrapolate its knowledge into more complicated situations. In this and the following works from Ziatdinov and co-workers [51,54], with the atomic positions accurately predicted by the CNNs from the STEM images of 2D materials such as graphene, $MoSe_2$, and WS_2, their dopant- and vacancy-related defect structures as well as their transformations over time can be captured and well correlated to the materials functionalities.

CNN models with similar architectures also help the segmentation of nanoscale materials in various microscopies. For example, U-Net trained on the simulated TEM images is proven to make accurate nanoparticle segmentation on LP-TEM videos with low

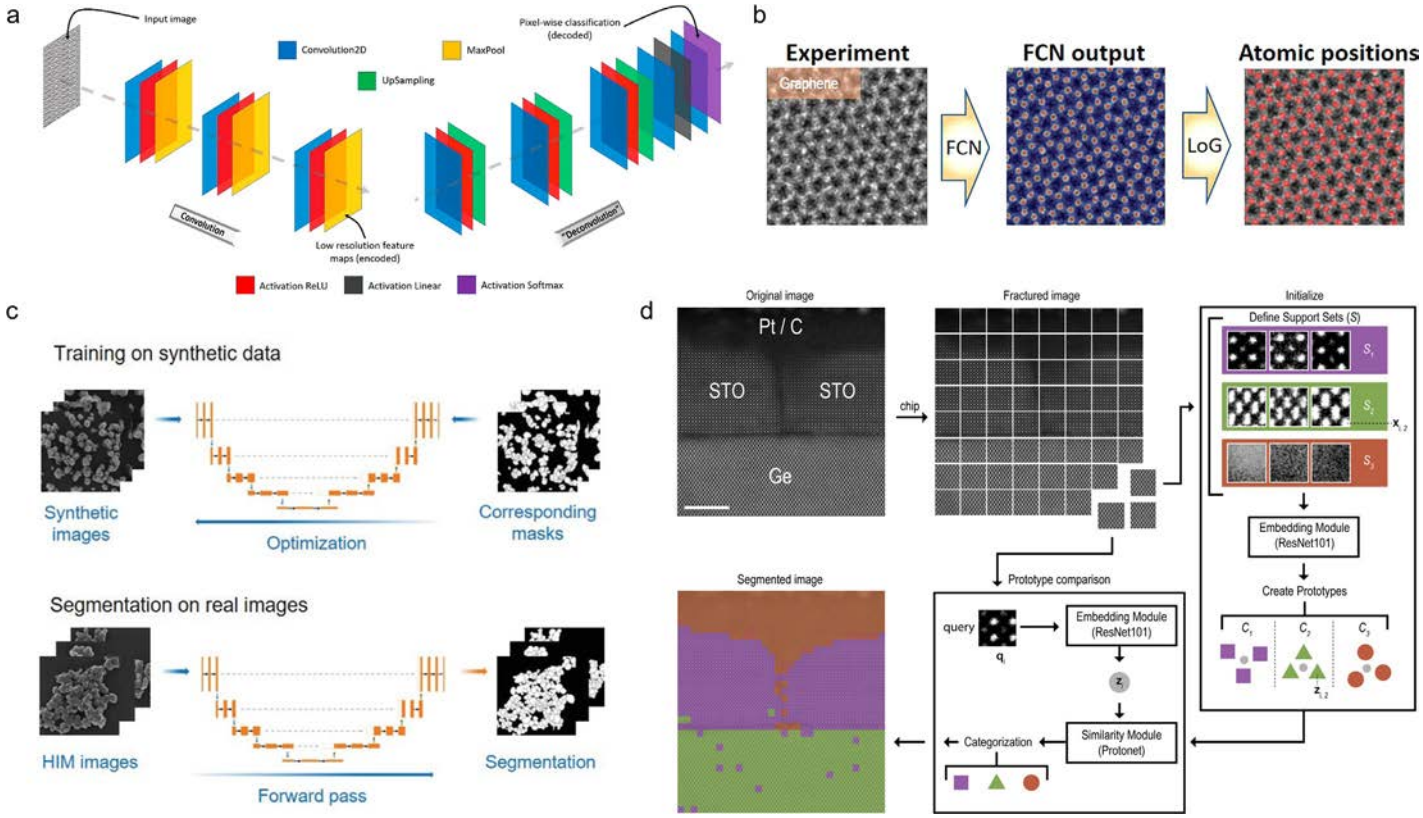


Fig. 2 Studies using ML in semantic segmentation. (A) Schematic architecture of an FCN that has an encoder-decoder type of structure. The final softmax layer outputs a pixelwise classification for atomic species and/or defects from STEM images. (B) Illustration of how atomic positions are obtained from the raw STEM image of graphene using FCN and Laplacian of Gaussian blob detection method. The FCN maps and atomic positions are overlaid on the experimental images. (C) HIM nanoparticle images and corresponding labels synthesized by render software are used to train a U-Net model for automated particle segmentation and subsequent quantitative, statistical assessment. After training, the model which predicts the most accurate segmentation on a real validation image is selected for further processing. (D) In the few-shot learning model, the raw STEM image is broken into several smaller chips and a few user-defined chips are used to represent desired segmentation classes in the support set. Each chip then acts as a query and is compared against the support set and get categorized, yielding the segmented image. Scale bar: 5 nm. *((B) Adapted with permission from M. Ziatdinov, O. Dyck, A. Maksov, X. Li, X. Sang, K. Xiao, et al., Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations, ACS Nano 11 (2017) 12742–52, https://doi.org/10.1021/acsnano.7b07504. Copyright 2017 American Chemical Society. (C) Adapted with permission from L. Mill, D. Wolff, N. Gerrits, P. Philipp, L. Kling, F. Vollnhals, et al., Synthetic image rendering solves annotation problem in deep learning nanoparticle segmentation, Small Methods 5 (2021) 2100223, https://doi.org/10.1002/smtd.202100223. (D) Adapted with permission from S. Akers, E. Kautz, A. Trevino-Gavito, M. Olszta, B.E. Matthews, L. Wang, et al., Rapid and flexible segmentation of electron microscopy data using few-shot machine learning, Npj Comput. Mater. 7 (2021) 1–9, https://doi.org/10.1038/s41524-021-00652-z.)*

signal-to-noise ratio and fluctuating feature intensities. Combined with the temporal information obtained from the video, the high-quality segmentation further helps the individualization and trajectory tracking of single nanoparticles, which finally leads to the revelation of the nanoparticle diffusion, self-assembly, and morphological transformation in their native, liquid environment [55]. Similarly, Mill et al. [56] train U-Net to segment SiO_2 and TiO_2 nanoparticles in HIM images (Fig. 2C). Their proposed rendering software-based synthetic data generation pipeline allows the trained model to segment partially obscured nanoparticles with accuracies comparable to human experts. Such integration of semiautomated training dataset generation and ML prediction requires less human input and thus makes the shape characterization of nanoparticles more efficient, which can potentially benefit the nano-toxicology or nano-medicine studies. Ziatdinov and co-workers [61,62] have also applied similar CNN models to segment the protein nanoparticles and nanofibers from AFM and SEM images. The locations and orientations extracted from the segmentation maps predicted by the models not only enable the study of protein self-assembly behavior, but also indicate that the CNN-based image segmentation can be robustly applied to both inorganic and organic nanomaterials imaged by a wide range of different microscopes.

Other than the image-to-image neural networks such as FCN and U-Net mentioned above, efforts in other types of MLs have also been spent to help the nanomaterial segmentation challenge. One example is that Lee et al. [39] implement a genetic algorithm to optimize the operation sequences and parameters selection during the conventional TEM image segmentation workflow. Such parameters include the pretreatment order, filter kernel size, edge detection, and so on. Although having potentials in making excellent segmentations, those parameters combined together can form into a high dimensional space which is difficult for human researchers to optimize. On the other hand, the genetic algorithm based on the natural selection process in species evolution, once trained on example images, can efficiently find the best combinations of the parameters that give good segmentation results. Their final model is able to efficiently and accurately extract the morphological properties from gold nanorods that densely pack under TEM. The applications of such model can be found in synthesis characterization and the evaluation and construction of nanoparticle big data. Akers et al. [50] develop a few-shot learning model to differentiate the phases, defects, and microstructural features in the STEM images of metal oxide heterostructures, thin films, and nanoparticles (Fig. 2D). In this study, one high-resolution STEM image is first divided into grids of subimages. Then a few subimages from each class of different semantic meanings are picked up and manually labeled as the "supporting set," while the rest majority of the subimages remains undistributed. Next, all subimages are embedded by a neural network (ResNet101 in this case) to give the transformed representations of the subimages in a different feature space and the distances between the unlabeled subimages and the center of each support set will be calculated and compared to give the labels for those unlabeled subimages. Finally, all

classified subimages are stitched together to give the semantic segmentation map of the original high-resolution image. Comparing with the image-to-image CNN models mentioned previously, the few-shot learning requires much smaller training (support) sets that usually range from one to a few images per class, and consequently can be applied more rapidly and flexibly to new materials and new classification tasks.

2.3 Object detection of nanomaterial microscopy images

The models and methods mentioned above are dedicated to achieving the semantic segmentation task in the microscopy images. However, the semantic segmentation is not always sufficient for detecting the individual nanomaterials such as nanoparticles, polymer micelles, or cells. When the particles closely touch each other, partially obscure each other in SEM view, or overlay with each other in TEM projection, the semantic segmentation, which simply classifies each pixel, tends to fuse the nanoparticles together instead of detecting each nanoparticle as an instance. In some scenarios such as the yield analysis in nanoparticle synthesis, people only care about the prediction precision, which is defined as among the nanoparticle retrieved by the algorithm, the number of nanoparticles that are correctly predicted; false-positive detections like nanoparticle dimers or aggregations can be easily filtered out by their abnormal shape or larger size [7]. To the contrast, in other scenarios, e.g., nanoparticle self-assembly study [5,63], the false-negative (missed) detections also matter because they might bring unphysical vacancies in the self-assembly structures. As a result, a good algorithm should not only be responsible for capturing correct particles, but also not miss any particles. Postprocessing algorithms including the watershed transformation [64], ultimate erosion for convex sets [65], and bounded erosion-fast radial symmetry [66] can be applied after the segmentation to force the connected nanoparticle apart. However, they only exploit the geometrical information along the segmented boundaries without referring to the intensity profiles in the original image, and usually work better for convex shapes with partial and simple overlaying, while the concave polygons and complex overlaying can frequently exist in the metal or metal oxide nanoparticle self-assemblies.

There are ML models capable for object detection as well, which usually means to directly locate and draw the bounding box for each instance of the object in an image. Although not necessarily giving the contours of objects as in the pixelwise classification case, object detection models can naturally capture the individual nanoparticles from the raw image, without involving any postprocessing steps to force the connected particles separated. The region-based CNNs (e.g., R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and RetinaNet) are designed for such object detections in real-life photos and can be applied to the microscopy images analysis as well. Taking the more advanced Faster R-CNN as an example, firstly, a region proposal network (RPN) is responsible for proposing potential region-of-interests (ROIs) where the objects roughly lie in, and then

proposal bounding boxes will be generated around the ROIs. In the next step, the proposed bounding boxes will be input into the classifier and regressor network which judges the exact class of each bounding box and refine their positions to precisely inscribe the detected objects, respectively. Shen et al. [67] employ the Faster R-CNN to locate defect clusters in irradiated steels (Fig. 3A). Trained on manually labeled and augmented experimental STEM images of FeCrAl alloys, their faster R-CNN model can localize and distinguish dislocation loops with four different types of morphology. With the detection accuracy within the uncertainty of human researchers, this model is also more efficient, reproducible, and scalable, which can potentially replace the human efforts in future STEM image analysis. Li et al. [52] utilize a similar Faster R-CNN to detect and differentiate the supramolecular assembly structures with opposite handedness made up from star-shaped hexadimethylphenylbenzene and fluorine-substituted hexadimethylphenylbenzene molecules imaged by scanning tunneling microscopy (STM) (Fig. 3B). Besides their high recognition rate over 90% and their training data augmented from one single-labeled image, several interesting conclusions are drawn from their observations: (1) The model trained on high-resolution STM images remains robust when tested against low-resolution images. (2) The model has good performance despite the difference in the spatial arrangements of the objects in training dataset and testing dataset. (3) High-quality training data with more obvious difference between the L- and R-handed species leads to better performance while training data with little distinction between species leads to low performance. Their application of object detection on STM has the potential in accelerating the routine tasks in image analysis, material database development, and advanced materials discovery. In other's studies, object detection CNNs have also been used for single particle picking in cryo-EM reconstruction [68,69].

Although capable of detecting instances, the object detection methods discussed above can only locate the object bounding boxes without delineating their contours, which is insufficient to characterize nanomorphology, centroid coordinate, and nanoparticle orientation. Adding another pixelwise classification CNN after the proposed bounding boxes in the region-based neural networks can equip the model with the ability of "instance segmentation" and this gives rise to the mask R-CNN model. In instance segmentation, a mask delineating the object boundary will be generated for every detected instance, instead of for the whole image in semantic segmentation, which can achieve the shape and orientation characterization for overlaying and aggregated nanoparticles. Such model and its variant have been used in the nanoparticle size characterization from TEM and SEM images [70–73] with good performance in separating overlaying nanoparticle (Fig. 3C), and also bears potential in analyzing dynamic processes such as nanoparticle diffusion and self-assembly from the electron microscopy video data. However, mask R-CNN has a complicated thus computationally expensive architecture and can still suffer from missing detections during the nonmaximum suppression (NMS) step when the overlaying area between adject bounding boxes is large [74]. To solve these

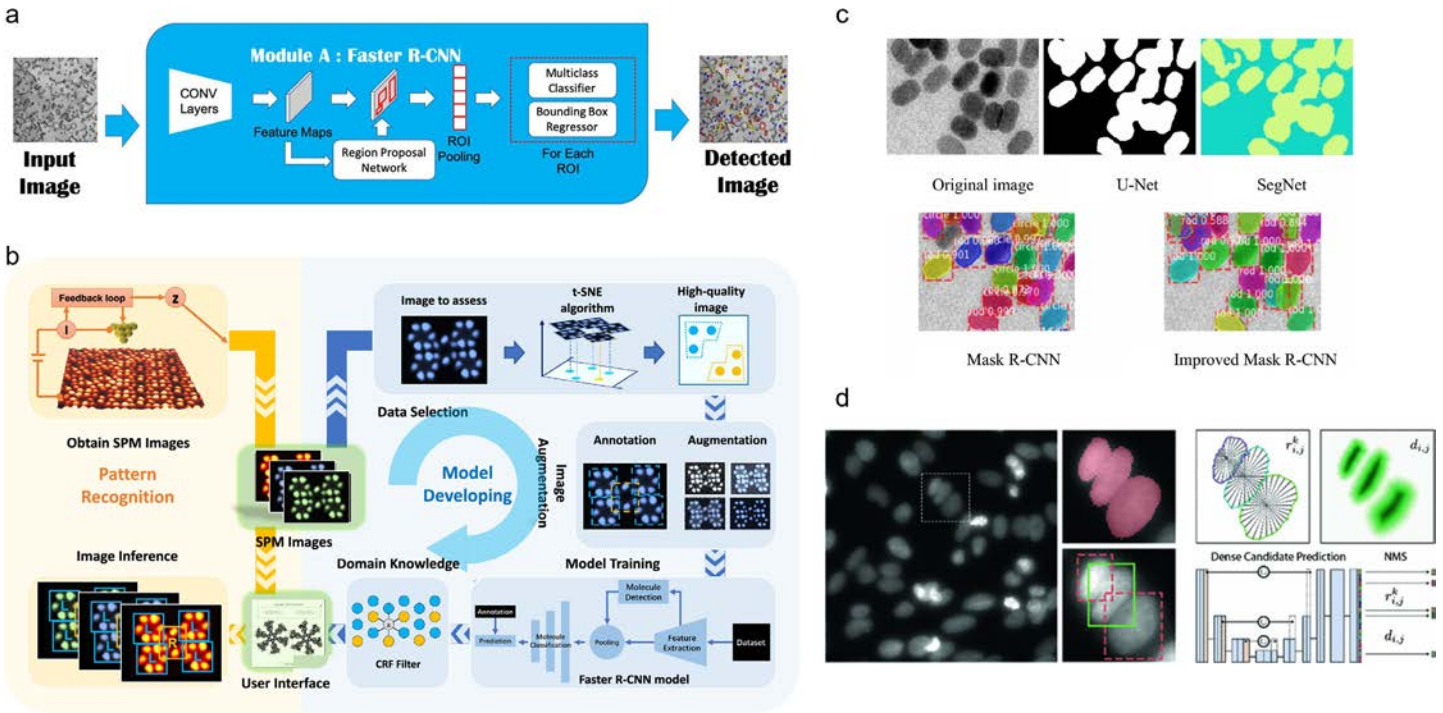


Fig. 3 Studies using ML in object detection and instance segmentation. (A) Schematic of the flow chart of Faster R-CNN-based automated defect detection from STEM images of irradiated ferritic steels. Input micrograph images first go through the Faster R-CNN detector to identify the loop locations and bounding boxes. Then for each identified bounding box, geometry fitting algorithms are called to determine the defect shape and size. (B) Flow diagram illustrating the automated chiral molecule detection and identification from original SPM images. First, the experimental STM images are obtained. Then image quality is rated by a t-SNE unsupervised learning method. The data with the desired quality are augmented to generate a training dataset for the core prediction model. Afterward, an optional conditional random field filter-based domain knowledge incorporation module can be utilized to enhance performance for nonideal STM images. Lastly, the trained model is utilized for the image inference tasks for automated pattern analysis of the SPM images. (C) Comparison of nanorod TEM image segmentation results obtained by semantic segmentation (U-Net and SegNet) and instance segmentation (Mask R-CNN and improved Mask R-CNN). The improved Mask R-CNN can have better performance on weak-edge nanoparticles and adhesion nanoparticles. (D) Segmentation errors of merging cells in semantic segmentation methods and suppression of valid cell instances in bounding-box detection methods. On the other hand, StarDist method predicts object probabilities and *star*-convex polygons parameterized by the radial distances with U-Net and then selects the final instances via NMS, which gives better segmentation results. *((A) Adapted from M. Shen, G. Li, D. Wu, Y. Liu, J.R.C. Greaves, W. Hao, et al., Multi defect detection and analysis of electron microscopy images with deep learning, Comput. Mater. Sci. 199 (2021) 110576, https://doi.org/10.1016/j.commatsci.2021.110576 with permission from Elsevier. (B) Adapted with permission from J. Li, M. Telychko, J. Yin, Y. Zhu, G. Li, S. Song, et al., Machine vision automated chiral molecule detection and classification in molecular imaging, J. Am. Chem. Soc. 143 (2021) 10177–88, https://doi.org/10.1021/jacs.1c03091. Copyright 2021 American Chemical Society. (C) Adapted with permission from F. Zhang, D. Zhao, Z. Xiao, J. Wu, L. Geng, W. Wang, et al., Rodlike nanoparticle parameter measurement method based on improved mask R-CNN segmentation, SIViP 15 (2021) 579–87, https://doi.org/10.1007/s11760-020-01779-0. Copyright 2020 Springer Nature. (D) Adapted with permission from U. Schmidt, M. Weigert, C. Broaddus, G. Myers, Cell detection with star-convex polygons, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Med. Image Comput. Comput. Assist. Interv. – MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 265–73, https://doi.org/10.1007/978-3-030-00934-2_30. Copyright 2018 Springer Nature.)*

problems, the instance segmentation can also be achieved by bounding-box-free and light-weight algorithms such as StarDist [74]. Different from the bounding-box-proposals in the region-based CNNs, the StarDist directly regresses the pixel-to-boundary distance for every pixel inside the object at different radial directions (Fig. 3D). Meanwhile, another regression will be conducted to find the pixels near the objects' center. Next, particle boundaries will be reconstructed and proposed by those radial distances measured at pixels near the objects' center. In the last step, an NMS can remove the duplicated particle boundaries to give the final instance segmentation results. Such proposal-free algorithms have been tested and applied to cell segmentation in optical microscopy images [74,75], object detection in real-life photos [76], as well as nanoparticle characterization in TEM and SEM images [77,78].

3. ML in 3D tomography reconstruction and segmentation

The 3D morphology as well as the 3D element distribution of nanomaterials can be revealed by tomography techniques such as electron tomography with TEM, high-angle annular dark-field imaging (HAADF)-STEM, and STEM-energy-dispersive X-ray (EDX). 3D morphologies such as convexly shaped nanoparticles and polymeric membranes with rich inner voids can be critical for their optical [79], catalytic [80], and permeation [81] properties and could not be sufficiently characterized by simple 2D projections. In such tomography techniques, the sample will be tilted by rotating the sample holder and 2D projections will be collected as a function of tilting angles. Then those projections, which are also known as sinograms, undergo reconstruction algorithms to give the volumetric image representing the 3D geometry of the sample material. While the simplest analytic reconstruction algorithms (Fig. 4A) such as filtered-back projection (FBP) directly inverse the projection process to backout the original geometry, they are usually prone to the artifacts presented in the experimental image acquisition. For example, the tilting range of sample holder in electron microscopies is usually limited to $\pm 60°$ or $\pm 70°$, which is narrower than the ideal tilting range of $\pm 90°$ to collect all information. The limited tilting angle can result in missing sinograms and causes feature elongation in the final reconstruction, which is also regarded as the "missing wedge" effect and can potentially lead to serious misinterpretations [82]. On the other hand, to avoid the beam damage introduced to the fragile samples such as polymer membranes [81] or samples under EDX tomography with high-dose rate, the total number of tilting series is often limited and thus causes sparsely sampled sinograms [83], which leads to reconstruction at low resolution. Moreover, the signal-to-noise ratio (SNR) in EDX tomography can be very low due to the low probability of X-ray generation and low efficiency of their detection [84], which also impact the reconstruction quality. To address those issues, more advanced iterative reconstructions (Fig. 4A) such as simultaneous iterative reconstruction technique [85], model-based iterative reconstruction, and discrete algebraic

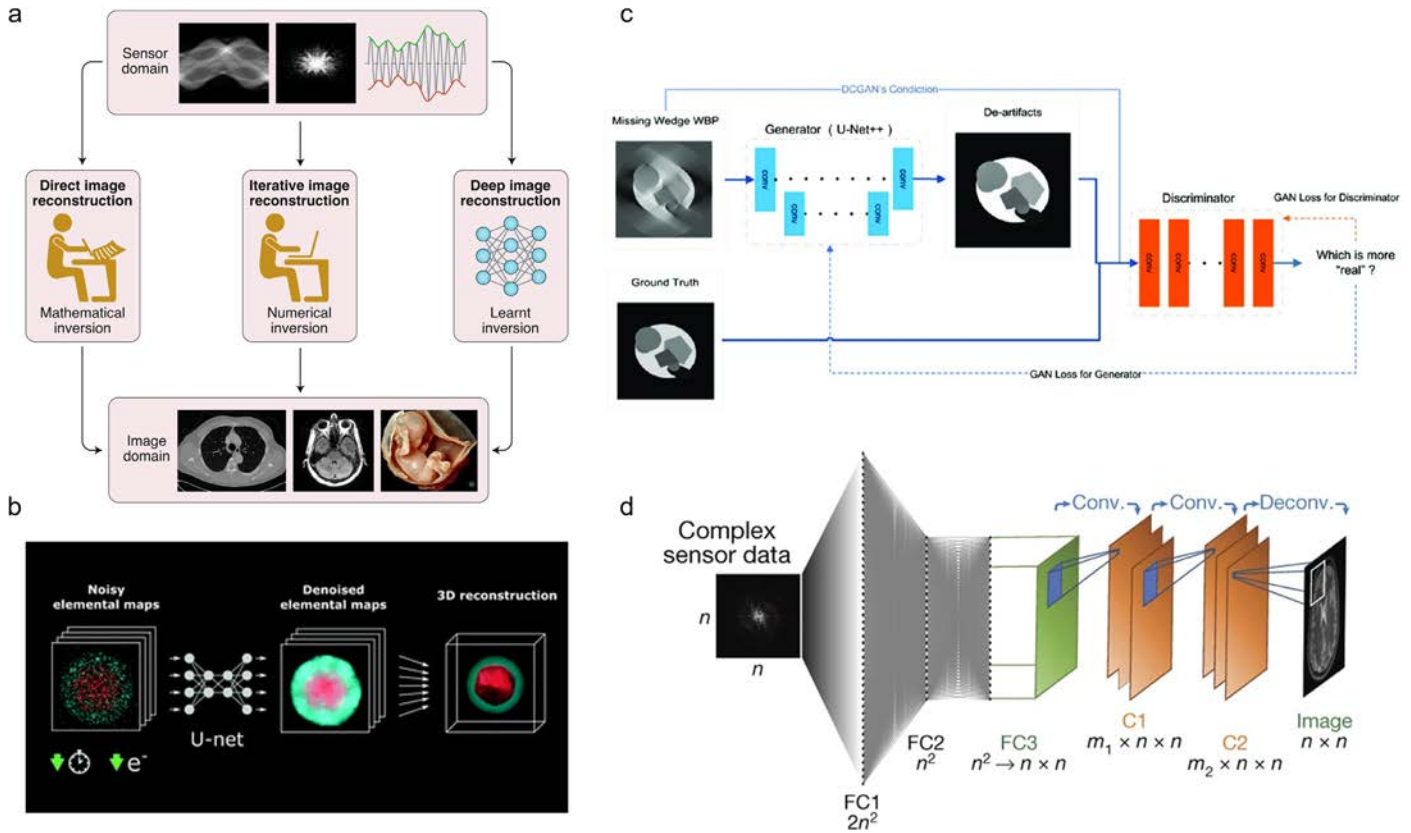


Fig. 4 Application of ML in tomography reconstruction. (A) Three types of tomographic reconstruction methods. In direct or analytic reconstruction, a mathematical inverse of the forward transform is used to compute images from sensor data. In iterative reconstruction, the current image estimate is iteratively updated such that its forward transform gradually approaches the sensor data. In deep tomographic reconstruction, the inversion does not need to rely on any explicit transform model but is learnt from representative big data. (B) A schematic showing using U-Net for elemental map denoising to address the problem of very high acquisition time and electron dose requirements for EDX tomography of nanoparticles. (C) The training pipeline of the IRDM model to predict information recovered and de-artifacted tomogram from FBP reconstructions. (D) A schematic showing AUTOMAP model for sensor data to reconstruction image domain transformation. The model is implemented with a deep neural network architecture composed of fully connected layers (FC1 to FC3), followed by convolutional layers (C1 and C2) that form a convolutional autoencoder. *((A) Adapted with permission from G. Wang, J.C. Ye, B. De Man, Deep learning for tomographic image reconstruction, Nat. Mach. Intell. 2 (2020) 737–48, https://doi.org/10.1038/s42256-020-00273-z. Copyright 2020 Springer Nature. (B) Adapted from A. Skorikov, W. Heyvaert, W. Albecht, D.M. Pelt, S. Bals, Deep learning-based denoising for improved dose efficiency in EDX tomography of nanoparticles, Nanoscale 13 (2021) 12242–9, https://doi.org/10.1039/D1NR03232A with permission from The Royal Society of Chemistry. (C) Adapted with permission from C. Wang, G. Ding, Y. Liu, H.L. Xin, 0.7 Å resolution electron tomography enabled by deep-learning-aided information recovery, Adv. Intell. Syst. 2 (2020) 2000152, https://doi.org/10.1002/aisy.202000152. (D) Adapted with permission from B. Zhu, J.Z. Liu, S.F. Cauley, B.R. Rosen, M.S. Rosen, Image reconstruction by domain-transform manifold learning, Nature 555 (2018) 487–92, https://doi.org/10.1038/nature25988. Copyright 2018 Springer Nature.)*

reconstruction technique [86] are proposed to compare the reconstructions with the experimental observations iteratively until the error vanishes and the solution converges [87]. While having enhanced performance comparing to the analytic algorithms, iterative reconstructions can potentially suffer from the feature smoothing and high computation costs [88]. Recently, other than analytic and iterative reconstruction, an increasing number of research shows that ML algorithms (Fig. 4A) can effectively help the tomography reconstruction in various applications from X-ray computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound in biomedical imaging to optical, X-ray, and electron tomography in material science [83,87,89].

Considering the transformation of the experimentally taken 2D sinograms to 3D volumetric image in x, y, z coordinates, the ML contribution in tomography reconstruction can be categorized as: (1) preprocessing of sinograms, (2) postprocessing of simply reconstructed tomograms, and (3) end-to-end reconstruction directly from sinogram to tomogram. To give an example of sinogram preprocessing, Yang et al. [89] train CNN with encoder-decoder architecture to enhance the SNR of the sinograms taken by transmission X-ray microscopy (TXM) at short exposure time. The CNN is trained on the X-ray projection series pairs taken at long and short exposure time and can map the noisy images to high-quality images which reduces the X-ray dose during imaging and prevents the sample from beam-induced damages. Their validation of the model on both synthetic datasets and real TXM measurements of mouse brain structures demonstrates the signal enhancement by at least a factor of 10 and the recovery of more structural information. In other studies [84], it has been shown that advantage exists in such sinogram preprocessing methods comparing to the reconstruction postprocessing because the noise in experimental sinograms can already lead to artifacts during the reconstruction and consequently leaves input corrupted for postprocessing. As a result, in this study, the simulated EDX tilting projections of various-shaped nanoparticles with randomizable geometric parameters and artificial noises are used to train a U-Net for elemental map denoising before the tomography reconstruction (Fig. 4B). While the trained U-Net shows better denoising performance than classical denoising approaches both quantitively and qualitatively on simulated EDX images, applying the denoised sinograms to reconstruct the experimentally obtained tilting projections yields adequate representation of the nanoparticle's shape and elemental distribution. Their application of U-Net in increasing the sinogram SNR has the potential to allow for reductions in EDX acquisition time and electron dose, which benefits the studies of beam-sensitive materials at high throughput. Meanwhile, in the work by Han et al. [83], with concerns on the fidelity of imaging procedure modeling in STEM-EDX tomography data simulation, an unsupervised ML workflow is proposed to bypass the lack of training data. In this workflow, a conventional reconstruction algorithm is first used to build a low-resolution reconstruction from the noisy and sparse sinograms obtained experimentally. Then in the next step, a neural work is used to generate high-resolution sinograms from the noisy reconstruction, filling the missing and under

sampled tilting angles. In their training process, by imposing the consistency between the original and enhanced sinograms, only the experimental sinogram but no matching ground truth is needed. Finally, the sinograms recovered by the neural network can be reconstructed by simple algorithms such as FBP to give the high-resolution tomograms. By using the gold nanoparticle and quantum dots as example systems, they show their training data-free, unsupervised learning algorithm can give high-quality 3D reconstructions that well correlate with the samples' optical characteristics and can be used broadly for comprehensive physicochemical analysis of nanoparticles.

For "postprocessing" type reconstruction, Wang et al. [17] train GAN to recover information and remove artifacts from the simply reconstructed tomograms with missing-wedge effects (Fig. 4C). In their GAN model, which is also known as information recovery and de-artifact model (IRDM), a U-Net++ is used as the generator and it takes the FBP reconstructions as the input and outputs the information recovered and de-artifacted tomograms. During the training process, a CNN discriminator is responsible for judging whether the tomogram is produced by the generator or from the ground truth library. Although the discriminator does not directly contribute to the reconstruction process, its derivative during the backpropagation is necessary to help the generator to give better reconstructions. The model is trained on publicly accessible datasets of biomedical, daily life, and phantom images and tested on the experimentally obtained atomic electron tomography with missing wedge up to 40°. Beside the superior performance of their IRDM comparing with the conventional FBP highlighted by the power spectrum, their research also provides some insights into the training dataset selection. For example, training dataset with high complexity can improve the robustness of IRDM against high missing wedge angles, and training datasets with features and missing wedge regions similar to the targeted sinograms are more appropriate to be selected. The U-Net type of encoder-decoder neural networks can also be implemented in 3D to directly segment or denoise volumetric images. An example of such model is the 3D U-Net for the automatic segmentation of confocal microscopic images [35]. But unlike the electron tomography, those studies focus more on the segmentation rather than the missing information recovery or artifact correction due to the difference in the imaging mechanisms. Recent efforts demonstrate that such 3D CNNs are also capable for volumetric image segmentation with the presence of high noise and missing wedge effect in cryo-electron tomograms [90,91]. For example, Moebel et al. [91] develop the DeepFinder, which is a 3D CNN-based platform for multiclass macromolecule identification from cryo-electron tomograms. During the training process, the expert labeled 3D coordinates of the macromolecules are first converted to the voxel-wise annotation maps to avoid the time-consuming step of generating voxel-wise ground truth by manual segmentation. After the training, the DeepFinder can predict the macromolecule categories voxel wisely and then use them to determine the location of macromolecule particles. The DeepFinder outperforms the conventional template matching and other deep learning methods in

both synthetic and experimental datasets, which makes it a promising algorithm for the semiautomated analysis of the cryo-electron tomography data.

The ML approaches can also directly learn to give the reconstructions with the sinogram as inputs and volumetric images as output. But despite efforts into such domain transform learning in medical CT [92–94] and positron emission tomography (PET) [95] imaging systems, their applications in the electron microscopy tomography techniques are still underexplored. For example, Zhu et al. [96] develop automated transform by manifold approximation (AUTOMAP), a supervised learning framework that allows the mapping between the senor data and the image domain (Fig. 4D). In AUTOMAP, the input sensor data will be passed thought fully connected layers followed by convolutional layers to give the final reconstruction. Although demonstrated on the MRI, the authors also anticipate that this algorithm can be applied in arbitrary image reconstruction tasks such as low-dose X-ray CT and PET with improved artifact reduction and reconstruction accuracy. Such direct ML reconstructions are implemented in an end-to-end manner and does not involve the assistance from other reconstruction methods, but disadvantages can still exist in the needs of training data and heavy computational cost [87].

4. ML-assisted analysis of nonimage data

While the recently fast-growing CNN benefits or directly enables the information extraction from real-space microscopy image data, other ML algorithms can contribute to the postprocessing and analysis of the large amount of resulting extracted information. For example, in the electron microscopy characterization of nanoparticle sample, the successful detection of individual particle contours has to be followed by the high-throughput and automatic data analysis [97] to reveal their shape attributes such as the aspect ratio of nanorods or the edge length and truncation of nanocubes which play important roles in determining their catalytic, optical, and biomedical properties. Meanwhile, the heterogeneity of the sample which is caused by the impurity species and the particle shape distribution is an indicator of the synthesis quality and can change the particle's self-assembly or other behaviors. Benefited from the "big data" nature of microscopy images of nanomaterial systems, unsupervised ML techniques are capable of analyzing the datapoints distribution in the feature space and clustering the nanoparticle shapes into groups with high similarity. Wang et al. [7] develop AutoDetect-mNP algorithm to classify the metal nanoparticle contours extracted from TEM images in an unsupervised manner. The AutoDetect-mNP utilizes four shape descriptors (area, eccentricity, aspect ratio, circularity) computed from the nanoparticle contours detected from TEM images as inputs and employ the *k*-means algorithm for the unsupervised classification. Their results show AutoDetect-mNP can reliably distinguish Au nanorods and triangular prisms from impurities and further differentiate shape subgroups with different aspect ratios and corner truncation without any human intervention. Similarly, in Lee

et al.'s work [39], the morphologies of gold nanorods are grouped in a two-step manner: A center-to-perimeter distance is first used to distinguish the particles into obvious geometries such as square, sphere, and rod. Then an unsupervised GMM which takes the major and minor axis length and solidity as the inputs is responsible for clustering them into subclasses with more shape details. Slater et al. [38] select solidity, maximum pixel intensity, and the projection area as the descriptors of the particle morphology and explore their classifications with three unsupervised learning algorithms including *k*-means, DBSCAN, and ordering points to identify the clustering structure (OPTICS) (Fig. 5A). For all clustering algorithms tested, the resulting datapoints from each

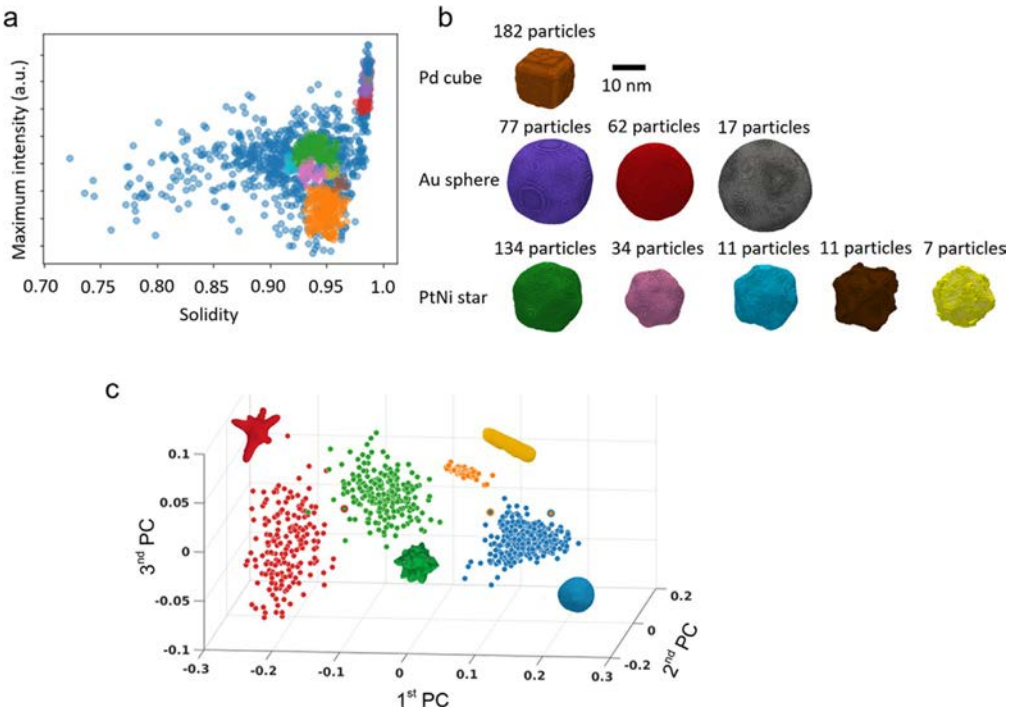


Fig. 5 ML works in nanoparticle morphology grouping. (A) Distribution of clusters of nanoparticles plotted as a function of particle maximum intensity and solidity. Particles are clustered by DBSCAN with area, maximum intensity, and solidity as input properties. (B) 3D reconstructions from the clusters shown in (A). Colors correspond to each cluster in (A). (C) Nanoparticle clustering using center of gravity method represented as scatter plots for the first three principal components (PCs). *Each dot* represents one individual nanoparticle. Resulting shape groups are colored in *blue*, *orange*, *green*, and *red*. *((A,B) Reproduced with permission from T.J.A. Slater, Y.-C. Wang, G.M. Leteba, J. Quiroz, P.H.C. Camargo, S.J. Haigh, et al., Automated single-particle reconstruction of heterogeneous inorganic nanoparticles, Microsc. Microanal. 26 (2020) 1168–75, https://doi.org/10.1017/S1431927620024642. (C) Adapted with permission from L. Boselli, H. Lopez, W. Zhang, Q. Cai, V.A. Giannone, J. Li, et al., Classification and biological identity of complex nano shapes, Commun. Mater. 1 (2020) 1–12, https://doi.org/10.1038/s43246-020-0033-2.)*

morphological group are passed into the single-particle reconstruction algorithm to reveal their averaged 3D geometries (Fig. 5B). The successful reconstruction not only confirms the high homogeneity within the groups found by the unsupervised classification, but also provides a new approach for the 3D reconstructions of heterogeneous inorganic nanoparticles. Comparing with their supervised or manual counterparts, in such unsupervised classifications of nanoparticle morphology, the advantage exists in the absence of training dataset and the reduced manual inputs. The absence of training dataset allows the algorithms to be robustly applied to nanoparticle systems with little previous knowledge. And the reduced manual inputs introduce less subjective bias from the human researchers, which truly reflects the natural distribution of the particle morphology.

It is noteworthy that in some unsupervised learning techniques such as *k*-means and GMM, the number of clusters still needs be user defined and can impact the clustering results significantly. To further decrease the level of such "supervision," metrics like maximum entropy [7] and Bayesian information criterion [39] can help the determination of optimal cluster numbers that give best datapoint separation. On the other hand, as one can tell, the selection of particle shape descriptors which serves as the input of the clustering algorithms can also be quite different from system to system. Examples include the solidity, convexity, area, eccentricity, aspect ratio, circularity, major/minor axis length, and even the maximum pixel intensity in the projection area. Obviously, different choices of the shape descriptors emphasize certain aspect of the particle geometry, leading to different classification results. Moreover, in most cases, especially for the particles with complicated convex contours, no descriptor can comprehensively keep all the information from the original particle contour. Considering this factor, the contour itself, if appropriately combined with dimension reduction techniques, can also be used as the unsupervised classification inputs (Fig. 5C), which are more generalizable to all different shapes regardless of the convexity [98]. Other than morphology classification, unsupervised learning algorithms such as PCA and *k*-means are also widely used in diffraction data classification which can automate and accelerate the crystal structure mapping in 4D STEM [10,12].

Another kind of information that can be extracted from electron microscopy data is the temporal information. For example, recent development of LP-TEM allows the real-time observation of nanoparticle diffusion, and the anomalous behavior in the resulting trajectories can lead to the studies in particle-substrate interaction, the heterogeneity of the liquid cell substrate, and even the liquid structure in the liquid cell environment. For such data with strong temporal and sequential correlation, certain types of neural network models like LSTM are designed to analyze and extract information from them, which has been already widely demonstrated in the trajectory analysis of colloidal particles [36,99–101] and biomedical molecules [102–106] imaged by optical microscopy. One example for such application is to overcome the long trajectory and large number of

particles needed in the particle diffusive dynamic characterization in LP-TEM. Jamali et al. [107] develop MoNet, which can classify as well as regress the power law dependence in the mean square displacement for short trajectories of individual nanoparticles, once trained on the simulated trajectory data. In MoNet, the input trajectory coordinates are first passed into six parallel 1D convolutional sublayers with different filter size and step size to capture local dynamics in different spatial resolutions. Then the outputs of the convolutional layers are concatenated and passed through several fully connected layers to give the final categorical or numerical regression results. Applying the trained MoNet model to experimental LP-TEM data further helps reveal the nanoparticle diffusion transition from fractional Brownian motion to continuous-time random walk (CTRW) when increasing the electron beam dose rate. This transition is caused by the electron beam radiolysis-induced passivation of the functional groups on the liquid cell membrane surface, which reduces the binding between nanoparticles and the liquid cell substrate and causes detrap and long-distance jumps as in the CTRW model.

5. Conclusion and outlook

Thanks to the pioneering work on model development in computer science and the successful application attempts in biomedical and optical microscopy image processing, ML, especially the CNNs, has now become a powerful tool in the processing and statistical analysis of electron microscopy data of nanomaterials. The direct information extraction from 2D electron microscopy images can be achieved by different designs of CNN architectures, covering tasks of increasing complexity from the single image classification and regression, image-to-image (pixelwise) classification and regression, object detection, to the instance segmentation. While more research in nanomaterial characterization is found to be focused on previous two tasks, the latter two are still under exploration probably due to the high computational cost and limited spatial accuracy, which should soon be overcome given the rapid-growing research efforts in model development. Similarly, beyond the instance segmentation, recent works in ML further show success in the object tracking, which means to link the trajectory of objects in a sequence of 2D image frames. Such trajectory tracking is helpful in the automation of nanoparticle diffusion study in real-time and real-space imaging such as LP-TEM and can be a potential future direction of research. In the 3D electron tomography with TEM and STEM-EDX, CNNs can help the reconstruction process by denoising, interpolating, and extrapolating the experimental sinogram, postfixing or segmenting the reconstruction given by simple algorithms, and implementing the actual sinogram-to-image domain transformation. Again, less work is found in the nanomaterial characterization field in the second and third scenarios, probably, due to the increasing model complexity and computational costs. It is also anticipated that the plenty of successful examples of the ML-based 3D reconstruction found in medical imaging field such as CT and PET can inspire more

work in the TEM and SETM-EDX tomography in the near future. Lastly, besides the CNNs, other supervised ML techniques such as fully connected neural networks, LSTM and unsupervised learning including the various dimension reduction techniques and clustering algorithms can contribute to the analysis of other type of nonimage information obtained from the microscopy characterization such as contour geometry, diffusion trajectory, and diffraction pattern. Implementing ML algorithms in nanomaterial study is a highly interdisciplinary research area, and one must be equipped with rich domain knowledge in material science as well as deep and wide understanding in statistics, mathematics, and ML models to sensor the needs and provide the solution. In this review, the authors attempt to have a glance on recent ML efforts in analyzing the nanomaterial microscopy data and, not intended to being comprehensive, to provide some useful insights on the ML model developing, selecting, and training in solving practical microscopy image analysis challenges.

Acknowledgment

We thank the support from National Science Foundation No. 1752517.

References

[1] T. Li, A.J. Senesi, B. Lee, Small angle X-ray scattering for nanoparticle research, Chem. Rev. 116 (2016) 11128–11180, https://doi.org/10.1021/acs.chemrev.5b00690.
[2] X. Ye, M.R. Jones, L.B. Frechette, Q. Chen, A.S. Powers, P. Ercius, et al., Single-particle mapping of nonequilibrium nanocrystal transformations, Science 354 (2016) 874–877, https://doi.org/10.1126/science.aah4434.
[3] L. Chen, A. Leonardi, J. Chen, M. Cao, N. Li, D. Su, et al., Imaging the kinetics of anisotropic dissolution of bimetallic core–shell nanocubes using graphene liquid cells, Nat. Commun. 11 (2020) 3041, https://doi.org/10.1038/s41467-020-16645-3.
[4] S.W. Chee, U. Anand, G. Bisht, S.F. Tan, U. Mirsaidov, Direct observations of the rotation and translation of anisotropic nanoparticles adsorbed at a liquid–solid interface, Nano Lett. 19 (2019) 2871–2878, https://doi.org/10.1021/acs.nanolett.8b04962.
[5] Z. Ou, Z. Wang, B. Luo, E. Luijten, Q. Chen, Kinetic pathways of crystallization at the nanoscale, Nat. Mater. 19 (2020) 450–455, https://doi.org/10.1038/s41563-019-0514-1.
[6] J.L. Vincent, R. Manzorro, S. Mohan, B. Tang, D.Y. Sheth, E.P. Simoncelli, et al., Developing and evaluating deep neural network-based denoising for nanoparticle TEM images with ultra-low signal-to-noise, Microsc. Microanal. (2021) 1–17, https://doi.org/10.1017/S1431927621012678.
[7] X. Wang, J. Li, H.D. Ha, J.C. Dahl, J.C. Ondry, I. Moreno-Hernandez, et al., AutoDetect-mNP: an unsupervised machine learning algorithm for automated analysis of transmission electron microscope images of metal nanoparticles, JACS Au 1 (2021) 316–327, https://doi.org/10.1021/jacsau.0c00030.
[8] C.S. Adorf, T.C. Moore, Y.J.U. Melle, S.C. Glotzer, Analysis of self-assembly pathways with unsupervised machine learning algorithms, J. Phys. Chem. B 124 (2020) 69–78, https://doi.org/10.1021/acs.jpcb.9b09621.
[9] B. Karlik, M.F. Yilmaz, M. Ozdemir, C.T. Yavuz, Y. Danisman, A hybrid machine learning model to study UV-vis spectra of gold nanospheres, Plasmonics 16 (2021) 147–155, https://doi.org/10.1007/s11468-020-01267-8.

[10] B.H. Martineau, D.N. Johnstone, A.T.J. van Helvoort, P.A. Midgley, A.S. Eggeman, Unsupervised machine learning applied to scanning precession electron diffraction data, Adv. Struct. Chem. Imaging 5 (2019) 3, https://doi.org/10.1186/s40679-019-0063-3.
[11] D. Chen, Y. Bai, S. Ament, W. Zhao, D. Guevarra, L. Zhou, et al., Automating crystal-structure phase mapping by combining deep learning with constraint reasoning, Nat. Mach. Intell. 3 (2021) 812–822, https://doi.org/10.1038/s42256-021-00384-1.
[12] F.I. Allen, T.C. Pekin, A. Persaud, S.J. Rozeveld, G.F. Meyers, J. Ciston, et al., Fast grain mapping with sub-nanometer resolution using 4D-STEM with grain classification by principal component analysis and non-negative matrix factorization, Microsc. Microanal. 27 (2021) 794–803, https://doi.org/10.1017/S1431927621011946.
[13] B. Kim, S. Lee, J. Kim, Inverse design of porous materials using artificial neural networks, Sci. Adv. 6 (2020), https://doi.org/10.1126/sciadv.aax9324, eaax9324.
[14] C. Yeung, R. Tsai, B. Pham, B. King, Y. Kawagoe, D. Ho, et al., Global inverse design across multiple photonic structure classes using generative deep learning, Adv. Opt. Mater. 9 (2021) 2100548, https://doi.org/10.1002/adom.202100548.
[15] M. Kilgour, N. Gastellu, D.Y.T. Hui, Y. Bengio, L. Simine, Generating multiscale amorphous molecular structures using deep learning: a study in 2D, J. Phys. Chem. Lett. 11 (2020) 8532–8537, https://doi.org/10.1021/acs.jpclett.0c02535.
[16] W. Ma, E.J. Kautz, A. Baskaran, A. Chowdhury, V. Joshi, B. Yener, et al., Image-driven discriminative and generative machine learning algorithms for establishing microstructure–processing relationships, J. Appl. Phys. 128 (2020), https://doi.org/10.1063/5.0013720, 134901.
[17] C. Wang, G. Ding, Y. Liu, H.L. Xin, 0.7 Å resolution electron tomography enabled by deep-learning-aided information recovery, Adv. Intell. Syst. 2 (2020) 2000152, https://doi.org/10.1002/aisy.202000152.
[18] Z. Zhou, X. Li, R.N. Zare, Optimizing chemical reactions with deep reinforcement learning, ACS Cent. Sci. 3 (2017) 1337–1344, https://doi.org/10.1021/acscentsci.7b00492.
[19] P. Leinen, M. Esders, K.T. Schütt, C. Wagner, K.-R. Müller, F.S. Tautz, Autonomous robotic nanofabrication with reinforcement learning, Sci. Adv. 6 (2020), https://doi.org/10.1126/sciadv.abb6987, eabb6987.
[20] J. Zhang, J. Yang, Y. Zhang, M.A. Bevan, Controlling colloidal crystals via morphing energy landscapes and reinforcement learning, Sci. Adv. 6 (2020), https://doi.org/10.1126/sciadv.abd6716, eabd6716.
[21] Y. Yang, M.A. Bevan, B. Li, Efficient navigation of colloidal robots in an unknown environment via deep reinforcement learning, Adv. Intell. Syst. 2 (2020) 1900106, https://doi.org/10.1002/aisy.201900106.
[22] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, et al., Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.
[23] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90, https://doi.org/10.1145/3065386.
[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv14091556 Cs (2015).
[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, ArXiv151203385 Cs (2015).
[26] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, ArXiv14114038 Cs (2015).
[27] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, ArXiv150504597 Cs (2015).
[28] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, ArXiv13112524 Cs (2014).
[29] R. Girshick, Fast R-CNN, ArXiv150408083 Cs (2015).
[30] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, ArXiv150601497 Cs (2016).

[31] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, ArXiv150602640 Cs (2016).
[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, ArXiv170802002 Cs (2018).
[33] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, ArXiv170306870 Cs, 2018.
[34] K. Hara, H. Kataoka, Y. Satoh, Learning Spatio-temporal features with 3D residual networks for action recognition, ArXiv170807632 Cs (2017).
[35] Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-net: learning dense volumetric segmentation from sparse annotation, ArXiv160606650 Cs (2016).
[36] A. Argun, G. Volpe, S. Bo, Classification, inference and segmentation of anomalous diffusion with recurrent neural networks, J. Phys. Math. Theor. 54 (2021), https://doi.org/10.1088/1751-8121/ac070a, 294003.
[37] Y. Lu, C. Lu, C.-K. Tang, Online video object detection using association LSTM, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2363–2371, https://doi.org/10.1109/ICCV.2017.257.
[38] T.J.A. Slater, Y.-C. Wang, G.M. Leteba, J. Quiroz, P.H.C. Camargo, S.J. Haigh, et al., Automated single-particle reconstruction of heterogeneous inorganic nanoparticles, Microsc. Microanal. 26 (2020) 1168–1175, https://doi.org/10.1017/S1431927620024642.
[39] B. Lee, S. Yoon, J.W. Lee, Y. Kim, J. Chang, J. Yun, et al., Statistical characterization of the morphologies of nanoparticles through machine learning based electron microscopy image analysis, ACS Nano 14 (2020) 17125–17133, https://doi.org/10.1021/acsnano.0c06809.
[40] M. Radovanović, A. Nanopoulos, M. Ivanović, Hubs in space: popular nearest neighbors in high-dimensional data, J. Mach. Learn. Res. 11 (2010) 2487–2531.
[41] V. Pestov, Is the k-NN classifier in high dimensions affected by the curse of dimensionality? Comput. Math. Appl. 65 (2013) 1427–1437, https://doi.org/10.1016/j.camwa.2012.09.011.
[42] K.L.I.I.I. Pearson, On lines and planes of closest fit to systems of points in space, Lond. Edinb. Dublin Philos. Mag. J. Sci. 2 (1901) 559–572, https://doi.org/10.1080/14786440109462720.
[43] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[44] L. McInnes, J. Healy, J. Melville, UMAP: uniform manifold approximation and projection for dimension reduction, ArXiv180203426 Cs Stat (2020).
[45] D.P. Kingma, M. Welling, Auto-encoding variational bayes, ArXiv13126114 Cs Stat (2014).
[46] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, ArXiv14062661 Cs Stat (2014).
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.
[48] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, et al., C.O.C.O. Microsoft, Common objects in context, ArXiv14050312 Cs (2015).
[49] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, et al., A benchmark for comparison of cell tracking algorithms, Bioinformatics 30 (2014) 1609–1617, https://doi.org/10.1093/bioinformatics/btu080.
[50] S. Akers, E. Kautz, A. Trevino-Gavito, M. Olszta, B.E. Matthews, L. Wang, et al., Rapid and flexible segmentation of electron microscopy data using few-shot machine learning, Npj Comput. Mater. 7 (2021) 1–9, https://doi.org/10.1038/s41524-021-00652-z.
[51] A. Maksov, O. Dyck, K. Wang, K. Xiao, D.B. Geohegan, B.G. Sumpter, et al., Deep learning analysis of defect and phase evolution during electron beam-induced transformations in WS2, Npj Comput. Mater. 5 (2019) 1–8, https://doi.org/10.1038/s41524-019-0152-9.
[52] J. Li, M. Telychko, J. Yin, Y. Zhu, G. Li, S. Song, et al., Machine vision automated chiral molecule detection and classification in molecular imaging, J. Am. Chem. Soc. 143 (2021) 10177–10188, https://doi.org/10.1021/jacs.1c03091.
[53] M. Ziatdinov, O. Dyck, A. Maksov, X. Li, X. Sang, K. Xiao, et al., Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations, ACS Nano 11 (2017) 12742–12752, https://doi.org/10.1021/acsnano.7b07504.

[54] M. Ziatdinov, O. Dyck, X. Li, B.G. Sumpter, S. Jesse, R.K. Vasudevan, et al., Building and exploring libraries of atomic defects in graphene: Scanning transmission electron and scanning tunneling microscopy study, Sci. Adv. 5 (2019), https://doi.org/10.1126/sciadv.aaw8989, eaaw8989.
[55] L. Yao, Z. Ou, B. Luo, C. Xu, Q. Chen, Machine learning to reveal nanoparticle dynamics from liquid-phase TEM videos, ACS Cent. Sci. 6 (2020) 1421–1430, https://doi.org/10.1021/acscentsci.0c00430.
[56] L. Mill, D. Wolff, N. Gerrits, P. Philipp, L. Kling, F. Vollnhals, et al., Synthetic image rendering solves annotation problem in deep learning nanoparticle segmentation, Small Methods 5 (2021) 2100223, https://doi.org/10.1002/smtd.202100223.
[57] Q. Luo, E.A. Holm, C. Wang, A transfer learning approach for improved classification of carbon nanomaterials from TEM images, Nanoscale Adv. 3 (2021) 206–213, https://doi.org/10.1039/D0NA00634C.
[58] M. Rashidi, R.A. Wolkow, Autonomous scanning probe microscopy in situ tip conditioning through machine learning, ACS Nano 12 (2018) 5185–5189, https://doi.org/10.1021/acsnano.8b02208.
[59] J. Hu, T. Liu, P. Choo, S. Wang, T. Reese, A.D. Sample, et al., Single-nanoparticle orientation sensing by deep learning, ACS Cent. Sci. 6 (2020) 2339–2346, https://doi.org/10.1021/acscentsci.0c01252.
[60] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, J. Liang, UNet++: a nested U-net architecture for medical image segmentation, ArXiv180710165 Cs Eess Stat (2018).
[61] M. Ziatdinov, S. Zhang, O. Dollar, J. Pfaendtner, C.J. Mundy, X. Li, et al., Quantifying the dynamics of protein self-organization using deep learning analysis of atomic force microscopy data, Nano Lett. 21 (2021) 158–165, https://doi.org/10.1021/acs.nanolett.0c03447.
[62] M.Y. Yaman, K.N. Guye, M. Ziatdinov, H. Shen, D. Baker, S.V. Kalinin, et al., Alignment of Au nanorods along de novo designed protein nanofibers studied with automated image analysis, Soft Matter 17 (2021) 6109–6115, https://doi.org/10.1039/D1SM00645B.
[63] Z. Ou, L. Yao, H. An, B. Shen, Q. Chen, Imaging how thermal capillary waves and anisotropic interfacial stiffness shape nanoparticle supracrystals, Nat. Commun. 11 (2020) 4555, https://doi.org/10.1038/s41467-020-18363-2.
[64] J. Roerdink, A. Meijster, The watershed transform: definitions, algorithms and parallelization strategies, Fund. Inform. (2000), https://doi.org/10.3233/FI-2000-411207.
[65] C. Park, J.Z. Huang, J.X. Ji, Y. Ding, Segmentation, inference and classification of partially overlapping nanoparticles, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1, https://doi.org/10.1109/TPAMI.2012.163.
[66] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, H. Haario, Segmentation of overlapping elliptical objects in silhouette images, IEEE Trans. Image Process. 24 (2015) 5942–5952, https://doi.org/10.1109/TIP.2015.2492828.
[67] M. Shen, G. Li, D. Wu, Y. Liu, J.R.C. Greaves, W. Hao, et al., Multi defect detection and analysis of electron microscopy images with deep learning, Comput. Mater. Sci. 199 (2021), https://doi.org/10.1016/j.commatsci.2021.110576, 110576.
[68] Y. Xiao, G. Yang, A fast method for particle picking in cryo-electron micrographs based on fast R-CNN, AIP Conf. Proc. 1836 (2017), https://doi.org/10.1063/1.4982020, 020080.
[69] A. Al-Azzawi, A. Ouadou, H. Max, Y. Duan, J.J. Tanner, J. Cheng, DeepCryoPicker: fully automated deep neural network for single protein particle picking in cryo-EM, BMC Bioinform. 21 (2020) 509, https://doi.org/10.1186/s12859-020-03809-7.
[70] F. Zhang, Q. Zhang, Z. Xiao, J. Wu, Y. Liu, Spherical nanoparticle parameter measurement method based on Mask R-CNN segmentation and edge fitting, in: Proc. 2019 8th Int. Conf. Comput. Pattern Recognit., New York, NY, USA: Association for Computing Machinery, 2019, pp. 205–212, https://doi.org/10.1145/3373509.3373590.
[71] F. Zhang, D. Zhao, Z. Xiao, J. Wu, L. Geng, W. Wang, et al., Rodlike nanoparticle parameter measurement method based on improved mask R-CNN segmentation, SIViP 15 (2021) 579–587, https://doi.org/10.1007/s11760-020-01779-0.
[72] C. Cabrera, D. Cervantes, F. Muñoz, G. Hirata, P. Juárez, D.-L. Flores, Mask R-CNN to classify chemical compounds in nanostructured materials, in: C.A. González Díaz, C. Chapa González, E.

Laciar Leber, H.A. Vélez, N.P. Puente, D.-L. Flores, et al., (Eds.), VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering, Springer International Publishing, Cham, 2020, p. 401, https://doi.org/10.1007/978-3-030-30648-9_52.
[73] P. Monchot, L. Coquelin, K. Guerroudj, N. Feltin, A. Delvallée, L. Crouzier, et al., Deep learning based instance segmentation of titanium dioxide particles in the form of agglomerates in scanning electron microscopy, Nano 11 (2021) 968, https://doi.org/10.3390/nano11040968.
[74] U. Schmidt, M. Weigert, C. Broaddus, G. Myers, Cell detection with star-convex polygons, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Med. Image Comput. Comput. Assist. Interv. – MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 265–273, https://doi.org/10.1007/978-3-030-00934-2_30.
[75] F.C. Walter, S. Damrich, F.A. Hamprecht, MultiStar: instance segmentation of overlapping objects with star-convex polygons, ArXiv201113228 Cs Eess (2021).
[76] D. Neven, B. De Brabandere, M. Proesmans, Van Gool L. instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth, ArXiv190611109 Cs (2019).
[77] B. Yildirim, J.M. Cole, Bayesian particle instance segmentation for electron microscopy image quantification, J. Chem. Inf. Model. 61 (2021) 1136–1149, https://doi.org/10.1021/acs.jcim.0c01455.
[78] A. Colliard-Granero, M. Batool, J. Jankovic, J.H. Jitsev, M. Eikerling, K. Malek, et al., Deep learning for the automation of particle analysis in catalyst layers for polymer electrolyte fuel cells, Nanoscale (2021), https://doi.org/10.1039/D1NR06435E.
[79] C.G. Khoury, T. Vo-Dinh, Gold nanostars for surface-enhanced raman scattering: synthesis, characterization and optimization, J. Phys. Chem. C 112 (2008) 18849–18859, https://doi.org/10.1021/jp8054747.
[80] Q. Cui, B. Xia, S. Mitzscherling, A. Masic, L. Li, M. Bargheer, et al., Preparation of gold nanostars and their study in selective catalytic reactions, Colloids Surf. Physicochem. Eng. Asp. 465 (2015) 20–25, https://doi.org/10.1016/j.colsurfa.2014.10.028.
[81] X. Song, J.W. Smith, J. Kim, N.J. Zaluzec, W. Chen, H. An, et al., Unraveling the morphology–function relationships of polyamide membranes using quantitative electron tomography, ACS Appl. Mater. Interfaces 11 (2019) 8517–8526, https://doi.org/10.1021/acsami.8b20826.
[82] P.A. Midgley, R.E. Dunin-Borkowski, Electron tomography and holography in materials science, Nat. Mater. 8 (2009) 271–280, https://doi.org/10.1038/nmat2406.
[83] Y. Han, J. Jang, E. Cha, J. Lee, H. Chung, M. Jeong, et al., Deep learning STEM-EDX tomography of nanocrystals, Nat. Mach. Intell. 3 (2021) 267–274, https://doi.org/10.1038/s42256-020-00289-5.
[84] A. Skorikov, W. Heyvaert, W. Albecht, D.M. Pelt, S. Bals, Deep learning-based denoising for improved dose efficiency in EDX tomography of nanoparticles, Nanoscale 13 (2021) 12242–12249, https://doi.org/10.1039/D1NR03232A.
[85] P. Gilbert, Iterative methods for the three-dimensional reconstruction of an object from projections, J. Theor. Biol. 36 (1972) 105–117, https://doi.org/10.1016/0022-5193(72)90180-4.
[86] K.J. Batenburg, J. Sijbers, DART: a practical reconstruction algorithm for discrete tomography, IEEE Trans. Image Process. 20 (2011) 2542–2553, https://doi.org/10.1109/TIP.2011.2131661.
[87] G. Wang, J.C. Ye, B. De Man, Deep learning for tomographic image reconstruction, Nat. Mach. Intell. 2 (2020) 737–748, https://doi.org/10.1038/s42256-020-00273-z.
[88] R. Aprilliyani, R.G. Prabowo, Basari, On the performance of SART and ART algorithms for microwave imaging, AIP Conf. Proc. 1933 (2018), https://doi.org/10.1063/1.5023980, 040010.
[89] X. Yang, V. De Andrade, W. Scullin, E.L. Dyer, N. Kasthuri, F. De Carlo, et al., Low-dose x-ray tomography through a deep convolutional neural network, Sci. Rep. 8 (2018) 2575, https://doi.org/10.1038/s41598-018-19426-7.
[90] C. Che, R. Lin, X. Zeng, K. Elmaaroufi, J. Galeotti, M. Xu, Improved deep learning based macromolecules structure classification from electron cryo tomograms, Mach. Vis. Appl. 29 (2018) 1227–1236, https://doi.org/10.1007/s00138-018-0949-4.
[91] E. Moebel, A. Martinez-Sanchez, L. Lamm, R.D. Righetto, W. Wietrzynski, S. Albert, et al., Deep learning improves macromolecule identification in 3D cellular cryo-electron tomograms, Nat. Methods 18 (2021) 1386–1394, https://doi.org/10.1038/s41592-021-01275-4.

[92] D.H. Ye, G.T. Buzzard, M. Ruby, C.A. Bouman, Deep back projection for sparse-view CT reconstruction, ArXiv180702370 Cs Eess (2018).
[93] Y. Li, K. Li, C. Zhang, J. Montoya, G.-H. Chen, Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions, IEEE Trans. Med. Imaging 38 (2019) 2469–2481, https://doi.org/10.1109/TMI.2019.2910760.
[94] L. Fu, B. De Man, A hierarchical approach to deep learning and its application to tomographic reconstruction, ArXiv191207743 Phys (2019).
[95] I. Häggström, C.R. Schmidtlein, G. Campanella, T.J. Fuchs, DeepPET: a deep encoder–decoder network for directly solving the PET image reconstruction inverse problem, Med. Image Anal. 54 (2019) 253–262, https://doi.org/10.1016/j.media.2019.03.013.
[96] B. Zhu, J.Z. Liu, S.F. Cauley, B.R. Rosen, M.S. Rosen, Image reconstruction by domain-transform manifold learning, Nature 555 (2018) 487–492, https://doi.org/10.1038/nature25988.
[97] C.R. Laramy, K.A. Brown, M.N. O'Brien, M. ChadA, High-throughput, algorithmic determination of nanoparticle structure from electron microscopy images, ACS Nano 9 (2015) 12488–12495, https://doi.org/10.1021/acsnano.5b05968.
[98] L. Boselli, H. Lopez, W. Zhang, Q. Cai, V.A. Giannone, J. Li, et al., Classification and biological identity of complex nano shapes, Commun. Mater. 1 (2020) 1–12, https://doi.org/10.1038/s43246-020-0033-2.
[99] T. Wagner, A. Kroll, C.R. Haramagatti, H.-G. Lipinski, M. Wiemann, Classification and segmentation of nanoparticle diffusion trajectories in cellular micro environments, PLoS One 12 (2017), https://doi.org/10.1371/journal.pone.0170165, e0170165.
[100] N. Granik, L.E. Weiss, E. Nehme, M. Levin, M. Chein, E. Perlson, et al., Single-particle diffusion characterization by deep learning, Biophys. J. 117 (2019) 185–192, https://doi.org/10.1016/j.bpj.2019.06.015.
[101] S. Bo, F. Schmidt, R. Eichhorn, G. Volpe, Measurement of anomalous diffusion using recurrent neural networks, Phys. Rev. E 100 (2019), https://doi.org/10.1103/PhysRevE.100.010102, 010102.
[102] G. Muñoz-Gil, M.A. Garcia-March, C. Manzo, J.D. Martín-Guerrero, M. Lewenstein, Single trajectory characterization via machine learning, New J. Phys. 22 (2020), https://doi.org/10.1088/1367-2630/ab6065, 013010.
[103] J. Janczura, P. Kowalek, H. Loch-Olszewska, J. Szwabiński, A. Weron, Classification of particle trajectories in living cells: machine learning versus statistical testing hypothesis for fractional anomalous diffusion, Phys. Rev. E 102 (2020), https://doi.org/10.1103/PhysRevE.102.032402, 032402.
[104] P. Kowalek, H. Loch-Olszewska, J. Szwabiński, Classification of diffusion modes in single-particle tracking data: feature-based versus deep-learning approach, Phys. Rev. E 100 (2019), https://doi.org/10.1103/PhysRevE.100.032410, 032410.
[105] D. Han, N. Korabel, R. Chen, M. Johnston, A. Gavrilova, V.J. Allan, et al., Deciphering anomalous heterogeneous intracellular transport with neural networks, elife 9 (2020), https://doi.org/10.7554/eLife.52224, e52224.
[106] H. Loch-Olszewska, J. Szwabiński, Impact of feature choice on machine learning classification of fractional anomalous diffusion, Entropy 22 (2020) 1436, https://doi.org/10.3390/e22121436.
[107] V. Jamali, C. Hargus, A. Ben-Moshe, A. Aghazadeh, H.D. Ha, K.K. Mandadapu, et al., Anomalous nanoparticle surface diffusion in LCTEM is revealed by deep learning-assisted analysis, Proc. Natl. Acad. Sci. 118 (2021), https://doi.org/10.1073/pnas.2017616118.

CHAPTER 11

Deep learning in biomedical informatics

Che-Lun Hung
Institute of Biomedical Informatics, National Yang Ming Chiao Tung University, Taipei, Taiwan

1. Introduction

With the massive influx of multimodal data in the last decade, the role of data analytics in health informatics has grown rapidly. Deep learning (DL) is defined as a technology based on artificial neural networks (ANNs), which have recently emerged as a powerful machine learning (ML) tool. The rapid increase in computing power, fast data storage, and parallelization, as well as predictive capabilities and the ability to generate automatically optimized advanced functions and semantic interpretation from input data, all contribute to the rapid adoption of this technology. This chapter introduces the latest developments in the use of deep learning in health informatics and makes an important analysis of the relative advantages, potential shortcomings, and future prospects of the technology. This chapter mainly focuses on the key applications of DL in the fields of computational biology, drug design, medical imaging, pervasive sensing, medical informatics, and public health. Artificial intelligence (AI) has become a trend in recent years, opening up an entirely new era of research in various fields. The demand for AI in healthcare has increased in both the academia and industry; therefore, the potential benefits of its applications have been proven. Previous studies have attempted to implement AI methods on medical images, electronic health records, molecular characteristics, and a variety of lifestyles. Researchers used data aggregated from multiple data sources to train models that mimic what clinicians do when they see patients and help in decision-making through results and interpretations. It included how to read clinical images, predict results, discover the connection between genotype and phenotype or phenotype and disease, analyze treatment responses, and track lesions or structural changes (e.g., hippocampal volume reduction). In addition, predictive research (e.g., disease or readmission predictions) and correlation and pattern recognition research have been extended to early warning systems with risk scores and overall pattern research and population care (such as predictive care for an entire population).

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00011-1
Copyright © 2023 Elsevier Inc.
All rights reserved.

2. Deep learning network

2.1 Convolutional neural networks

Convolutional neural networks (CNNs) have been widely used in image recognition tasks. There are several reasons why CNN models are gaining popularity. For instance, CNN models have a deeper architecture; thus, it can analyze more detailed features in images than traditional neural networks (NNs). The typical structure of CNNs has three layers, that is, the input layer, hidden layer, and output layer. Raw images are fed into the input layer and further transferred to the hidden layer to perform feature extraction tasks. The hidden layer is composed of convolution, pooling and fully connected layers. Feature extraction in the convolution and pooling layers is the main characteristic of a CNN model. The convolution layer can automatically extract features from raw images by convoluting the raw images with a sliding window of a fixed-size filter. The function of the pooling layer is to reduce the dimensionality of feature maps while retaining relevant features. Generally, the fully connected layer is the last layer of a CNN model, which then flattens feature maps to perform classification.

2.2 Recurrent neural network

A recurrent neural network (RNN) is a type of ANN model, which considers history information that is widely applied in the field of natural language and time sequence data. An RNN can continually update the hidden state in an RNN model. The operation of an RNN considers the current input and previous hidden state to constantly update the condition of the hidden state. Although RNN models have achieved high performance on temporal data, they exhibit the vanishing and exploding gradient problem for long-term sequence data. RNN models often cannot memorize long sequence data; therefore, they cannot improve performance when processing long time-series data.

2.3 Long short-term memory

Long short-term memory (LSTM) is a variation of an RNN model. An RNN can only memorize short-term information, but LSTM can handle long time-series data. Moreover, an RNN model has the vanishing gradient problem for the long sequence data; however, LSTM can prevent this problem during training. An LSTM model can recall previous long-term time-series data and has automatic control for retaining relevant features or discarding irrelevant features in the cell state. An LSTM model has three gates to control features, that is, the input gate, forget gate, and output gate. The input gate controls new information to flow into the cell state. The forget gate removes previous unimportant information from the cell state. The output gate regulates extracted information from the cell state and then decides the next hidden state. An LSTM model can automatically save or remove stored memory using these gates.

2.4 Autoencoder

An autoencoder (AE) is described as an unsupervised learning DL-based model that can automatically discover the potential rule of data. The architecture of an AE consists of three parts, that is, an encoder, code, and decoder. The main purpose of the encoder is to compress input information and then generate the code. The code of an AE represents the latent space, which is the summary of the input data after processing the compression operator. Furthermore, the decoder reconstructs the input data based on the code. The core of an AE model is the use of compression and decompression processes to make input and output values have the same meaning.

2.5 Applications

2.5.1 Transfer learning

Transfer learning is a machine learning method which is used to train a target model on a target dataset using learned features obtained from a pre-trained model on a base dataset [1–4]. It not only can improve the accuracy of training target model, but also is able to significantly improve overfitting issue caused by the small target dataset. Usually, the pre-trained network parameters are regarded as the initial network parameters for a target network. In other words, these weights can also be the initialized layer of the target network. Then, the target network is trained on the target dataset with pre-trained features [1].

Generative adversarial networks (GANs) employ two networks, generator and discriminator, to contest with each other to learn a representative distribution from the training data [5]. A generator network generates fake images from a random noise input image, while a discriminator network predicts whether the input images are real images or generated images. The parameters in both generator and discriminator are updated via backpropagation. For approaching optimization, the generator continues to generate fake images which are more and more similar to real images. Meanwhile, the discriminator has to distinguish real or generated fake images. Finally, the well-trained discriminator is able to use to perform classification tasks.

2.5.2 Computational biology

For the past 20 years, with the gathering of an outsized amount of biological and medical data, as well as the emergence of genome sequences, protein structures, and drug inhibitors, many biomedical problems have emerged [6–10]. Computational biology is an interdisciplinary field that involves the analysis and discovery of biological phenomena on large collections of biological data, such as genetic sequences, cell populations or protein samples, by using computational methods to make new predictions or discover new biology. In computational biology, the major biological data are based on the genome,

and its common applications include transcriptomics [9], splicing analysis [9,11], gene expression [11] and variants [7,12], and DNA methylation [7,8], among others [7,9,11].

2.5.3 Transcriptomics

For transcriptomics, DL has been applied successfully in transcriptomics data with high-dimensional matrixes. Ibrahim et al. [13] proposed a multi-level feature selection approach (MLFS) based on DBN with active learning to discriminate genes and miRNAs based on expression profiles that resulted in the classification of six cancers. In addition to DBN, Kelley et al. [14] developed an open-source framework, Basset, under CNN, to explore the effects of sequence variants on chromatin regulation and annotate the non-coding genome for DNase-seq data. For CNN, Jing et al. [15] and Washburn et al. [16] also reported transcriptomic applications. Jing et al. attempted to predict transcription factor-binding sites using a general CNN model; Washburn et al. used different gene families to constrain datasets to explain the evolution in the first approach, and then in the second approach, adopting evolutionary information of orthologous genes to control the evolutionary divergence in the training process. In addition, for time series-based DL methods, Quang and Xie [17] presented two approaches to predict noncoding function de novo from sequences alone; one is pure RNN, another is an integration framework with CNN and BLSTM (CNN + BLSTM).

2.5.4 Splicing analysis

For splicing analysis, Leung et al. [18] proposed a DNN model with dropout to predict alternative splicing (splicing patterns) from five tissue-specific RNA-seq datasets in mouse, whose method exhibits better performance than that of Bayesian network (BN). Liu et al. [19] also proposed another DNN model combined with HMM (DNN + HMM), PEDLA, for identifying the cis-regulatory regions and replication timing domains, respectively. For other DL methods, Lee and Yoon [20] developed a DBN-based method integrated with RBM's training process (for class imbalanced prediction) by boosting contrastive divergence with categorical gradients to predict splice junction at the DNA level. Yang et al. [21] presented an integration architecture BiRen, which is integrated CNN and BRNN (CNN + BRNN), to predict enhancer patterns from the DNA sequence alone. Using the integration strategy, Bretschneider et al. [22] proposed the hybrid method COSSMO, which has two models, one is LSTM, another integrates Resnet and LSTM (Resnet + LSTM), to explain the competitive effects and predict the percentage selected index distribution of splice sites.

2.5.5 Gene expression

For gene expression, Denas and Taylor [23] designed a CNN model as a preprocessing job to translate the ChIP-seq data into the gene's transcription factor activity profiles. After that, Zeng et al. [24] attempted to use the transfer-learning approach (TLA) with

CNN from natural images to biological images. The pre-trained CNN has been used to annotate gene expression pattern images automatically in the mouse brain at various structural levels. Singh et al. [25] also developed DeepChrome, a CNN framework, to predict gene expression through feature extraction from histone modification in REMC database. For other DL methods, Chen et al. [26] exploited a DNN-based method with dropout, D-GEX, for gene expression profiling on both microarray and RNA-seq data, whose results outperformed those by linear regression (LiR). To explain how dependencies among histone modifications control the differential patterns of gene regulation, Sekhon et al. [27] presented an attention-based architecture, DeepDiff, which utilizes multiple LSTM modules to predict cell type-specific gene expression.

For genetic variants and DNA methylation, Quang et al. [28] proposed a DNN model, DANN, which outperforms LiR and SVM to annotate the pathogenicity of genetic variants. Zhou and Troyanskaya [29] developed an approach DeepSEA, a CNN-based model, to predict chromatin effects of sequence alterations with single-nucleotide sensitivity and the evaluation of disease-associated sequence variants. Wang et al. [30] designed a stacked DnAE (SDnAE)-based software, DeepMethy, to predict the methylation state of DNA CpG dinucleotides according to genome topology and DNA sequence patterns. Zeng and Gifford [31] presented a sequence-based framework, CpGenie, to learn regulatory codes of DNA methylation using CNN first, and then predict the impact of sequence variation on proximal CpG-site DNA methylation. Some studies integrate multiple methods in genetic variants and DNA methylation. For example, Angermueller et al. [32] proposed an integrated architecture with CNN and RNN (CNN + RNN), DeepCpG, for predicting missing CpG status for genome-wide analysis of single-cell DNA methylation data (within five cell types).

2.5.6 Genomics

In addition to the aforementioned applications in computational biology, there are many more applications, and numerous DL methods have been proposed for addressing such problems. Using an individual DL method, Xu et al. [33] developed an RNN and particle swarm optimization approach to investigate genetic regulatory networks under time-series gene expression data, whereas Chuang et al. [34] adopted RL and binary particle swarm optimization for operon prediction in bacterial genomes. Both Ralha et al. [35] and Bocicor et al. [36] also used RL for prediction. Ralha et al. proposed a system, BioAgent, to improve the biological sequence annotation performance. Bocicor et al. designed an RL-based approach to solve the DNA fragment assembly problem. There are more applications that use CNN and DNN to solve the genomics problem. For the trans-eQTL prediction problem, Witteveen [37] presented MASSQTL, an approach with nine DNN models from various encoded biological features. Ma et al. [38] reported DCell, a visible NN (VNN)-based system, which simulates a basic eukaryotic cell. Gudenas and Wang [39] designed a DNN-based model, DeepLncRNA, to predict subcellular

localization based on lncRNA transcript sequences. Umarov and Solovyev [39] analyzed sequence characteristics of prokaryotic and eukaryotic promoters using the CNN model. Several studies have presented multiple approaches simultaneously to compare their performance. For instance, Ditzler et al. [40] implemented two approaches including DBN and RNN, for metagenomics, and then compared them with multilayer perceptron NN (MPNN) to explore their applicability. Lin and Wong [41] designed and implemented two models, CNN and deep feedforward NN (DFNN), to predict off-target mutations in CRISPR-Cas9 and GUIDE-seq data sets.

With regard to combinations of more than two DL, ML, and statistical techniques, Liu et al. [42] proposed a hybrid architecture consisting of a combination of DNN and HMM (DNN + HMM) to identify cis-regulatory regions and replication timing domains, respectively, whose results outperformed those of a Gaussian mixture model-HMM hybrid. Fan and Zhang [43] designed a predictor, lncRNA-MFDL, which is a combination of RBM and Deep Stacked Network (DSN) (RBM + DSN), for the identification of lncRNAs from Gencode and Refseq. To predict translation initiation sites from QTI-seq data, Zhang et al. [44] presented TITER, a framework developed by integrating CNN and LSTM (CNN + LSTM). In addition, Arango-Argoty et al. [45] developed deep antibiotic resistance gene (DeepARG) models, DeepARG-SS for short sequences and DeepARG-LS for long genome sequences, to construct a dissimilarity matrix from all known categories of antibiotic resistance genes (ARGs). DeepARG is implemented using the Python Lasagne (PL) module (https://github.com/Lasagne/Lasagne), which has many versions based on complex methods such as CNN + LSTM + FNN. Furthermore, Li et al. [46] proposed an approach, DECRES, based on complex methods with randomized deep feature selection (RDFS), random forest (RF), and FNN (RDFS + RF + FNN) to recognize the enhancer and promoter regions in the human genome. Armenteros et al. [47] designed a combination model with RNN, conditional random field classification (CRFC), and optimized transfer learning (OTL) (RNN + CRFC + OTL) to perform the signal peptide predictions among three types of prokaryotic signal peptides.

3. Drug discovery

3.1 Computer-aided drug design

In the past few decades, computer-aided drug design (CADD) methods have been used for the rapid discovery of chemical libraries to match predicted and actual drug activity and reduce the cost of developing new active compounds. CADD is employed to simulate the interactions between active compounds and their targets, including receptors, enzymes, and transporters [48]. The simulation results are used to design complementary structures for their goals; then, many chemical databases are screened to identify new

biologically active compounds. Generally, the step of CADD is to simulate the active compound, then screen the compound database, and optimize the lead compound. Optimized leads can speed up the drug discovery process, thereby minimizing resource requirements. CADD approaches include two aspects: structure-based drug design and ligand-based drug design. The structure-based design approaches include docking, de novo ligand design, fragment-based drug discovery, and structure-based pharmacophore modification. The ligand-based design approaches include two classes, quantitative structure–affinity relationships (QSARs) and pharmacophore-based design. ANNs have been successfully used in CADD, especially in QSARs [5,49]. ANN methods are used in QSARs to represent chemical structures, preprocess and select the relevant descriptors, and so forth. The most widely used ANNs for QSARs are backpropagation NNs (BPNN), which is a forward NN with multilayered perception [50]. A BPNN is a robust method with several abilities such as superior fault tolerance, parallel co-processing, self-organizing, and self-learning. It is capable of approximating any nonlinear dependence of output variables, such as properties and activities on the values of input variables such as descriptors. It is currently the era of big data. Data, such as chemical libraries and biological data, have been noted to increase dramatically. Therefore, to screen such a significant number of chemical libraries and biological data, new approaches and platforms are needed. Graphics processing units (GPU) are a massively parallel architecture consisting of thousands of CUDA cores designed for handling large datasets. In 2006, Hinton et al. [51] proposed a DBN that can construct a network with many hidden layers. Krizhevsky et al. [52] introduced the first fast GPU implementation of a CNN, AlexNet, which consists of eight layers, to improve classification accuracy and computational performance. Afterward, several new deep CNNs have been proposed, such as VGG, ResNet, and GoogLeNet (Inception), for the ImageNet competition. These significant successful results have attracted the attention of many researchers and leading pharmaceutical companies. Hence, more DL algorithms have been used in CADD. These algorithms are described below. Ref. [9] used a DNN in drug discovery for the following four tasks: (1) new drug molecule identification; (2) protein engineering; (3) gene expression data analysis; and (4) pharmacodynamics modeling. We then summarize relevant research results in the following subsections.

3.1.1 New drug molecule identification

Computational models (e.g., virtual screening), as an effective and feasible way, have been used to identify new drug candidates from large chemical libraries to facilitate drug discovery processes. Zhang et al. [53] used a DL NN to analyze and predict compound selectivity. Ma et al. [54] used DNN and GPU to perform predictions on a set of large diverse QSAR datasets collected from Merck's drug discovery. Pereira et al. proposed DeepVS [55], which uses a DL NN to improve the performance of docking-based virtual screening. Kadurin et al. [56] presented a generative adversarial AE (AAE) for generating

novel molecular fingerprints with a defined set of parameters. They used the AAE in screening 72 million compounds in PubChem and select candidate molecules with potential anti-cancer properties. Hughes et al. [57] developed a deep CNN and quantitatively summarized the knowledge from hundreds of epoxidation reactions using a database of 702 epoxidation reactions to build a model that accurately predicted sites of epoxidation for developing safer drugs. Wang et al. [58] proposed a new calculation method for predicting diffusion tensor imaging measures from drug molecular structure and protein–sequence using a stacking AE with DL. Shin et al. [59] used DNN to classify the human colorectal carcinoma cell line (Caco-2) for predicting cellular permeability. IBM Research AI [60] proposed an end-to-end NN model, which can predict drug–target interactions directly from low-level representations.

3.1.2 Protein engineering

Researchers also use DL methods to explore and discover protein structure and function. To identify the function of proteins, many efforts have been made to mimic the interaction between proteins and other biomolecules such as DNA. Hassanzadeh and Wang [61] proposed the DeeperBind, which uses a long- and short-term recursive CNNs for predicting protein-binding specificity associated with DNA investigation. DL methods can also be used for predicting the biological function of proteins directly from their original three-dimensional (3D) electron density and electrostatic potential fields [62]. Schwartz et al. [63] have proposed a multi-task, multi-label DNN for protein annotation and discovery. They collected more than 90 million records, 80% of which were used as training sets, 10% to validate model training, and 10% used as a test set for the final evaluation of model performance. According to their experiment results, using a DNN can significantly reduce the time required for analysis and the computational cost compared to traditional methodologies. In their research, they also pointed out that through some creative DNN architecture, almost any imaginable experiment can contribute to the entire, including interactive networks, gene expression, epigenetics, phenotypic effects, drug binding, and clinical outcomes.

3.1.3 Gene expression data analysis

With the advent of next-generation sequencing technology, a large amount of heterogeneous genomics data can now meet the requirements of DL methods. Liang et al. [64] proposed an ML model to cluster cancer patients from multi-platform observation data. The model can identify both intra- and cross-modality correlations and meaningful disease subtypes from multi-platform cancer data. Alipanahi et al. [65] proposed a DL technology called DeepBind to identify causal disease variants and indicate how variations affect DNA- and RNA-binding within a specific sequence. Aliper et al. [66] demonstrated how to use DNNs to train on large transcriptional response datasets to classify drugs into treatment categories based on their transcriptional profiles. Chaudhary

et al. [67] have also used a DL-based model in hepatocellular carcinoma (HCC) classification, which could effectively distinguish the survival subgroups of patients in six cohorts and achieve HCC prognosis predictions. Xiao et al. [68] developed a DL-based ensemble model for cancer prediction. The model was evaluated on three public RNA-seq datasets for three cancers, that is, lung adenocarcinoma, gastric adenocarcinoma, and breast invasive cancer.

3.1.4 Pharmacodynamics modeling

Pharmacodynamics modeling has been identified to be critical for determining the interaction between a drug and related targets. Given the diversity of drug molecules and their targets, potential drug–protein interactions are complex and have many potential conformations. For example, Wen [69] et al. used a DNN for drug–target prediction. Kwon and Yoon [70] used CNNs for chemical–chemical interaction prediction. Lee et al. [71] used LSTM and feedforward NNs to predict the intraoperative bispectral index during the combined use of propofol and remifentanil. The DL hyperparameters of their model include eight memory cells in the long and short-term storage layers and 16 nodes in the hidden layer of the feedforward NNs. Model training and testing were then performed using separate datasets for 131 and 100 cases. The average consistency correlation coefficient was 0.561. Zhao et al. [72] developed a model based on ML and DL in symptom analysis, disease prediction, and medicine law of Traditional Chinese Medicine (TCM). They conducted experiments on relevant outpatient medical records and confirmed that the model can reflect the basic principles of clinical diagnosis of TCM combinations. The prediction model has good performance in evaluating indicators.

4. Medical images

Image detection and recognition focus on detecting a particular element in a medical image. In most cases, images are volumetric. Therefore, efficient parsing may be required. In this regard, a popular strategy to attempt is marginal space learning [73] because it is efficient and enables the robust detection of organs. Its DL counterpart [74] is even more efficient, as its probabilistic boosting trees are replaced with an NN-based boosting cascade. However, the entire volume of an image should still be processed to reliably detect anatomical structures. Using Ref. [74] improves efficiency even further by replacing the search process with a man-made agent that follows anatomy to detect anatomical landmarks using deep RL. The strategy can detect several landmarks in a complete CT volume in a few seconds. Bier et al. proposed a stimulating method by which anatomical landmarks are detected in two-dimensional (2D) X-ray projection images [75]. In their method, they trained projection-invariant feature descriptors from 3D annotated landmarks employing a DNN. Other popular detection methods are the so-called region proposal CNNs. In [76], they are used to robustly detect tumors in

mammographic images. Detection and recognition can also be used in many other ways, and an excellent body of literature exists. Here we only report on two other applications. In histology, cell detection and classification are a crucial task, which is addressed by Aubreville et al. using guided spatial transformer networks [77] that allow detection refinement before a particular classification is completed. The task of mitosis classification benefits from this procedure. CNNs are also highly effective for other image classification tasks. In [78], they are used to automatically detect images containing motion artifacts in confocal laser-endoscopy images.

4.1 Image segmentation

In addition, image segmentation has greatly benefited from recent developments in DL. In image segmentation, we aim to figure out the outline or structure of an organ as accurately as possible. Again, approaches based on CNNs seem to dominate. Here, we report only Holger Roth's DeepOrgan [79], which is the brain magnetic resonance (MR) segmentation performed by Moeskops et al. using CNNs [80], a completely convolutional multi-energy 3D U-net presented by Chen et al. [81], and a U-net-based stent segmentation in X-ray projection domain by Breininger et al. [82] as representative examples. As described by Nirschl et al., segmentation using deep CNNs can also be performed in two dimensions. Regarding histopathological images [83], Middleton et al. experimented with the fusion of NNs and active contour models in 2004 before the emergence of DL [84]. However, their approach neither used DNNs nor end-to-end training, which may be desirable for a state-of-the-art method. Hence, revisiting traditional segmentation approaches and integrating them with DL in an end-to-end manner seem to be a promising scope of research. Fu et al. follow a similar idea by mapping Frangi's vesselness into an NN [85]. They demonstrated that they can adjust convolution kernels within the initiative of the algorithm toward the precise task of vessel segmentation in ophthalmic fundus imaging. Another interesting class of segmentation algorithms is the use of recurrent networks for medical image segmentation. Poudel et al. demonstrated this for a fully connected convolutional RNN on multi-slice magnetic resonance imaging cardiac data [86], whereas Andermatt et al. showed the effectiveness of gated recurrent units for brain segmentation [87].

4.2 Image registration

While the perceptual tasks of image detection and classification receive a lot of attention with respect to applications of DL, image registration is yet to attract such attention. Nevertheless, there are several promising works on image registration in the literature, which indicate that there are numerous opportunities. One typical problem in point-based registration is identifying good feature descriptors that allow correct identification of the corresponding points. Wu et al. proposed using AEs to mine relevant features in an

unsupervised way [88]. Schaffert et al. promoted this even further and used the registration metric as a loss function for learning good feature representations [89]. An alternative way to unravel 2D/3D registration problems is to estimate the 3D pose directly from 2D point features [90]. For full volumetric registration, samples of DL-based approaches are also determined. The quicksilver algorithm can model a deformable registration and use a patch-wise prediction directly from an image appearance [1]. Another approach is to model a registration problem as an impact problem, which is addressed using an agent and RL. Liao et al. proposed predicting rigid registration to predict the best subsequent motion so that two volumes are aligned [2]. This approach can also be applied to non-rigid registration employing a statistical deformation model [3]. In this case, actions are movements within the vector space of the deformation model. Agent-based approaches also apply to point-based registration problems. Zhong et al. demonstrated this for intraoperative brain shift using imitation learning [4].

4.3 Computer-aided diagnosis

Computer-aided diagnosis is considered as one of the most challenging problems within the field of medical image processing. Here, we are not only playing a supporting role but also quantifying the evidence for diagnosis. On the contrary, the diagnosis is predictable. Therefore, decision-making should be thorough and must be reliable. The analysis of chest radiographs is a significant amount of work for radiologists and is performed routinely. Hence, reliable support to stop human error is extremely desirable. For example, Diamant et al. [91] used transfer-learning techniques in this regard. A similar workload is imposed on ophthalmologists when reading volumetric optical coherence tomography data. Google's DeepMind has only recently proposed to support this process in terms of referral decision support [92]. Many other studies have been conducted in this context, for instance, automatic cancer assessment in confocal laser-endoscopy in several tissues of the top and neck [1–4,76–93], DL for mammogram analysis [94], and classification of skin cancer [95].

4.4 Physical simulation

Support for physical modeling is a new area of DL. So far, this has been used in the gaming industry to calculate real-world physics engines [96] or to calculate smoke simulations in real time [97]. Meister et al. pioneered the introduction of DL into biomedical modeling. [98]. Based on this observation, researchers began to introduce this method into the field of medical imaging, for example, the depth scattering estimation of Maier et al. [99]. Unberath et al. went further to emulate an entire X-ray formation process in their DeepDRR [100]. Horger et al. [101] demonstrated that even noise of unknown distributions is often learned, leading to an efficient generative noise model for realistic physical simulations. In addition, other physical processes have been examined using DL. In

[102], cloth decomposition using DL embedding prior physical operators with precision learning is proposed. Furthermore, the identification of physically less plausible interrelations is attempted. Han et al. [103] then planned to convert MR volumes to computed tomography volumes. Stimpel et al. went further by predicting X-ray projections from MR projection images [104]. Although these observations seem promising, such endeavors should be carefully followed. Schiffers et al. demonstrated that cycle GANs may produce correctly appearing fluorescence images from fundus images in ophthalmology [105]. However, undesired effects appear, as occasionally drusen are mapped onto microaneurysms during this process. Cohen et al. showed even worse effects [106]. In their study, cancers disappeared or were created during the modality-to-modality mapping. Hence, such approaches need to be handled with care.

5. Electronic health records

5.1 Medical informatics

Medical informatics focuses on the analysis of large-scale aggregate data in the medical environment to enhance and develop clinical decision support systems, or it focuses on evaluating medical data to ensure quality assurance and accessibility of medical services. Electronic Health Record (EHR) is an extremely rich source of patient information, including detailed medical history information, such as diagnosis, diagnostic examinations, medications, treatment plans, immune records, allergies, radiological images, and sensor multiple time-series (such as EEG nursing unit), and laboratory and test results. Effectively mining these big data will provide valuable insights for disease management [107,108]. However, this is not easy for the following reasons:

(1) Due to different lengths, irregular sampling, lack of structured reports, and data loss, data are often complicated. The quality of reports varies greatly between organizations and individuals.

(2) Several petabytes of multimodal datasets, including medical images, sensor data, laboratory results, and unstructured text reports.

(3) The long-term time dependence between clinical events and disease diagnosis and treatment complicates learning. For example, there are often long and varying delays in distinguishing the onset of a disease from the onset of symptoms.

(4) Traditional ML methods cannot be extended to large and unstructured datasets.

(5) Lack of interpretability of results hinders the adaptability of the methods in the clinical environment.

DL methods have been carefully designed to scale large distributed datasets well. The success of DNNs is largely due to their ability to learn novel features/patterns and understand supervised and unsupervised data representations. A hierarchical system of supervision. DNNs can combine multiple DNN architecture components to effectly process multimodal information. Therefore, it is not surprising that DL is rapidly adopted in

medical informatics research. For example, Shin et al. [109] proposed a combined text image CNN to identify semantic information that links radiographic images and reports from a typical picture archive and a hospital's communication system. Liang et al. [110] have used a modified version of a CDBN as an effective training method for large-scale hypertension datasets and performed TCM diagnosis through a manually converted EHR database. Putin et al. [111] used a DNN to identify markers that predict the age of humans based on simple blood tests. Nie et al. [112] proposed a DL network for automatic disease inference, which requires manual collection of key symptoms or problems related to a disease.

In another study, Mioto et al. [113] showed that a set of denoising AEs can be used to automatically infer features from a large EHR database and represent patients without additional effort. These conventional functions can be used in several situations. The authors demonstrated that their system can predict the likelihood that a patient will develop a specific disease (such as diabetes, schizophrenia, and cancer). In addition, Futoma et al. [114] compared the ability of different models based on large EHR databases to predict hospitalization rates. Although training DNN models is not simple, the prediction accuracy of DNNs is much higher than traditional methods (such as penalty logistic regression).

Lipton et al. [115] used LSTM RNNs to solve the time dependence in EHRs using multivariate time-series data in an intensive care monitoring system. RNNs were used because they can remember sequential events, improving the modeling of different time delays between the onset of emergency clinical events (such as respiratory distress, asthma attacks, and symptoms). In related research, Mehrabi et al. [116] proposed the use of a DBN to discover common time patterns and characterize disease progression. The author also emphasizes that the identification and interpretation of newly discovered patterns require further research.

The motivation behind these studies is to develop a universal system to accurately predict the length of stay, future diseases, readmission rates, and mortality, to improve clinical decision-making, and optimize clinical pathways. Early prediction of healthcare is directly related to saving patients' lives. In addition, the discovery of novel models may lead to new hypotheses and research questions. In computational phenotyping research, the goal is to discover meaningful data-driven and disease characteristics.

For example, Che et al. [117] emphasized that although DNNs are superior to traditional ML methods in their ability to predict and classify clinical events, they suffer from the problem of model interpretability, which is important for clinical adaptability. They pointed out that the interpretation of individual units of DNNs may be misleading, and the behavior of DNNs is more complicated than originally thought. They suggested that once a DNN is trained using big data, a simpler model can be used to distribute knowledge and imitate the predictive performance of the DNN. To explain the functions of DL models, such as stacked denoizing AEs and LSTM-based RNNs, they use

gradient-enhanced decision trees (GBDTs). A GBDT is a collection of weak predictive models. In this work, they represented linear combinations of functions.

DL provides unprecedented powerful functions and efficiency by mining large-scale multimodal unstructured information stored in hospitals, cloud providers, and research organizations, paving the way for personalized healthcare. Although it has the potential to surpass traditional ML methods, proper initialization and adjustment are important to avoid overfitting. Noisy and sparse datasets can cause significant performance degradation, which indicates that some challenges need to be addressed. In addition, using DL systems in clinical practice requires the ability to track and interpret extracted features and patterns.

5.2 Public health

Public health aims to primarily prevent diseases, extend lifespans, and promote medical care by analyzing the spread of diseases and social behaviors related to environmental factors. Public health research transforms small-scale local populations into large-scale populations covering several continents, such as epidemics and pandemics. Its applications include epidemic monitoring, modeling of lifestyle diseases such as obesity related to geographic areas, monitoring and prediction of air quality, and drug safety monitoring. Traditional predictive models often grow exponentially with the size of the data and use complex models derived from physics, chemistry, and biology. Therefore, adjusting these systems can vary, depending on the parameter settings and temporary characteristics that only experts can provide. However, existing computational methods are also able to accurately model several phenomena, including the progression of diseases or the spread of air pollution. However, their ability to integrate real-time information is limited, which has been deemed essential for controlling epidemics or the adverse effects of newly approved drugs. In contrast, DL methods have a strong generalization ability. They are identified as data-driven methods that automatically build hierarchical models and encode information within their structure. Most DL algorithm designs are based on online ML. Therefore, as new training datasets appear, the optimization of the cost function will proceed in sequence. One of the simplest online optimization algorithms used in DNNs is stochastic gradient descent. For these reasons, DL, recommendation systems, and network analysis are recommended as the key analysis methods for public health research [118].

For example, monitoring and predicting the concentration of air pollutants is a representative area, where DL has been successful. Ong et al. [119] pointed out that poor air quality is responsible for approximately 60,000 deaths each year and the main cause of many chronic obstructive pulmonary diseases. They described a system that predicts the concentration of major air pollutants in Japan based on sensor data obtained from 52 cities. Their proposed DNN consists of stacked AEs and is trained online. The

difference between this deep architecture and the standard deep AE is that output components are gradually added during the training process. To track a large number of sensors and interpret results, they took advantage of the sparsity in their data and fine-tuned the DNN on the basis of a regularization method. However, they pointed out that DL methods as data-driven methods are affected by the inaccuracy and incompleteness of real-world data.

Another interesting application is tracking epidemiological and lifestyle disease outbreaks via social media. Social media can provide a wealth of information as regards disease progression in real time, such as influenza and Ebola. Zhao et al. [120] used the social media service Twitter to constantly track the health of the public. A DNN was used to mine popular features, and then the features are combined in a simulated environment to model the progression of a disease. The text in Twitter messages can also be used to gain insights into antibiotics and infectious intestinal diseases. Authors in [121] used a DBN to classify antibiotics-related Twitter posts into nine categories (side effects, needs/needs, advertisements, advice/information, animals, general use, drug resistance, abuse, and others). They then randomly selected Twitter messages for manual tagging and classification to obtain a classifier. They used a training set of 412 manually labeled examples and 150,000 unlabeled examples. Their RBM-based DL method was pre-trained in a layer-by-layer process. Fine-tuning was based on standard backpropagation and labeled data. In [113], DL was used to create keywords related to three types of infectious intestinal diseases (*Campylobacter*, *Norovirus*, and food poisoning). As compared to the officially recorded cases, their results show that social media can predict bowel diseases well.

Social media can also provide information that is not normally recorded to track certain stigmatizing behaviors; for example, Garimella et al. [122] used geotagged images from Instagram to track lifestyle diseases, such as obesity, drinking, and smoking, and compared self-categorization of images from users with annotations obtained using DL algorithms. Research has found that even though self-annotation can often provide useful demographic information, machine-generated annotations are more useful for behaviors such as excessive drinking and drug abuse. In [123], a DL method based on RBMs was developed to model and predict activity levels and prevent obesity by considering self-motivation, social influence, and environmental events.

People are becoming more interested in how to characterize and track human behavior using cell phone metadata. Metadata usually includes the duration and location of calls or text messages and can provide valuable demographic information. A CNN has been used to predict demographic information from cell phone metadata, which is represented as a 2D matrix of time. The CNN consists of a series of five horizontal convolutional layers, a vertical convolution filter, and two dense layers. This method achieves high accuracy for age and gender prediction while eliminating the need for manual features [109].

Using electronic medical records, mobile networks, and social media to mine online data and metadata about individuals and large-scale populations is a way to inform public health policies. In addition, mining food and drug records to identify adverse events can provide an important large-scale alert mechanism. We provide some examples that use DL to early identify and model the spread of epidemics and public health risks. However, strict regulations to protect data privacy restrict the access and aggregation of relevant information. For example, you can use Twitter messages or Facebook posts to identify new mothers who are at risk of postpartum depression. Although this is affirmative since this information will slander specific individuals, it is but controversial whether this information should be obtained. Therefore, we need to strike a balance between ensuring that individuals can control access to their private medical information and providing ways to make the information available for public health research [124]. The complexity and limited interpretability of DL models constitute an obstacle to making informed decisions about the precise operation of DNNs, which, in turn, may limit the application of DNNs to sensitive data.

5.3 Deep learning in healthcare: Limitations and challenges

Although for different AI tasks, DL technology can bring substantial improvements compared with traditional ML methods, many researchers and scientists are still skeptical about its use in medical applications. Because DL theory has not yet provided a complete solution, many questions remain unanswered; thus, these doubts have arisen. The following four aspects summarize some potential problems related to DL:

(1) Although some recent works have been conducted to visualize advanced features using weight filters in CNNs [125,126], the entire DL model is usually unexplainable. Therefore, most researchers use DL methods as black boxes and cannot explain why it provides good results or cannot apply modifications in a case of incorrect classification.

(2) As we have emphasized in the previous section, to train a reliable and effective model, expressing new concepts requires a lot of training data. Although we have recently witnessed a surge in available healthcare data, and many organizations have begun to effectively convert medical records from paper records to electronic records, disease-specific data remain scarce. Therefore, not all applications (especially rare diseases or events) are well suited for DL. A common problem may occur during DNN training (especially in the case of small datasets), namely, overfitting, which may occur when the number of parameters in the network is proportional to the total number of samples in the training set. In this case, the network can remember training examples but cannot generalize them to new examples that have not been observed. Therefore, although the error on a training set is driven to a small value, the error on a new dataset will be extremely high. To avoid the overfitting problem

and improve generalization ability, regularization methods such as dropout [127] are usually used in the training process.

(3) Another important aspect to require under consideration when deep learning tools are employed is that for several applications the data cannot be directly used as input for the DNN. Therefore, preprocessing, normalization or change of input domain is usually required before the training. Moreover, the setup of the many hyperparameters that control the architecture of a DNN, like the dimensions and therefore the number of filters during a CNN, or its depth, remains a blind exploration process that sometimes requires accurate validation. Finding the right preprocessing of the info and therefore the optimal set of hyperparameters are often challenging, since it makes the training process even longer, requiring significant training resources and human expertise, without which is not possible to get an efficient classification model.

(4) The last aspect we want to emphasize is that many DNNs can be easily fooled. For instance, [128] showed that some small changes (such as imperceptible noise in the image) can be made to the input samples, resulting in sample classification errors. Nevertheless, it must be pointed out that almost all machine learning algorithms are susceptible to such problems. The value of a particular feature can be deliberately set high or low to cause misclassification in logistic regression. Similarly, for decision trees, the samples can be guided along the wrong partition by simply switching a single binary feature on the final layer. Therefore, in general, any machine learning model is susceptible to such manipulations. On the other hand, the work in Ref. [129] discusses the opposite problem. The author shows that even without samples, it is possible to obtain meaningless synthetic samples, which will be strictly classified. This is also the true limitation of the deep learning paradigm, but it is a detrimental to other machine learning algorithms as well.

In summary, we believe that healthcare informatics today may be a human–machine collaboration which will ultimately become a symbiosis within the future. As more data becomes available, DL systems can elaborate and deliver where human interpretation is difficult. This will make diagnoses of diseases faster and smarter and reduce uncertainty of the deciding process. Finally, the last boundary of DL might be the feasibility of integrating data across disciplines of health informatics to support the longer term of precision medicine.

References

[1] X. Yang, R. Kwitt, M. Styner, M. Niethammer, Quicksilver: fast predictive image registration—a deep learning approach, Neuroimage 158 (2017) 378–396.

[2] R. Liao, S. Miao, P. de Tournemire, S. Grbic, A. Kamen, T. Mansi, et al., An Artificial Agent for Robust Image Registration, AAAI, 2017, pp. 4168–4175.

[3] J. Krebs, T. Mansi, H. Delingette, L. Zhang, F.C. Ghesu, S. Miao, et al., Robust non-rigid registration through agent-based action learning, in: Medical Image Computing and Computer-Assisted Intervention—MICCAI, Springer, 2017, pp. 344–352.

[4] X. Zhong, S. Bayer, N. Ravikumar, N. Strobel, A. Birkhold, M. Kowarschik, et al., Resolve intraoperative brain shift as imitation game, in: MICCAI Challenge 2018 for Correction of Brainshift With Intra-Operative Ultrasound (CuRIOUS 2018). CuRIOUS, Lecture Notes in Computer Science, Springer, 2018, pp. 129–137.
[5] T. Aoyama, Y. Suzuki, H. Ichikawa, Neural networks applied to structure-activity relationships, J. Med. Chem. 33 (1990) 905–908.
[6] B. Tang, Z. Pan, K. Yin, et al., Recent advances of deep learning in bioinformatics and computational biology, Front. Genet. 10 (2019) 214.
[7] D. Ravi, C. Wong, F. Deligianni, et al., Deep learning for health informatics, IEEE J. Biomed. Health Inform. 21 (2017) 4–21.
[8] M. Mahmud, M.S. Kaiser, A. Hussain, et al., Applications of deep learning and reinforcement learning to biological data, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 2063–2079.
[9] P. Mamoshina, A. Vieira, E. Putin, et al., Applications of deep learning in biomedicine, Mol. Pharm. 13 (2016) 1445–1454.
[10] M. Wainberg, D. Merico, A. Delong, et al., Deep learning in biomedicine, Nat. Biotechnol. 36 (2018) 829–838.
[11] M.K.K. Leung, A. Delong, B. Alipanahi, et al., Machine learning in genomic medicine: a review of computational problems and data sets, Proc. IEEE 104 (2016) 176–197.
[12] C. Cao, F. Liu, H. Tan, et al., Deep learning and its applications in biomedicine, Genomics Proteomics Bioinformatics 16 (2018) 17–32.
[13] R. Ibrahim, N.A. Yousri, M.A. Ismail, et al., Multi-level gene/miRNA feature selection using deep belief nets and active learning, Proc. Eng. Med. Biol. Soc. 2014 (2014) 3957–3960.
[14] D.R. Kelley, J. Snoek, J.L. Rinn, Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks, Genome Res. 26 (2016) 990–999.
[15] F. Jing, S.W. Zhang, Z. Cao, et al., An integrative framework for combining sequence and epigenomic data to predict transcription factor binding sites using deep learning, IEEE/ACM Trans. Comput. Biol. Bioinform. (2019), https://doi.org/10.1109/TCBB.2019.2901789.
[16] J.D. Washburn, M.K. Mejia-Guerra, G. Ramstein, et al., Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence, PNAS 116 (2019) 5542–5549.
[17] D. Quang, X. Xie, DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences, Nucleic Acids Res. 44 (2016), e107.
[18] M.K.K. Leung, H.Y. Xiong, L.J. Lee, et al., Deep learning of the tissue-regulated splicing code, Bioinformatics 30 (2014) i121–i129.
[19] F. Liu, H. Li, C. Ren, e al., PEDLA: predicting enhancers with a deep learning-based algorithmic framework, Sci. Rep. 6 (2016) 28517.
[20] T. Lee, S. Yoon, Boosted categorical restricted Boltzmann machine for computational prediction of splice junctions, Proc. Int. Mach. Learn. 37 (2015) 2483–2492.
[21] B. Yang, F. Liu, C. Ren, et al., BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone, Bioinformatics 33 (2017) 1930–1936.
[22] H. Bretschneider, S. Gandhi, A.G. Deshwar, et al., COSSMO: predicting competitive alternative splice site selection using deep learning, Bioinformatics 34 (2018) i429–i437.
[23] O. Denas, J. Taylor, Deep modeling of gene expression regulation in erythropoiesis model, in: Proceedings of the ICMLRL, Atlanta, USA, 2013, pp. 1–5.
[24] T. Zeng, R. Li, R. Mukkamala, et al., Deep convolutional neural networks for annotating gene expression patterns in the mouse brain, BMC Bioinf. 16 (2015) 147.
[25] R. Singh, J. Lanchantin, G. Robins, et al., DeepChrome: deep-learning for predicting gene expression from histone modifications, Bioinformatics 32 (2016) i639–i648.
[26] Y. Chen, Y. Li, R. Narayan, et al., Gene expression inference with deep learning, Bioinformatics 32 (2016) 1832–1839.
[27] A. Sekhon, R. Singh, Y. Qi, DeepDiff: DEEP-learning for predicting DIFFerential gene expression from histone modifications, Bioinformatics 34 (2018) i891–i900.
[28] D. Quang, Y. Chen, X. Xie, DANN: a deep learning approach for annotating the pathogenicity of genetic variants, Bioinformatics 31 (2015) 761–763.

[29] J. Zhou, O.G. Troyanskaya, Predicting effects of noncoding variants with deep learning-based sequence model, Nat. Methods 12 (2015) 931–934.
[30] Y. Wang, T. Liu, D. Xu, et al., Predicting DNA methylation state of CpG dinucleotide using genome topological features and deep networks, Sci. Rep. 6 (2016) 19598.
[31] H. Zeng, D.K. Gifford, Predicting the impact of non-coding variants on DNA methylation, Nucleic Acids Res. 45 (2017), e99.
[32] C. Angermueller, H.J. Lee, W. Reik, et al., DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning, Genome Biol. 18 (2017) 67.
[33] R. Xu, D. Wunsch II, R. Frank, Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (2007) 681–692.
[34] L.-Y. Chuang, J.-H. Tsai, C.-H. Yang, Operon prediction using particle swarm optimization and reinforcement learning, in: Proceedings of the TAAI, Hsinchu, Taiwan, 2010, pp. 366–372.
[35] C.G. Ralha, H.W. Schneider, M.E.M.T. Walter, et al., Reinforcement learning method for BioAgents, in: 11th Brazilian Symposium on Neural Network, Sao Paulo, Brazil, 2010, pp. 109–114.
[36] M.-I. Bocicor, G. Czibula, I.-G. Czibula, A reinforcement learning approach for solving the fragment assembly problem, in: Proceedings of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2011, pp. 191–198.
[37] M.J. Witteveen, Identification and elucidation of expression quantitative trait loci (eQTL) and their regulating mechanisms using decodive deep learning, Master thesis, 2014, pp. 1–17.
[38] J. Ma, M.K. Yu, S. Fong, et al., Using deep learning to model the hierarchical structure and function of a cell, Nat. Methods 15 (2018) 290–298.
[39] B.L. Gudenas, L. Wang, Prediction of LncRNA subcellular localization with deep learning from sequence features, Sci. Rep. 8 (2018) 16385.
[40] G. Ditzler, R. Polikar, G. Rosen, Multi-layer and recursive neural networks for metagenomic classification, IEEE Trans. Nanobioscience 14 (2015) 608–616.
[41] J. Lin, K.-C. Wong, Off-target predictions in CRISPR-Cas9 gene editing using deep learning, Bioinformatics 34 (2018) i656–i663.
[42] F. Liu, C. Ren, H. Li, et al., De novo identification of replication-timing domains in the human genome by deep learning, Bioinformatics 32 (2015) 641–649.
[43] X.-N. Fan, S.-W. Zhang, lncRNA-MFDL: identification of human long non-coding RNAs by fusing multiple features and using deep learning, Mol. Biosyst. 11 (2015) 892–897.
[44] S. Zhang, H. Hu, T. Jiang, et al., TITER: predicting translation initiation sites by deep learning, Bioinformatics 33 (2017) i234–i242.
[45] G. Arango-Argoty, E. Garner, A. Pruden, et al., DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data, Microbiome 6 (2018) 23.
[46] Y. Li, W. Shi, W.W. Wasserman, Genome-wide prediction of cis-regulatory regions using supervised deep learning methods, BMC Bioinf. 19 (2018) 202.
[47] J.J.A. Armenteros, K.D. Tsirigos, C.K. Sønderby, et al., SignalP 5.0 improves signal peptide predictions using deep neural networks, Nat. Biotechnol. 37 (2019) 420–423.
[48] R.B. Silverman, M.W. Holladay, The Organic Chemistry of Drug Design and Drug Action, third ed., Academic Press, San Diego, CA, 2014, pp. 123–202.
[49] T. Aoyama, Y. Suzuki, H. Ichikawa, Neural networks applied to pharmaceutical problems. III. Neural networks applied to quantitative structure-activity relationship (QSAR) analysis, J. Med. Chem. 33 (9) (1990) 2583–2590.
[50] J. Taskinen, J. Yliruusi, Prediction of physicochemical properties based on neural net-work modeling, Adv. Drug Deliv. Rev. 55 (2003) 1163–1183.
[51] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, T. Sainath, Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Process. Mag. 29 (2012) 82–97.
[52] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, Nevada, USA, 2012, 2012, pp. 1097–1105.

[53] R. Zhang, J. Li, J. Lu, R. Hu, Y. Yuan, Z. Zhao, Using deep learning for compound selectivity prediction, Curr. Comput. Aided Drug Des. 12 (1) (2016) 5–14.
[54] J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl, V. Svetnik, Deep neural nets as a method for quantitative structure–activity relationships, J. Chem. Inf. Model. 55 (2) (2015) 263–274.
[55] J.C. Pereira, E.R. Caffarena, C.N. dos Santos, Boosting docking-based virtual screening with deep learning, J. Chem. Inf. Model. 56 (12) (2016) 2495–2506.
[56] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, A. Zhavoronkov, The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology, Oncotarget 8 (7) (2016) 10883–10890, https://doi.org/10.18632/oncotarget.14073.
[57] T.B. Hughes, G.P. Miller, S.J. Swamidass, Modeling epoxidation of drug-like molecules with a deep machine learning network, ACS Cent. Sci. 1 (4) (2015) 168–180, https://doi.org/10.1021/acscentsci.5b00131.
[58] L. Wang, Z.H. You, X. Chen, S.X. Xia, F. Liu, X. Yan, Y. Zhou, K.J. Song, A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network, J. Comput. Biol. 25 (3) (2018) 361–373.
[59] M. Shin, D. Jang, H. Nam, K.H. Lee, D. Lee, Predicting the absorption potential of chemical compounds through a deep learning approach, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (2) (2018) 432–440.
[60] K.Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, P. Zhang, Interpretable Drug Target Prediction Using Deep Neural Representation, IJCAI, 2018, pp. 3371–3377.
[61] H.R. Hassanzadeh, M.D. Wang, DeeperBind: enhancing prediction of sequence specificities of DNA binding proteins, in: Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); Shenzhen, China, 2016, pp. 178–183.
[62] V. Golkov, M.J. Skwark, A. Mirchev, G. Dikov, A.R. Geanes, J. Mendenhall, J. Meiler, D. Cremers, 3D deep learning for biological function prediction from physical fields, arXiv (2017). preprint arXiv: 1704.04039.
[63] A.S. Schwartz, G.J. Hannum, Z.R. Dwiel, M.E. Smoot, A.R. Grant, J.M. Knight, et al., Deep semantic protein representation for annotation, discovery, and engineering, BioRxiv (2018), 365965.
[64] M. Liang, Z. Li, T. Chen, J. Zeng, Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach, IEEE/ACM Trans. Comput. Biol. Bioinform. 12 (4) (2015) 928–937.
[65] B. Alipanahi, A. Delong, M.T. Weirauch, B.J. Frey, Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning, Nat. Biotechnol. 33 (8) (2015) 831–838.
[66] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, A. Zhavoronkov, Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data, Mol. Pharm. 13 (7) (2016) 2524–2530.
[67] K. Chaudhary, O.B. Poirion, L. Lu, L.X. Garmire, Deep learning-based multi-omics integration robustly predicts survival in liver cancer, Clin. Cancer Res. 24 (6) (2018) 1248–1259.
[68] Y. Xiao, J. Wu, Z. Lin, X. Zhao, A deep learning-based multi-model ensemble method for cancer prediction, Comput. Methods Programs Biomed. 153 (2018) 1–9.
[69] M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, H. Lu, Deep-learning-based drug–target interaction prediction, J. Proteome Res. 16 (4) (2017) 1401–1409, https://doi.org/10.1021/acs.jproteome.6b00618.
[70] S. Kwon, S. Yoon, DeepCCI: end-to-end deep learning for chemical–chemical interaction prediction, in: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Boston, MA, USA, 2017, pp. 203–212.
[71] H.C. Lee, H.G. Ryu, E.J. Chung, C.W. Jung, Prediction of bispectral index during target-controlled infusion of propofol and remifentanil a deep learning approach, Anesthesiology 128 (3) (2018) 492–501.
[72] G. Zhao, X. Zhuang, X. Wang, W. Ning, Z. Li, J. Wang, et al., Data-driven traditional chinese medicine clinical herb modeling and herb pair recommendation, in: Proceedings of the 7th IEEE International Conference on Digital Home (ICDH 2018), Guilin Guangxi, China, 2018, pp. 160–166.

[73] Y. Zheng, D. Comaniciu, Marginal space learning, in: Marginal Space Learning for Medical Image Analysis, Springer, 2014, pp. 25–65.
[74] F.C. Ghesu, E. Krubasik, B. Georgescu, V. Singh, Y. Zheng, J. Hornegger, et al., Marginal space deep learning: efficient architecture for volumetric image parsing, IEEE Trans. Med. Imaging 35 (2016) 1217–1228.
[75] B. Bier, M. Unberath, J.-N. Zaech, J. Fotouhi, M. Armand, G. Osgood, et al., X-ray-transform invariant anatomical landmark detection for pelvic trauma surgery, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Medical Image Computing and Computer Assisted Intervention—MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 55–63.
[76] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hasoul, R. Ben-Ari, E. Barkan, A region based convolutional network for tumor detection and classification in breast mammography, in: Deep Learning and Data Labeling for Medical Applications, Springer, 2016, pp. 197–205.
[77] M. Aubreville, M. Krappmann, C. Bertram, R. Klopfleisch, A. Maier, A guided spatial transformer network for histology cell differentiation, in: Association TE (Ed.), Eurographics Workshop on Visual Computing for Biology and Medicine, 2017, pp. 21–25.
[78] M. Aubreville, M. Stöve, N. Oetter, G.M. de Jesus, C. Knipfer, H. Neumann, et al., Deep learning-based detection of motion artifacts in probe-based confocal laser endomicroscopy images, Int. J. Comput. Assist. Radiol. Surg. (2018), https://doi.org/10.1007/s11548-018-1836-1.
[79] H.R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E.B. Turkbey, et al., DeepOrgan: multi-level deep convolutional networks for automated pancreas segmentation, in: International Conference on Medical Image Computing, Computer-Assisted Intervention, Springer, 2015, pp. 556–564.
[80] P. Moeskops, M.A. Viergever, A.M. Mendrik, L.S. de Vries, M.J. Benders, I. Isgum, Automatic segmentation of MR brain images with a convolutional neural network, IEEE Trans. Med. Imaging 35 (2016) 1252–1261.
[81] S. Chen, X. Zhong, S. Hu, S. Dorn, M. Kachelriess, M. Lell, et al., in: B. van Ginneken, M. Welling (Eds.), Automatic Multi-Organ Segmentation in Dual Energy CT Using 3D Fully Convolutional Network, MIDL, 2018.
[82] K. Breininger, S. Albarqouni, T. Kurzendorfer, M. Pfister, M. Kowarschik, A. Maier, Intraoperative stent segmentation in X-ray fluoroscopy for endovascular aortic repair, Int. J. Comput. Assist. Radiol. Surg. 13 (2018).
[83] J.J. Nirschl, A. Janowczyk, E.G. Peyster, R. Frank, K.B. Margulies, M.D. Feldman, et al., Deep learning tissue segmentation in cardiac histopathology images, in: Deep Learning for Medical Image Analysis, Elsevier, 2017, pp. 179–195.
[84] I. Middleton, R.I. Damper, Segmentation of magnetic resonance images using a combination of neural networks and active contour models, Med. Eng. Phys. 26 (2004) 71–86.
[85] W. Fu, K. Breininger, R. Schaffert, N. Ravikumar, T. Würfl, J. Fujimoto, et al., Frangi-Net: a neural network approach to vessel segmentation, in: A. Maier, T. Deserno, H. Handels, K.H. Maier-Hein, C. Palm, T. Tolxdorff (Eds.), Bildverarbeitung Für Die Medizin, Springer, 2018, pp. 341–346.
[86] R.P. Poudel, P. Lamata, G. Montana, Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation, in: Reconstruction, Segmentation, and Analysis of Medical Images, Springer, 2016, pp. 83–94.
[87] S. Andermatt, S. Pezold, P. Cattin, Multi-dimensional gated recurrent units for the segmentation of biomedical 3D-data, in: Deep Learning and Data Labeling for Medical Applications, Springer, 2016, pp. 142–151.
[88] G. Wu, M. Kim, Q. Wang, B.C. Munsell, D. Shen, Scalable high-performance image registration framework by unsupervised deep feature representations learning, IEEE Trans. Biomed. Eng. 63 (2016) 1505–1516.
[89] R. Schaffert, J. Wang, P. Fischer, A. Borsdorf, A. Maier, Metric-driven learning of correspondence weighting for 2-D/3-D image registration, in: German Conference on Pattern Recognition (GCPR), 2018.
[90] S. Miao, J.Z. Wang, R. Liao, Convolutional neural networks for robust and real-time 2-D/3-D registration, in: Deep Learning for Medical Image Analysis, Elsevier, 2017, pp. 271–296.

[91] I. Diamant, Y. Bar, O. Geva, L. Wolf, G. Zimmerman, S. Lieberman, et al., Chest radiograph pathology categorization via transfer learning, in: Deep Learning for Medical Image Analysis, Elsevier, 2017, pp. 299–320.
[92] J.R. De Fauw, B. Ledsam, S. Romera-Paredes, N. Nikolov, S. Tomasev, H. Blackwell, et al., Clinically applicable deep learning for diagnosis and referral in retinal disease, Nat. Med. 24 (2018) 1342.
[93] M. Aubreville, C. Knipfer, N. Oetter, C. Jaremenko, E. Rodner, J. Denzler, et al., Automatic classification of cancerous tissue in laser endomicroscopy images of the oral cavity using deep learning, Sci. Rep. 7 (2017), 41598–017.
[94] G. Carneiro, J. Nascimento, A.P. Bradley, Deep learning models for classifying mammogram exams containing unregistered multi-view images and segmentation maps of lesions, in: Deep Learning for Medical Image Analysis, Elsevier, 2017, pp. 321–339.
[95] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, et al., Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (2017) 115.
[96] J. Wu, I. Yildirim, J.J. Lim, B. Freeman, J. Tenenbaum, Galileo: perceiving physical object properties by integrating a physics engine with deep learning, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, 2015, pp. 127–135.
[97] M. Chu, N. Thuerey, Data-driven synthesis of smoke flows with CNN based feature descriptors, ACM Trans. Graph. 36 (2017) 69.
[98] F. Meister, T. Passerini, V. Mihalef, A. Tuysuzoglu, A. Maier, T. Mansi, Towards fast biomechanical modeling of soft tissue using neural networks, in: Medical Imaging Meets NeurIPS Workshop at 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018.
[99] J. Maier, Y. Berker, S. Sawall, M. Kachelrieß, Deep scatter estimation (DSE): feasibility of using a deep convolutional neural network for real-time X-ray scatter prediction in cone-beam CT, in: Medical Imaging 2018: Physics of Medical Imaging, vol. 10573, International Society for Optics and Photonics, 2018, p. 105731L.
[100] M. Unberath, J.-N. Zaech, S.C. Lee, B. Bier, J. Fotouhi, M. Armand, et al., DeepDRR—a catalyst for machine learning in fluoroscopy-guided procedures, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Medical Image Computing and Computer Assisted Intervention—MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 98–106.
[101] F. Horger, T. Würfl, V. Christlein, A. Maier, Towards arbitrary noise augmentation—deep learning for sampling from arbitrary probability distributions, in: International Workshop on Machine Learning for Medical Image Reconstruction, Springer, 2018, pp. 129–137.
[102] A. Maier, F. Schebesch, C. Syben, T. Würfl, S. Steidl, J.-H. Choi, et al., Precision learning: towards use of known operators in neural networks, in: J.K.T. Tan (Ed.), 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 183–188.
[103] X. Han, MR-based synthetic CT generation using a deep convolutional neural network method, Med. Phys. 44 (2017) 1408–1419.
[104] B. Stimpel, C. Syben, T. Würfl, K. Mentl, A. Dörfler, A. Maier, MR to X-ray projection image synthesis, in: F. Noo (Ed.), Proceedings of the 5th International Conference on Image Formation in X-ray Computed Tomography (CT-Meeting), 2018, pp. 435–438.
[105] F. Schiffers, Z. Yu, S. Arguin, A. Maier, Q. Ren, Synthetic fundus fluorescein angiography using deep neural networks, in: A. Maier, T.M. Deserno, H. Handels, K.H. Maier-Hein, C. Palm, T. Tolxdorff (Eds.), Bildverarbeitung für die Medizin 2018, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 234–238.
[106] J.P. Cohen, M. Luck, S. Honari, Distribution matching losses can hallucinate features in medical image translation, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Medical Image Computing and Computer Assisted Intervention—MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 529–536.
[107] J. Andreu-Perez, C.C.Y. Poon, R.D. Merrifield, S.T.C. Wong, G.Z. Yang, Big data for health, IEEE J. Biomed. Health Inform. 19 (4) (2015) 1193–1208.
[108] G.-Z. Yang, D.R. Leff, Big data for precision medicine, Engineering 1 (3) (2015) 277. [Online]. Available http://engineering.org.cn/EN/abstract/article_12197.shtml.

[109] H. Shin, L. Lu, L. Kim, A. Seff, J. Yao, R.M. Summers, Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation, in: CoRR, vol. abs/1505.00670, 2015. [Online]. Available http://arxiv.org/abs/1505.00670.
[110] Z. Liang, G. Zhang, J.X. Huang, Q.V. Hu, Deep learning for healthcare decision making with EMRS, in: Proc. Int. Conf. Bioinformat. Biomed, Nov 2014, pp. 556–559.
[111] E. Putin, et al., Deep biomarkers of human aging: application of deep neural networks to biomarker development, Aging 8 (5) (2016) 1–021.
[112] L. Nie, M. Wang, L. Zhang, S. Yan, B. Zhang, T.S. Chua, Disease inference from health-related questions via sparse deep learning, IEEE Trans. Knowl. Data Eng. 27 (8) (2015) 2107–2119.
[113] R. Miotto, L. Li, B.A. Kidd, J.T. Dudley, Deep patient: an unsupervised representation to predict the future of patients from the electronic health records, Sci. Rep. 6 (2016) 1–10.
[114] J. Futoma, J. Morris, J. Lucas, A comparison of models for predicting early hospital readmissions, J. Biomed. Inform. 56 (2015) 229–238.
[115] Z.C. Lipton, D.C. Kale, C. Elkan, R.C. Wetzel, Learning to diagnose with LSTM recurrent neural networks, in: CoRR, vol. abs/1511.03677, 2015. [Online]. Available http://arxiv.org/abs/1511.03677.
[116] S. Mehrabi, et al., Temporal pattern and association discovery of diagnosis codes using deep learning, in: Proc. Int. Conf. Healthcare Informat, Oct. 2015, pp. 408–416.
[117] Z. Che, S. Purushotham, R. Khemani, Y. Liu, Distilling knowledge from deep networks with applications to healthcare domain, arXiv:1512.03542 (2015).
[118] T. Huang, L. Lan, X. Fang, P. An, J. Min, F. Wang, Promises and challenges of big data computing in health sciences, Big Data Res. 2 (1) (2015) 2–11.
[119] B.T. Ong, K. Sugiura, K. Zettsu, Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting pm2. 5, Neural Comput. Appl. 27 (2015) 1–14.
[120] L. Zhao, J. Chen, F. Chen, W. Wang, C.-T. Lu, N. Ramakrishnan, Simnest: social media nested epidemic simulation via online semi supervised deep learning, in: Proc. IEEE Int. Conf. Data Mining, 2015, pp. 639–648.
[121] R.L. Kendra, S. Karki, J.L. Eickholt, L. Gandy, Characterizing the discussion of antibiotics in the Twittersphere: what is the bigger picture? J. Med. Internet Res. 17 (6) (2015), e154.
[122] V.R.K. Garimella, A. Alfayad, I. Weber, Social media image analysis for public health, in: Proc. CHI Conf. Human Factors Comput. Syst, 2016, pp. 5543–5547. [Online]. Available https://doi.org/10.1145/2858036.2858234.
[123] N. Phan, D. Dou, B. Piniewski, D. Kil, Social restricted Boltzmann machine: human behavior prediction in health social networks, in: Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Aug. 2015, pp. 424–431.
[124] E. Horvitz, D. Mulligan, Data, privacy, and the greater good, Science 349 (6245) (2015) 253–255.
[125] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing Higher Layer Features of a Deep Network, Univ. Montreal, Montreal, QC, Canada, 2009. Tech. Rep. 1341.
[126] D. Erhan, A. Courville, Y. Bengio, Understanding Representations Learned in Deep Architectures, Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, 2010. Tech. Rep. 1355.
[127] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[128] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in: Proc. IEEE Conf. Comput. Vis. Pattern Recogn, 2015, pp. 427–436.
[129] C. Szegedy, et al., Intriguing properties of neural networks, in: CoRR, vol. abs/1312.6199, 2013. [Online]. Available http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SzegedyZSBEGF13.

CHAPTER 12

Autonomous experimentation in nanotechnology

Aldair E. Gongora[a], Verda Saygin[a], Kelsey L. Snapp[a], and Keith A. Brown[a,b]
[a]Department of Mechanical Engineering, Boston University, Boston, MA, United States
[b]Physics Department and Division of Materials Science & Engineering, Boston University, Boston, MA, United States

1. Introduction

Nanotechnology is a field that is characterized by sophisticated and often expensive experiments, highly complex materials defined by a huge number of potential physical variables, and useful computational tools that can predict some, but not all, important properties. As such, experiments are often a bottleneck in progress in nanoscience. Two concepts have emerged that facilitate this process, namely automation and algorithmic experimental selection. Automation allows researchers to increase the throughput and repeatability of experiments, while algorithmic experimental selection represents an approach to systematize the selection of experiments to more efficiently use each experiment to achieve a given experimenter-defined goal. **Autonomous experimentation (AE)** represents the convergence of automation and experiment selection powered by machine learning in which robotic platforms perform experiments that are selected by machine learning algorithms in an iterative fashion. This is in contrast with high-throughput experimentation (HTE) where the focus is on using automation to increase the number of experiments performed rather than using all available information to select each experiment in a closed-loop fashion. This chapter seeks to review progress in the field of AE with particular relevance to nanotechnology and discuss the unique opportunities that arise when considering the applications of AE in nanoscience.

It is worth emphasizing that AE as a paradigm encompasses many terms that have emerged, including self-driving labs [1], materials acceleration platforms [2], robot scientists [3–5], and autonomous research systems (ARES) [6]. However they are named, for a system to be capable of AE, it must be able to iteratively perform the following tasks without human intervention of any kind [7]:

- Create hypotheses
- Propose an experiment to test the hypothesis
- Conduct the experiment
- Record the results

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00012-3
Copyright © 2023 Elsevier Inc.
All rights reserved.

- Analyze the results
- Update the hypothesis

In addition to noting what is required, it is equally important to be explicit about what is *not* required. First, there are no specific hardware requirements, such as having a robotic arm or being a mobile platform. While many implementations of AE have used robots or other methods to make their systems more versatile or to allow them to use lab equipment designed for human use, this is not required. Second, there is no limitation on the type of scientific knowledge that can be explored. Some AE implementations have focused on traditional scientific knowledge, such as fundamental structure-property relationships in materials science. Others, however, have focused on procedural knowledge such as how to manufacture materials or synthesize compounds. Finally, there is often some focus on the scientific process itself such as determining the accuracy of measurement or learning how often a component needs to be replaced.

In the following sections, we detail the development, existing applications, and future opportunities for AE in nanotechnology. We begin with (Section 2) a discussion of the development of AE capabilities with a particular focus on three complementary facets of the technology, namely automation, machine learning for experiment selection, and broader artificial intelligence for knowledge generation. Next, we discuss (Section 3) case studies of AE systems in use for nanoscience including the synthesis of carbon nanotubes, the optoelectronic properties of organic thin films, and the synthesis of nanoparticles. Finally, we discuss (Section 4) some existing and emerging platform technologies for AE exploration in nanoscience that could enable entire autonomous labs to be realized in microscopic domains including both nano- and microscale fluid handling and scanning probe technologies. We conclude with a discussion of noteworthy future challenges and opportunities facing the exploitation of AE in nanoscience.

2. Development of AE capabilities

Although the rapid proliferation of many AE systems in recent years is striking, many major advances over the past half century have set the stage for the rapid development and exploitation of AE systems. In particular, these enabling advances can be broadly broken up into innovations in (1) automation and robotics; (2) domain knowledge, discovery systems, and artificial intelligence; and (3) machine learning, optimization, and data analysis. These categories are crosscutting with the iterative stages that define an AE cycle (Fig. 1). Interestingly, the degree to which each of these systems must be tailored for the particular scientific field varies widely. For instance, the methods required to tailor a robotic system to automate a specific process vary greatly between, for example, chemistry and mechanics. In contrast, many machine learning methods can be applied across scientific disciplines with little or no modification. As a middle ground, many of the techniques to leverage domain knowledge can employ processes that apply to multiple disciplines but differ in their specific implementations.

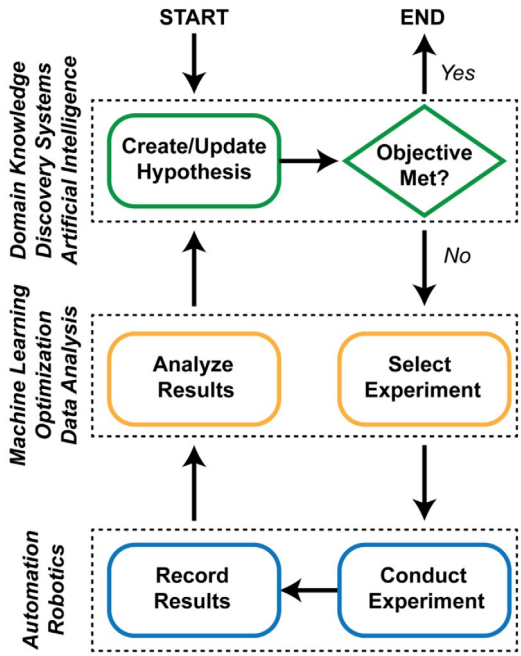


Fig. 1 AE systems are often conceptualized as loops. Here, we break the traditional learning loop into three distinct sections. Domain knowledge/discovery systems/artificial intelligence creates or updates a hypothesis based on results and ultimately draws conclusions from the research. Machine learning/optimization/data analysis selects experiments and interprets results, often through statistical models of the system of interest. Finally, automation/robotics performs the experiments and analyzes the results, often through a network of connected lab equipment.

2.1 Automation and robotics

As technology has progressed, laboratory equipment has become more complex and automated. The field of chemistry is a great example of this progress. Automated synthesis and characterization of molecules date to the early 1960s. By 1971, researchers from Purdue University had designed a computer-controlled system that integrated sensors and actuators to control the temperature of the system, including the capability to add four different reagents in controlled quantities and stir the resulting solution [8]. When completed, the system could automatically characterize the reaction rates of the resulting solution using spectrophotometry.

Synthetic systems have become more sophisticated, particularly driven by biotech and pharmaceutical companies' need to screen millions of compounds for potential clinical relevance. High-throughput screening (HTS) became widespread in the 1990s, leveraging microtiter plates (MTP) with 96 wells that were ultimately miniaturized to include 384 and then 1536 individual wells [9]. The advantages of increased well density were twofold. First, it allowed more experiments to be done per plate,

increasing throughput. Second, it decreased the costs of supplies, as the sample volumes decreased from the 100–200 μL range to <50 μL. Plates as dense as 3456 wells (1–2 μL) have been used, but higher costs and difficulties working at such small volumes have limited the widespread use of the 3456-well format. Facing physical limits on the density of wells and diminishing returns from miniaturization, many researchers shifted focus in the mid-2000s to more tightly integrated systems. Initial HTS systems focused solely on the screening aspect of the research. However, later efforts focused on the entire process, including assay preparation, counter screening, and hit validation. These high-throughput commercial systems are ripe for conversion into AE systems and have been so used by both industry and researchers [10,11]. It is worth noting, however, that the conventional plate-based approach performs a plates worth of experiments in parallel, which limits the feedback cycle of using new data to inform subsequent experimental selections.

In addition to increasing the throughput of chemical synthesis, work has been done to make systems that are more robust and versatile. Two methods have been utilized to accomplish this. The first method has been to create modular components that can be used for many different applications. An early example of this approach is a system of modular lab equipment developed by Legrand and Foucard in the 1970s [12]. This system combined various sensors, actuators, and a control system to manage chemical reactions. The automation allowed multiday reactions to be run continuously and allowed fine control of parameters, increasing yield while decreasing experiment time relative to manual experimentation. More sophisticated attempts followed. For example, in 2018 researchers at MIT created a continuous-flow chemical synthesis system that had five different bays that could host different modules [13]. This allows a single machine to be configured for different synthesis requirements. The second major method that has been used to make experimental systems more robust and versatile has focused on developing robotic systems that use regular laboratory equipment designed for human researchers. Advances in robotics have made it cheaper and easier to design robotic systems that both conduct experiments and record results. This means that researchers can often use off-the-shelf equipment, both for robotic manipulation of samples and for conducting experiments, thus decreasing the burden of designing an entire AE system from scratch. The first robot-based AE system, named "Adam," was developed in 2004. The system unified the capabilities of a built-in freezer, liquid handlers, incubators, plate readers, robot arms, a centrifuge, bar code readers, and environmental sensors [4]. Four separate computers were needed to coordinate all of its actions. Although the researchers determined that the capital and maintenance expenses were not cost-effective at the time, Adam was able to demonstrate impressive increases in productivity [7]. While studying the genetics of yeast, researchers observed 3 and 100 × reductions of experiments to obtain comparable results to human researchers [3]. This result, and its successor "Eve," represented a seminal and pioneering work in the field.

While Adam leveraged commercially available automation hardware that had been developed for the biological and pharmaceutical communities, other disciplines lacked existing widespread automation solutions. Thus, further development of AE systems experienced somewhat of a lull between the innovation that constituted Adam and the mid-2010s. When AE systems started emerging, systems most often included bespoke hardware/automation solutions that were custom developed for the needs of the material system. For example, the ARES system, which constituted the first AE system in the materials space, was developed to automate carbon nanotube (CNT) growth by chemical vapor deposition (CVD) [6]. To realize closed-loop experimentation, the team at the Air Force Research Laboratory had to develop a custom process for automating both the synthesis and characterization of CNTs. In particular, they utilized a 532 nm laser to selectively heat 10 μm cylinders that were arrayed on a thermally insulating wafer. In addition to controlling the temperature of the pillar to facilitate synthesis, the laser was also used as an excitation source for Raman spectroscopy, allowing the growth rates of the CNT to be monitored in-situ. In addition to the temperature of the reaction, the system can also control the pressure and the gas composition of the reaction environment. This system is discussed further as a case study in Section 3.1.

To provide another example of a bespoke automation solution, this time at the macroscale, the Intelligent Towing Tank was designed to research vortex-induced vibrations caused by separated flow around a structure [14]. To accomplish this task, researchers mounted a carriage onto rails running the length of a 10-m-long tank of water. The carriage was able to travel at a speed of up to 1.5 m/s along the length of the tank, while also allowing lateral and rotational movements. Like ARES, these custom hardware solutions allowed the team to automate the research pipeline and include it into a broader AE system.

Despite the proliferation of AE systems, each is commonly built as a unique, one-off system. A major challenge presented by this is that it requires researchers to be experts not only in their own area of research, but also in programming, networking, and robotics, to say nothing about increasing the capital cost associated with realizing an AE system. For AE systems to become widespread, robust and customizable hardware platforms are needed that are easy to create, run, and maintain.

One approach for facilitating the development of hardware compatible with AE systems is to integrate robots into already functioning lab environments with minimal changes to the lab equipment and layout. Researchers at the University of Liverpool demonstrated this capability by outfitting a KUKA Mobile Robot with a gripper necessary to perform a wide range of chemistry experiments [15]. The robot used both laser positioning for laboratory navigation and a touch feedback system for fine positioning at each of the six stations. Although mobile robots would certainly still have limitations and need some accommodations to the lab environment, they could be easier to implement in shared labs and have lower initial capital costs than custom systems like "Adam."

Another method to facilitate the widespread adoption of AE systems is to make it easier to control and coordinate the research process. ChemOS consists of six software modules that facilitate and consolidate control over all processes of an AE system [16]. It is designed to allow control over multiple robots or experimental systems simultaneously and can coordinate robots in different geographical locations. It is easy to see how such a system could not only make it easier for individual researchers to set up AE systems in the lab but also for AE systems at different universities to work together in coordinated research plans.

Despite efforts to make the hardware used in AE systems more broadly applicable, it seems unlikely that there will ever be a single system that is used for all disciplines. A robot designed to use a human-centric lab may decrease the initial capital costs, but for most applications, a system designed for automation from the start is likely to be more efficient and more reliable. Furthermore, the uniqueness of the Adam, ARES, and the Intelligent Towing Tank show the vast differences in requirements for different research areas. As AE systems continue to proliferate in other areas of science, it is likely that the machinery necessary to automate experiments and collect results will continue to vary greatly by research discipline. Therefore, a key barrier to this proliferation is the engineering work necessary to automate these research tasks. Given the acceleration in development that is possible when leveraging established automation solutions, researchers should look to off-the-shelf solutions and promote a culture of open-source hardware development.

2.2 Domain knowledge, discovery systems, and artificial intelligence

Although automation can improve throughput and decrease researcher hours, automation alone is not enough to create an AE system. The system needs to be able to interpret the scientific goal, develop a testable hypothesis, and then update that hypothesis when new results are returned. To some extent, AE systems will likely always have some human input, at the least in defining the scientific goal at the highest level. In a sense, how human researchers frame that goal, and the limits that are placed on the available parameter space, often contain interesting information and constitute a hypothesis in and of itself. For example, when building a machine to grow CNTs or synthesize molecules, the very capabilities designed into the AE system contain decades of wisdom on how to best accomplish these goals and embody the theory that the chosen conditions will encompass interesting or useful outcomes. Additionally, AE systems are often given search spaces, and these spaces usually contain knowledge about which parameters and values are considered likely to lead to success.

Additionally, there have been attempts to systematically incorporate domain knowledge into AE systems, allowing them to take advantage of the vast wealth of scientific knowledge that has been previously codified. For example, simulation by solving established governing equations offers a way to incorporate domain knowledge into a

problem. In mechanics, finite element analysis (FEA) can provide insights into how a mechanical structure will perform. By incorporating FEA into the AE system, a preliminary estimate of the space can be achieved. Physical testing can then refine and confirm the simulated predictions, rather than starting with a blank slate. Simulation can also be an effective input to an AE system even when simulation cannot accurately predict the ultimate objective, as recently demonstrated by the Bayesian experimental autonomous researcher (BEAR) from Boston University [17]. While running a campaign to maximize toughness, the BEAR used FEA simulations of yield force to narrow the searchable space. Although it was not able to simulate toughness directly or find high performing structures with simulation alone, it was able to rule out structures that could not perform well because of yield forces that were too high. Further, the simulated yield force was also used to transform experimental data so that the AE system would instead learn a measure of the system ductility. This example showed that it is possible to use simulations of related systems in a systematic way to accelerate AE.

Although simulation can be a powerful tool, it can be time intensive and require expensive supercomputers. Thus, coordinated sharing of simulated data is of high value. For instance, the Materials Project is an effort to systematically create and share high-quality simulation data for materials research [18]. The Materials Project uses supercomputer clusters at several national laboratories to simulate key materials properties for hundreds of thousands of materials. This information is then stored in a standardized format and accessible through an API. The project has even created standardized Python objects to help researchers work with the information contained in the database. By creating a system to create and share domain knowledge in a structured way, the Materials Project provides valuable tools to create and improve AE systems.

Ultimately, it is valuable to push the depth to which AE systems can interpret experimental results. Part of the motivation of this comes from early attempts in the 20th century to develop computer programs that could reason and think like a scientist. One such example, called Dendral, was developed by scientists from Stanford. The system originally used rules of molecular fragmentation that scientists defined to help understand data from mass spectroscopy [19]. As an example of increasing the level of reasoning employed by the system, researchers later augmented Dendral to discover the fragmentation rules on its own. This key shift meant that the system was not only finding primary scientific knowledge (understanding the data from mass spectroscopy), but also developing a broader understanding of the system that extended beyond the specific data being analyzed. Other systems followed using increasingly sophisticated collections of heuristics. Some of these works include BACON for the rediscovery of empirical laws such as conservation of momentum or Snell's law of refraction [20], ABACUS for identifying more complicated relationships such as free-falling bodies in viscous liquids and analysis of chemical compounds [21], KEKADA for reproducing the heuristics and learning process that Hans Krebs used to discover the Urea cycle [22], FAHRENHEIT for identifying test

procedures and measurement error handling [23], and Eureqa that rediscovered of Newton's 2nd law by studying pendula and spring systems [24].

In addition to these focused systems, attempts to create more robust and broadly applicable systems are under development. For example, IBM's Watson made headlines when it beat the best human Jeopardy players in 2011 [25]. The underlying system, DeepQA, uses natural language processing to create order from unstructured information. IBM has since adapted this system to work on medical diagnosis, but there is no fundamental limitation that would prevent DeepQA or other systems from analyzing other areas of scientific research. Ultimately, the goal is to develop a system that can parse scientific literature, find potential research opportunities, run the necessary experiments, and publish the results without human intervention. To that end, the Nobel Turing Challenge offers the ultimate goal: to create an AE system that can produce scientific insight worthy of a Nobel prize [26].

2.3 Machine learning, optimization, and data analysis

Finally, a great amount of work has been done in selecting the most efficient experiments for each iteration of an AE cycle. AE systems often work in multidimensional spaces where the set of possible experiments numbers in at least the millions. Furthermore, experimental results necessarily include stochastic variations with unknown properties, further complicating the process of drawing conclusions from sparse data. When facing extremely large spaces with multiple dimensions, human researchers tend to utilize brute force methodologies, such as searching on a regular grid or one factor at a time (OFAT). However, both methods tend to be inefficient in high-dimensional spaces and are prone to identifying local optima rather than global optima. Machine learning (ML) can provide a key insight in this process by providing statistical models of the system under study that allow smarter choices in experiment selection, decreasing the number of experiments necessary to achieve a researcher's goals. Although there are countless methods for selecting experiments, one overarching characteristic is that selecting experiments sequentially such that each experiment utilizes all available data is crucial for the efficient exploration of complex parameter spaces.

To address the efficient sequential selection of experiments to build predictive models, the field of active learning has emerged within the broader field of machine learning. The field of active learning focuses on the closed-loop phenomenon of selecting actions or experiments that influence the data added to the training set [27]. A popular active learning algorithm that has received much popularity due to its success in optimizing unknown and expensive functions is Bayesian optimization (BO), which is comprised of two main components, the belief model and the decision policy [28–31]. The belief model is built using the data acquired and subsequent experiments are selected using a decision policy. The belief model may be generated using numerous

mathematical functions with Gaussian processes and neural networks being popular choices. The probabilistic nature of the belief model often enables the prediction of properties such as the mean and uncertainty of the space. To select the subsequent experiment or sets of experiments, the decision policy in the form of an acquisition function is utilized. Decision policies often balance competing goals of exploration (selecting experiments in areas with large uncertainty) and exploitation (selecting experiments in areas of the space that are likely to be high performing). To represent these goals, decision policies are formulated to be exploratory (e.g., a maximum variance, MV, decision policy), exploitative (e.g., pure exploitation decision policy), or to seek a balance between exploratory and exploitative behavior (e.g., expected improvement, EI). In addition to the success BO has seen in numerous applications, the availability of open-source software has facilitated the rapid adaptation and utilization of BO in numerous applications ranging from AE to hyper-parameter tuning of deep neural networks. In particular, the ML community has developed and maintained several open-source libraries and tools, such as ChemOS and Phoenics [29,32]. While fields such as chemistry have evidently led the way in software development and distribution, other fields seeking to adapt AE can certainly benefit from the plethora of freely available software and tutorials.

Two AE systems have sought to quantify the improvements that can be gained through efficient selection of tests. The first study looked at the formation of self-assembled inorganic molecules in a three dimension space [33]. The purpose was to map this space as completely as possible, finding where polyoxometalate molecules would self-assemble into large single crystals. Interestingly, the paper benchmarks their ML algorithm against both human experts (PhD students who were familiar with the relevant science) and against random selection. The AE system first used a random forest classifier to determine where crystallization was likely to occur based on an initial training set. It then probed the boundary of this region by selecting experiments where the classifier showed high levels of uncertainty. The ML algorithm improved its classifier accuracy by about 16 percentage points, while human scientists and random selection improved by about 8 and 0 percentage points, respectively. The machine algorithm aggressively probed the boundary where crystallization occurred while the human researchers used OFAT methods, which limited the breadth of their search. In another example, the BEAR benchmarked their Bayesian optimization (BO) approach against a grid search [34]. As a benchmark, the grid search consisted of 1800 tests at 600 distinct points in a four-dimensional test space. Then, six separate campaigns of 100 sequential experiments were run using two distinct decision policies: MV and EI. All six campaigns matched the performance of the grid search and five of the six campaigns (all three campaigns that utilized EI and two of the three campaigns that utilized MV) outperformed the grid method, despite using significantly fewer experiments.

These two examples not only highlight the possible advantages of machine learning, but also some of the diverse options available for further research. Traditional BO assumes

a batch size of one. However, the first study picked batches of 10 experiments at a time, while the BEAR did experiments both with singles batches and with 6 asynchronous experiments running simultaneously. Additionally, the BEAR examined two different acquisition functions: maximum variance (exploring the space) and expected improvement (find a minimum or maximum). Further optimization of the experiment selection process can be gained through careful study of these and related issues.

Finally, it is important to note that while much scientific research can be thought of as an optimization (i.e., a campaign with the goal to identify the best combination of parameters) or exploration (i.e., a campaign with the goal to learn the most information about parameter space) in a multidimensional space, this does not mean that the experimental selection techniques are easily transferable. For areas where research consists of exploring a well-defined multidimensional space, many machine learning techniques require no adjustments to move between very different fields of study. In contrast, other research areas, such as chemical synthesis, are not easily formulated as a search in a continuous, multidimensional space due to the combinatorial nature of molecular design together with the complexities of which materials are inherently stable or synthesizable. In this field and other related fields, black-box optimizers are poorly suited and the process of defining a suitable parameter space is a major component of the challenge of bringing AE systems to bear. Additional solutions include using similarity calculations to predict similar chemical characteristics, using functional groups to filter libraries, and using diverse drug libraries as initial starting points [35]. Along those lines, researchers recently demonstrated an autoencoder that converts SMILES strings (an ASCII-based representation of a chemical structure) into a continuous multidimensional space [36]. This space can then be used to linearly interpolate between two chemical compounds or to optimize using machine algorithms. Similar, clever solutions likely exist not only in chemical synthesis but also in a variety of other research fields.

2.4 Overview of the development of autonomous experimentation (AE) systems

While we have previously focused on discrete components of AE systems, namely, (1) automation and robotics; (2) domain knowledge, discovery systems, and artificial intelligence; and (3) machine learning, optimization, and data analysis, it must be acknowledged that functioning AE systems have been developed and applied in fields such as chemistry, biology, mechanics, and nanoscience. Even though the discussion of each AE system is beyond the scope of this paper, it is useful to put into context the chronological development of these systems (Fig. 2). The first AE system was developed in 2004 in the field of biology and called Adam, while the second AE system was 12 years later in 2016 in the field nanoscience and called ARES. Thereafter, articles describing the

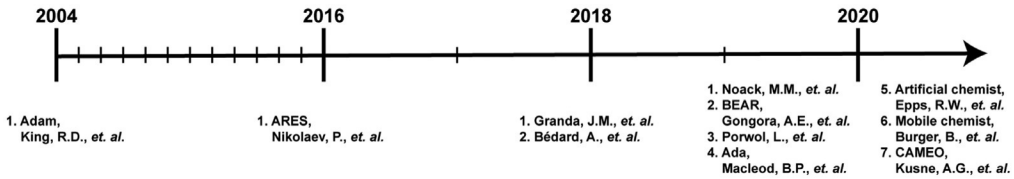


Fig. 2 Historical timeline on the development of autonomous experimentation (AE) systems [1,3,6,13,15,34,37–41].

development of two distinct systems in chemistry were published in 2018. Two years later in 2020, an astounding seven articles were published describing the development and application of AE systems in chemistry, mechanics, and nanoscience. While the intricacies on the development and application of each AE system merits abundant discussion, the role of AE systems in nanotechnology is discussed in the following section. The following section examines and details the development and application of AE systems, describes the enhancement observed using the AE system, and expounds on the impact and implications of AE systems in the field of nanoscience.

3. Case studies of AE in nanotechnology

Nanotechnology presents numerous opportunities for the utilization of AE. Still in its infancy, AE in nanotechnology has emerged and been shown to be an excellent approach for rapidly exploring vast and complex parameter spaces to optimize parameters and conditions and for knowledge discovery. In the field of nanoscience, AE systems have been successfully applied to study the synthesis of CNTs, thin films, and nanoparticles. Common to the development of AE systems for these applications was the compatibility of experimentation methods, such as processing and characterization at various stages of synthesis, with automation. Consequently, significant efforts were dedicated to the design and development of custom and modular hardware and software components to achieve reliable, high-throughput, and automated experimentation methods. As described above, the development of hardware for automated experimentation leveraged advancements in robotics for handling and transport requirements and while the active learning component of AE systems leveraged open-source libraries and tools from the ML community. In the case studies described in this section, AE systems have demonstrated efficient exploration of parameter spaces resulting in the reduction of necessary experiments by over an order of magnitude relative to manual approaches. In addition to reducing the number of experiments needed to find optimum parameters, the machine learning models used in AE have enabled researchers to deepen their understanding of the parameter space by extracting key insights motivating future research efforts to consider

the incorporation of physics knowledge, access to previously collected materials databases, and the exploration of alternative machine learning methods such as reinforcement learning.

3.1 Carbon nanotube synthesis

In the past two decades, CNTs have garnered substantial research interest due to their remarkable electrical, optical, mechanical, and electronic properties with potential transformative applications in electronics, medicine, energy, and composite materials [42–45]. CNTs are composed of concentrically arranged graphite sheets with diameters between 0.4 and 100 nm and are categorized as single-walled CNTs (SWNTs) or multiwall CNTs (MWNTs) [46–48]. SWNTs consist of a single graphene sheet rolled in a tubular geometry with typical diameters between 0.4 and 2 nm, while MWNTs include multiple concentric graphene sheets rolled in a tubular geometry and possess typical diameters between 1.5 and 100 nm. To synthesize CNTs, approaches such as laser ablation, arc discharge, plasma torch, and chemical vapor deposition (CVD) have been employed; however, the most frequently used technique currently is CVD as it enables the production of CNTS in large quantities with specified properties such as tube diameter, tube length, and tube alignment [49]. While CNTs present unprecedented properties and potential, the synthesis of CNTs using CVD is not sufficiently understood or controlled to a high enough degree to enable reliable defect-free high-throughput production [50,51]. The challenge stems from the vast number of synthesis conditions that must be considered when synthesizing CNTs using CVD [52], which is further amplified when considering that the experiments are painstakingly time-consuming and laborious. To address the challenge of traversing parameter space for optimum CNT synthesis conditions, AE has emerged as an approach for accelerating the understanding and production of reliable CNTs. In fact, exploring the development of AE in CNT growth provides an interesting window into AE development more generally as the sophistication and autonomy of systems to study CNT growth have been progressively improved throughout the years.

The development and application of AE in CNT synthesis began with the development of semiautomated and automated experimentation systems such as Robofurnace [53] and the adaptive rapid experimentation and in-situ spectroscopy system [54], respectively. Robofurnace was an automated CVD system for high throughput and reliable synthesis of thin films and was utilized to investigate the variability of CNT forest height and density by performing 26 identical CNT growth experiments. By using Robofurnace, the coefficients of variation of CNT forest height and density were reduced by 50% and 62%, respectively, when compared to manual experimentation. Additionally, Robofurnace was used to develop a rapid CVD recipe for CNT forest growth achieving a 10-fold improvement in CNT forest mass density compared to a benchmark recipe using a manual tube furnace. Moreover, Robofurnace enabled 16 or more experiments

per day in comparison to 2–4 experiments using a manual tube furnace. The adaptive rapid experimentation, on the other hand, was an automated experimentation system that possessed in-situ spectroscopy inside the CVD chamber and was able to conduct over 100 experiments per day with automated control and in-situ Raman characterization. From an experimental throughput perspective, this system was a major advance as it was estimated as performing 100 times more experiments per day than could be performed manually. By using this automated system, regions of selectivity toward SWNT and MWNT nucleation growth were mapped in the four-dimensional parameter space (temperature, partial pressure of H_2, partial pressure of C_2H_4, and partial pressure of H_2O) using statistical linear regression analysis. These works present the advantages of automated experimentation to increase experimental throughput and determine complex relationships between input parameters and performance, and motivate the inclusion of active learning to circumvent the need for human intervention in experimental design.

While the previously discussed studies demonstrate the utility of automation for CNT synthesis, efforts to further increase the pace of research in CNT synthesis led to the development of an AE system for CNT synthesis, namely an autonomous research system (ARES) [6]. ARES was reported to be capable of designing, executing, and analyzing its own experiments to learn synthesis conditions for CNT growth. By combining the previously developed adaptive rapid experimentation and in-situ spectroscopy system with ML, ARES was able to autonomously learn the synthesis conditions for growing SWNTs at target growth rates. Using a random forest model and a genetic algorithm, ARES iteratively conducted experiments with series of experiments conducted in grouped tasks of 29–94 experiments. Convergence tests compared the experimental growth rates to the ARES-predicted growth rates where ARES was observed to converge after 500 experiments where larger yield of CNTs is observed with larger growth rates (Fig. 3A). Prior to commencing the experimental campaign, 84 experiments were selected in a grid throughout parameter space to build an initial predictive model. During the experimental campaign, the target growth rate was constant for initial experiments but later varied to promote coverage of the parameter space, showing an interesting interaction between the autonomous system and human experimenters. This approach led to 8% of model predictions matching the experimental values in the initial phases of the experimental campaign and 68% of model predictions matching experimental values at later stages in the campaign. While this approach was able to learn conditions for SWNT growth with over 500 experiments, the active learning approach used in ARES was updated to utilize BO [56] with the goal of further accelerating the pace of research.

In the ARES BO framework, the upper confidence bound (UCB) decision policy and a Matern 5/2 kernel belief model were utilized. Two experimental campaigns were conducted with the growth rate as the objective, treated as the square root of the growth rate in the implementation. One campaign was warm started with 25 experiments known to produce successful growth rates, while the second campaign was warm started with

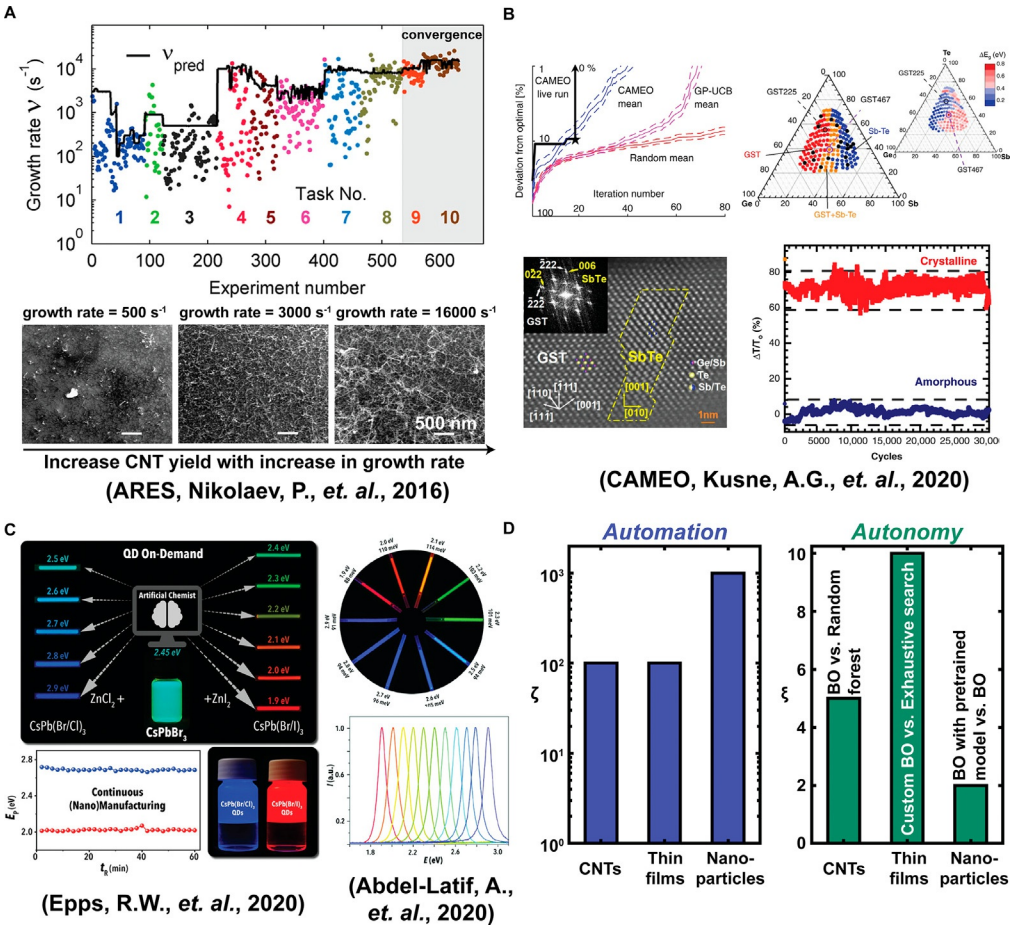


Fig. 3 Autonomous experimentation (AE) case studies in carbon nanotube growth, thin-film applications, and nanoparticle synthesis. (A) Experimental campaign of ARES with experimentally measured and predicted growth rates with converge after 500 experiments with SEM images depicting an increase in the yield of CNTs with an increase in growth rate [6]. (B) Experimental campaigns conducted using CAMEO to identify the material with the largest bandgap difference between amorphous and crystalline states [37]. (C) Artificial chemist enables quantum dot (QD) synthesis such as synthesis of perovskite QDs for 11 target emission colors along with a demonstration of continuously measured peak emission energy for 1 h at 2.7 eV *(blue)* and 2.0 eV *(red)* [55]. (D) Enhancement in throughput due to the use of automation ζ and enhancement in efficiency due to the use of custom active learning approaches ξ.

48 experiments selected uniformly at random throughput the parameter space. Both campaigns converged to optimum conditions in approximately 100 experiments which was a fivefold decrease relative to the previously used random forest-based approach. Moreover, the optimized growth rate found using BO was greater than growth rates found by the random forest planner by a factor of eight. A benefit of the BO approach

is the ability to use the belief model to model the parameter space. This was beneficial in contributing to the discovery of multiple optima in growth parameters that produce similar growth rates with similar diameter distributions of CNTs. These results demonstrate the efficacy of BO as an efficient sequential optimization algorithm for AE systems.

3.2 Optoelectronic properties of thin films

Properties of thin films are highly sensitive to defects resulting in significant deviation from bulk properties [57]. Designing and fabricating thin films with optimal properties poses a challenge due to the requirement for time and labor-intensive exploration of a vast parameter space of compositions, deposition parameters, and processing parameters which affect the structure and properties of the film [58]. With a limited number of accurate computational and theoretical models for predicting performance, experiment remains the primary way of studying thin organic films. A particular class of thin-film properties that have received significant attention are optoelectronic properties as a result of their increasing application in solar energy conversion, efficient solid-state lighting, and flexible electronics [59,60]. However, progress has been limited due to lack of scalable, high-quality, and reliable fabrication processes [61]. While new thin-film photovoltaic technologies are continuously emerging to address the issues of low performance [62], rapidly exploring the utility of these technologies remains a major bottleneck to significant progress in designing the next generation of thin films with optimum optoelectronic properties [63]. In the field of thin films, AE has emerged as a promising approach to traverse the complex parameter space of thin-film design and fabrication.

Prior to the utilization of AE, many efforts in the field of thin films were geared toward increasing experimental throughput and accelerating the experimental cycle of thin-film fabrication. By combining HTE and ML techniques, new photovoltaic materials were realized with over an order of magnitude reduction in experimental time relative to a manual process [64]. Specifically, over this 2-month period, a total of 96 precursor solutions were created which were synthesized using 28 solid precursors based on recently developed compounds. A total of 75 unique chemical compositions were deposited into thin films and 65 (87%) exhibited properties that were promising for optoelectronic applications. By implementing automated solutions, such as automated characterization, a 35-fold increase in throughput was observed relative to their manual laboratory baseline. The advancements described in this work provided evidence for the utility of automation and ML in accelerating the experimental cycle further motivating efforts toward AE.

The optimization of hole mobility of 2,2′,7,7′-tetrakis(*N*,*N*-di-*p*-methoxyphenyl-amine) 9,9′-spirobifluorene (spiroOMeTAD), an organic hole transport material common to perovskite solar cells, was demonstrated using an AE system. The AE systems, called "Ada," was capable of synthesizing, processing, characterizing organic thin films, and selecting subsequent experiments using BO [1]. "Ada" achieved this by automating

robotics at five key experimental steps: (1) measuring and mixing solutions; (2) depositing solutions as thin films; (3) annealing films; (4) imaging the films to detect morphologies, defects, and impurities; and (5) characterizing the optical and conductive properties of the films. From the optical and conductive properties obtained from automated experimentation, "Ada" selected subsequent experiments using ChemOS's Phoenics global BO algorithm [16,29]. Specifically, the hole mobility of the thin films was assessed using the pseudo-mobility ratio, which constitutes the quotient of the sheet conductance of a thin film and the absorptance of oxidized spiroOMeTAD in the film. The parameter space that was explored was two dimensional and consisted of the annealing time and the dopant concentration, where the dopant concentration was prepared from stock solutions of spiroOMeTAD and cobalt (III) dopant with a fixed amount of plasticizer 4-tert-butylpyridine. In the study, 2 AE campaigns were conducted each with an experimental budget of 35 experiments where each campaign was completed under 30 h. Both campaigns converged to the optimum doping ratio and annealing time of ~0.4 and ~75 s, respectively. Capitalizing on the benefit of the BO approach to utilize the belief model to explore the parameter space, a nontrivial local optimum with a high doping ratio and a high annealing time was also determined which represented an unexpected scientific observation.

AE systems have also been employed to study the optical properties of inorganic thin films. Specifically, an AE system employed to optimize the properties of thin films was the exploration of a Ge-Sb-Te ternary system to identify the optimal contrast in bandgap between amorphous and crystalline states for applications in photonically addressable phase change memory material (PCM) [37]. To achieve this, a closed-loop autonomous system for materials exploration and optimization (CAMEO) was developed which was able to autonomously control a high-throughput X-ray diffraction system at a synchrotron beamline, request specific input from a human in the loop, access prior experimental and theory-based knowledge of the target material system, and incorporate physics knowledge such as the Gibbs phase rule, and utilize BO to select subsequent experiments. CAMEO was able to identify the composition with the largest bandgap in 19 experiments (approximately 10 h) which was approximately a 10-fold reduction in the number of previously required experiments in the full set of 177 compositions (Fig. 3B). Additionally, CAMEO was compared to simulated campaigns of BO with a Gaussian process belief model and UCB decision policy (GP-UCB). Based on these simulated learning campaigns, CAMEO was able to identify the composition with the largest bandgap in 19 experiments, while a traditional BO formulation such as GP-UCB needed 54 experiments. Furthermore, the composition found by CAMEO outperformed a popular alternative composition in a photonic switch device by exhibiting an enhanced switching contrast resulting in 50% more interval states. From, the campaigns a new PCM was identified, and high-resolution transmission electron microscopy revealed a complex nanocomposite structure consisted of FCC-Ge-Sb-Te (GST) and Sb-Te phases where the dotted lines indicated the

atomically sharp interface (Fig. 3B). The success of CAMEO demonstrated the additional acceleration that AE systems can achieve when incorporating physics-based prior knowledge and human cooperation, which is a promising direction for the development and expansion of future AE systems.

3.3 Synthesis and optimization of nanoparticles

Nanoparticles possess unique size-related physio-chemical, optical, magnetic, and electronic properties; however, controlling the size, morphology, and properties of nanoparticles remains a challenge as numerous synthetic parameters and conditions must be considered [65,66]. Microfluidic reactors, such as tube-based flow reactors, have attained increasing popularity as a means for controlled synthesis of high-quality colloidal nanoparticles and have presented unprecedented opportunities for accelerating fundamental and applied studies of nanoparticles [67,68]. With the growing interest in flow reactor technology, versatile and reconfigurable microfluidic reactors have emerged enabling automated microfluidic reactors motivating the development and application of AE systems for nanoparticle research.

To tune the quantum yield and composition polydispersity of metal halide perovskite quantum dots (QDs) at target bandgaps through halide exchange reactions, the "Artificial Chemist" was designed and developed [41]. The Artificial Chemist comprised a precursor formulation module, a flow reactor module, an in-situ QD characterization module, and a BO active learning algorithm. By varying the conditions of the halide exchange reactions, the Artificial Chemist was able to simultaneously tune the photoluminescence quantum yield and the emission linewidth at target bandgaps. Notably, the Artificial Chemist enabled the study of >1400 reactions across 11 target peak emission energies and eight different reaction optimization algorithms (Fig. 3C). Among the eight different optimizations, three algorithms that were top performers were an ensemble neural network (NNE)-based BO approach with an EI decision policy, and NNE-based BO approach with an UCB decision policy, and a covariance matrix adaption evolution strategy (CMA-ES) approach. These 3 approaches were able to achieve their objective within 25 experiments with NNE-UCB being reported as the most consistent approach with the greatest performance. Additionally, it was reported that when the Artificial Chemist was pretrained, an NNE-based BO approach with an exploitative decision policy further reduced the number of experiments necessary by twofold. The development of the Artificial Chemist as a modular and versatile AE system for flow synthesis of nanoparticles has opened opportunities for reliable and high-throughput manufacturing of advanced and highly tuned nanoparticles.

A distinct AE system was used to efficiently explore the synthesis parameter space of lead halide perovskite (LHP) QDs and their corresponding peak emission energies [55]. The parameter space considered in this study included eight independent flow rate

parameters. Using the AE system, the synthesis parameters for 10 different target peak emissions were found in less than 40 min per target peak emission. Using an ENN-BO approach, the AE system first utilized 200 experiments to explore the parameter space using a maximum variance decision policy and then utilized five exploitative experiments for each target peak emission. Using this BO approach with the AE system, 10 optimized LHP QD production that corresponded to 10 target emission colors that ranged from blue to red were identified in the eight-dimensional parameter space (Fig. 3C). In another study with a smaller two-dimensional parameter space, a different team developed and utilized an AE system to explore the synthesis of cesium lead bromide QDs, a similar QD parameter space explored by the Artificial Chemist [41], but with a focus on optimizing the circular dichroism (CD) signal [69]. Driven by the customized materials acceleration operating system in the cloud (MAOSIC) enabled the optimization of the CD signal as a function of temperature and concentration in 250 experimental loops using stable noisy optimization by branch and fit (SNOBFIT)-based reinforcement learning.

3.4 Enhancement observed due to the introduction of automation and autonomy

The incorporation of AE in the previously discussed case studies demonstrated enhancement in experimental throughput due to the application of automated experimentation and enhancement in the research pace (Fig. 3D). In particular, experiment throughput enhancement ζ was reported to be a factor of 100 for ARES, an expected 100 times increase from a fully automated workflow for thin-film synthesis, and 1000 times increase in nanoparticle synthesis. While the substantial increase in experimental throughput due to automation experimentation is certainly desirable, AE systems have also demonstrated the ability to employ active learning approaches such as BO to further enhance the pace of research, quantified here as the enhancement factor ξ. Notably, ARES reported $\xi = 5$ where the random forest approach utilized 500 experiments while BO utilized 100 experiments. Additionally, CAMEO reported $\xi = 10$ where 19 experiments were needed to find an optimum material composition relative to the full set of 177 composition spots. Finally, the artificial chemist reported $\xi = 2$ using a pretrained BO approach that required 25 experiments while the other BO approaches required more than 50 experiments.

4. Platform technologies for AE in nanoscience

The case studies described above show how totally different platform technologies for AE are often needed to accommodate different experimental types in nanoscience. That need not be the case as robust and transferable experimental platforms such as MTP have demonstrated clear advantage in advancing a community holistically. Thus, a major current emphasis in nanoscience is identifying platform technologies that can enable rapid

exploration of many distinct systems with a single unified set of hardware. Motivated by the achievements of MTP in providing a standardized format for automated chemical processing, such a platform technology that is inherently built for nanotechnology research could take advantage of the small size of nanoscale systems to increase experimental throughput in a fashion akin to Moore's law. However, many challenges remain to realizing such a general system, often motivated by the unique complications intrinsic to nanoscale systems or the difficulties in integrated characterization hardware that is typically not utilized in an automated framework. This section details some recent efforts and ongoing work in this area.

4.1 Platforms for miniaturized liquid handling

Many nanoscience experiments are either performed in liquid or can be prepared in a liquid state. Thus, one popular approach to generating platform technologies is to look to the miniaturization of liquid handling. Automated liquid handling for nanotechnology research requires metering, mixing, and characterizing samples. While MTP-based HTE typically achieves this through a series of discrete large instruments such as liquid dispensing robots and photoluminescence plate readers, these require fluid volumes on the μL scale. Thus, one major emphasis has been to reduce the volumetric scale of these reactions. One such examples of automated liquid dispensing into a microwell chip at the nanoliter scale was presented by Wang et al. [70].

Inkjet printing is a technique that uses electrical, piezoelectric, or thermal effects to generate and sputter droplets on a surface in micro- and nanoscale using various inks. In traditional inkjet printing applications, results of the printing are observed afterward and there is no way determining or measuring the printed droplets during the patterning process. This limitation leads to not having control over adjusting printing results in-situ. To overcome this, Zhang et al. proposed an inkjet printing method that enables in-situ monitoring of size of the printed droplets by using a scalar diffraction vision system. Their system is able to measure the size of printed drops and adjust printing conditions [71]. In comparison to conventional inkjet printing, this method showcases real-time result monitoring, and it has the capability to automate all of the result inspection process. Similarly, it is of importance to understand the droplet jetting behavior in inkjet printing, such that consistent droplet deposition quality can be achieved. Huang et al.'s proposed method aims to use an unsupervised learning method by using the videos data as the study subjects to observe the droplet behavior in the inkjet printing process. This method was reported to be capable of learning latent representations of the droplet jetting process video data, which can be crucial to understand and predict the droplet behavior [72]. With this method's capabilities, this can be used for real-time monitoring of the inkjet printing process in the future, and enable an automated system where printing conditions can be optimized based on the droplet jetting behavior in-situ.

Rather than requiring discrete fluid drops prepared by printing, another approach is to use microfluidic devices with the capability to continuously flow and mix liquid samples. To produce discrete reaction vessels, these systems often produce emulsions with discrete drops. For example, Du et al.'s work DropLab is a fully automated microfluidic platform that controls droplet generation and screening and is capable of automated pL-scale liquid metering, sequential droplet formation, and sample introduction into the microfluidic device [73], as well as generation of droplet assays, and parallel screening of the droplet assays with low sample volumes. DropLab has been demonstrated for automated screening of enzyme inhibition assays and protein crystallization, as well as identification of reducible carbohydrates (Fig. 4A) [74].

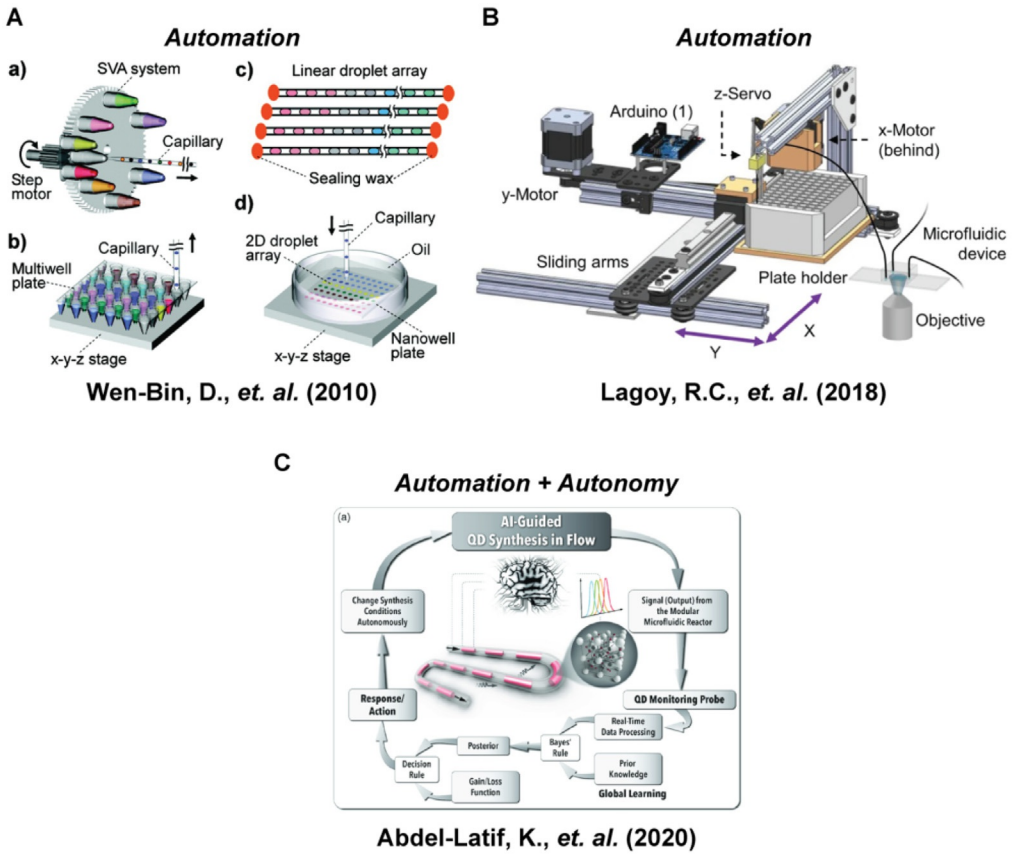


Fig. 4 Development of platforms for liquid handling. (A) DropLab system with depiction of the slotted-vial array, multiwell plate, wax seal for long-term storage, and deposition of droplets in the nanowell plates through a capillary [74]. (B) Robotic platform for fluid delivery comprised of a multiwell plate holder, microfluidic device, and microscope stage [75]. (C) Pipeline for AI-guided quantum dot (QD) synthesis in flow [66].

Interestingly, the concepts of microtiter plates and microfluidic systems have been merged to take advantage of their respective advantages. For example, Lagoy et al. reported the automated integration of microwell plates and microfluidic devices with a robotic system that automatically and reliably transfers fluid into any microfluidic system from a microwell plate through a single inlet tube. Their system can perform automated measurement of neural dose responses of chemical concentrations, multistep cell staining, and chemical screening (Fig. 4B) [75].

While autonomous fluid handling for microfluidic devices still have room to grow, one of the important applications in this area is autonomous material synthesis in flow using microfluidic devices. Similar to the advantages that microwell plates bring into sample preparation and interrogation, microfluidic reactors can highly reduce chemical consumption and waste generation, and can offer high-throughput and efficient chemical synthesis. The nature of microfluidic nanoreactors offers accelerated exploration of multivariable synthesis of target materials thanks to their superior heat and mass transfer. Combining the potential of microfluidic nanoreactors with AI-based experiment selection strategies make flow syntheses techniques an ideal approach for accelerated material's study. There have been various studies where microfluidic reactors have been utilized for autonomous exploration of colloidal nanomaterial syntheses, mainly quantum dots [55,76–78]. Latif et al.'s work shows AI-guided LHP quantum dot synthesis, which is fully autonomous and includes the generation of a large set of spectroscopic data in real time for experiment selection and data mining. The system they developed is capable of rapid formulation optimization of colloidal synthesis space without any user guidance, and their AI-based strategy integrated with microfluidic reactors provides a single platform capable of autonomous learning, optimization, and manufacturing of quantum dots (Fig. 4C) [55].

4.2 Scanning probes for patterning and interrogating nanoscale materials

As a platform technology for nanoscience, scanning probes provide unsurpassed abilities to observe, characterize, and manipulate materials at the atomic scale. Critically, as these single systems marry the capabilities to both image and manipulate, they can provide a path to closed-loop nanoscale experimentation in a single instrument. SPM techniques differ from each other based upon the nature of the probe in use and the type of information collected. For instance, scanning tunneling microscopy (STM) utilizes a sharp conducting tip as a probe to measure probe-sample current to image surfaces at the atomic level, while atomic force microscopy (AFM) uses the deflection of a microfabricated cantilever to measure the force between a sharp tip and the substrate, thus enabling topographic imaging, characterization, and manipulation of materials.

While most often used for characterization tasks, scanning probes are exceptional, and in some cases superlative tools, for fabricating nanoscale materials. For example, dip-pen

nanolithography (DPN) is a scanning probe-based lithography technique that uses the AFM cantilever tip as a "pen" to transfer and pattern various materials ranging from small molecules, biological materials, nanoparticles, and polymers in a direct-write fashion. Critically, it has been used to pattern liquids at the sub-aL scale [79]. While conventional DPN is open-loop and it therefore lacks precise control over the patterned amounts [80–82], Saygin et al.'s work is the first reported example of tracking and controlling the amount of liquid patterned with DPN in a closed-loop at the fL scale. While setting up experiments was not fully automated, the process for measuring the patterned amounts of liquid, predicting how much could be patterned based on cantilever-liquid state, and selecting patterning parameters for target amounts was all automated (Fig. 5A) [83]. Given the importance of preparing and manipulating liquid samples in small scales, this work lays the foundation toward fully automated liquid sample preparation with AFM.

While preparing samples with high precision, either by using DPN or other nanoscale processes, is necessary for nanoscale material discovery, the materials have to be characterized. SPM imaging techniques have exceptional control and high precision at the atomic scale; however, there are still experimental limitations that require improvement. Some of these limitations can be summarized as the quality of the probe tip and the need to correct tip quality during imaging, the long required time to take quality images, and the need for identifying small areas of interest prior to postexperiment analysis. Interestingly, many approaches have been proposed to address these problems and research groups have come up with their own techniques. While these challenges have not been fully addressed, the strategies employed include many facets of ML that are encouraging from the perspective of automation and incorporation into AE systems more generally.

One major challenge that must be addressed is that SPM does not only depend on the sample surface, but also the shape and state of the probe tip itself. Indeed, the probe is often the limiting factor in determining whether it is possible to get a good quality image with high resolution. This is especially true in STM where tip flaws must be repeatedly corrected to maintain good quality imaging. Identifying and correcting STM tip conditions have been mainly user-controlled tasks, where manual tip conditioning requires a pulse (or a crash), which followed by a scan to check if the tip can image with high quality. In order to optimize the time required not only to manually identify and correct STM tips, but also to minimize human error, several groups developed automated tip state detection and conditioning techniques. The software developed by Rashidi and Wolkow can detect STM tip states autonomously based on the image being taken via CNNs. When the software detects a double tip artifact, it can then act accordingly to correct the tip through automated indentation with the surface, which they explain as "autonomous tip sharpening" [86]. While Rashidi and Wolkow's method was only demonstrated while scanning the H:Si(100) surface, Gordon et al broadened their automated tip state detection and correction method while scanning on both metallic and

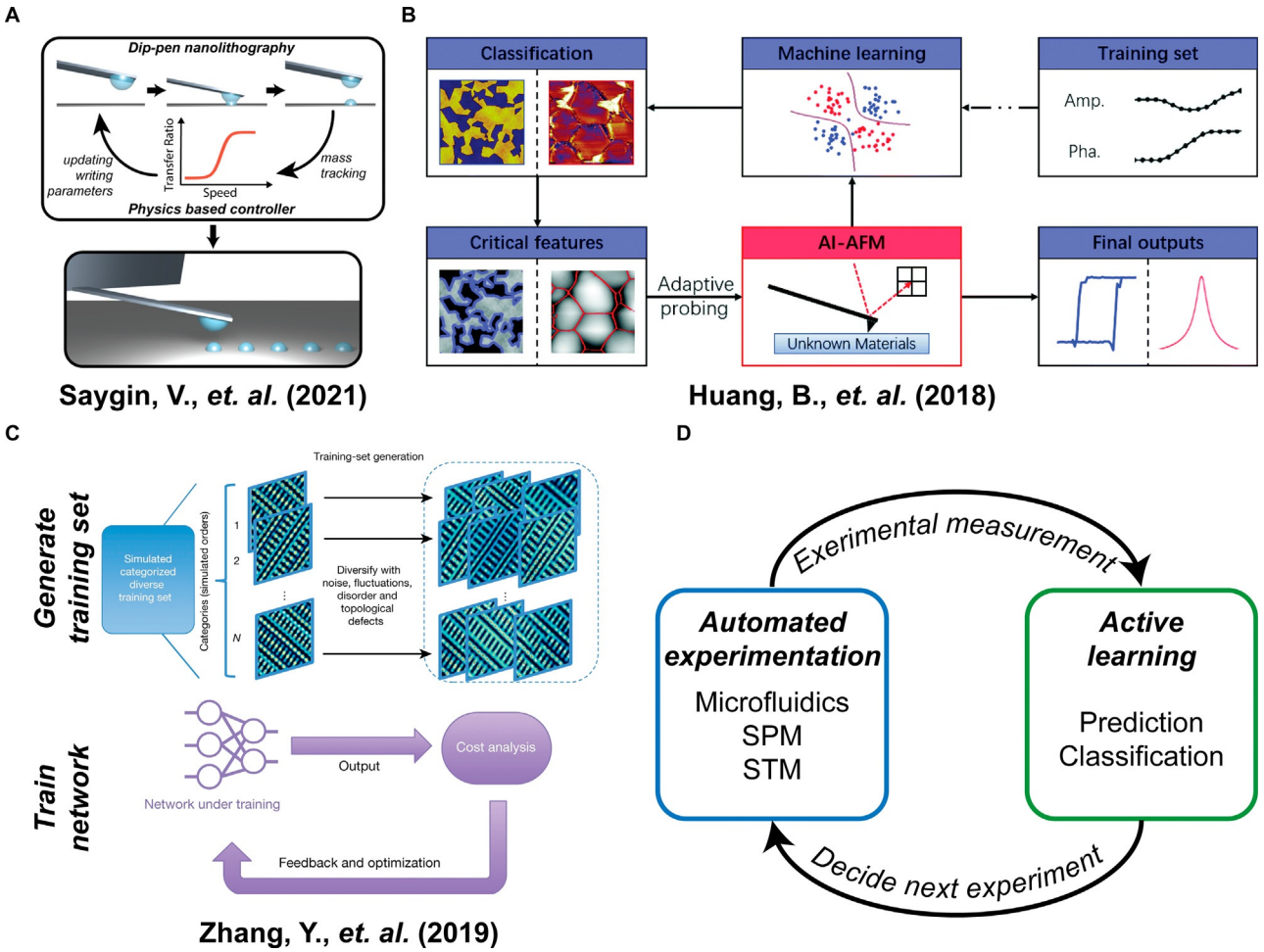


Fig. 5 Incorporation of machine learning approaches and scanning probes for patterning and interrogating nanoscale materials. (A) Development of a closed-loop dip-pen nanolithography approach [83]. (B) Workflow for AI-aided atomic force microscopy (AFM) for the classification of samples in real time [84]. (C) Pipeline for training an artificial neural network for analysis of electronic quantum matter image arrays obtained from scanning probe microscopy (SPM) [85]. (D) Concept schematic for the integration of automated scanning probe approaches and active learning.

semiconducting surfaces using CNNs [87]. In the following year, Gordon et al published a new method where they show tip state detection is achieved with partial scans by combining several methods as a hybrid approach. This hybrid method is reported to be at least an order of magnitude faster than the previously reported CNN protocols [88].

One of the biggest limitations in obtaining a high-resolution AFM image is the required time to complete a full scan. To address this problem, people have implemented autonomous processes for constructing images. Undersampling or subsampling is one of the complementary approaches for high-speed AFM, which is done by simply reducing the number of pixels in the scanned image. A complete image is then formed from the undersampled data by postprocessing [89]. Where the required time for a conventionally taken AFM image is measured in the order of minutes, undersampling reduces this time significantly. Luo and Andersson proposed an alternative way for reconstructing subsampled AFM image using a deep neural network approach, where usually the reconstruction methods for subsampled AFM images rely on optimization. They report that with deep neural networks (DNNs), reconstruction of the subsampled image is significantly faster than optimization methods, which is in the order of seconds in comparison to several minutes [90]. Wu et al proposed a path planning method for high-speed imaging with super resolution, where two different key automated applications are combined: generation and reconstruction of low-resolution image through CNNs and automated detection of area of interest for further probing. In the proposed method, the authors performed an initial fast scan to generate a low-resolution AFM image, which was then reconstructed into a higher resolution image using a CNN. The reconstructed "superresolution" AFM image was then further analyzed with an advanced object detection approach to determine and locate the objects of interest. Followed by this, the detected objects were scanned with low speed and high resolution, the results of which were then matched with the previously reconstructed AFM image to generate a single, superresolution image [91].

While reducing the time required to take a high-resolution AFM image is important, identifying and locating the area of interest on a sample is still an important challenge which relies on human input. It is important to be able to detect the objects or areas of interest prior to postanalysis of an image. Huang et al proposed an SVM-based AI algorithm that is capable of recognizing and identifying different areas of interest in real time while imaging with an AFM (Fig. 5B). Their AI-AFM can differentiate ferroelectric domains from nonferroelectric mappings by analyzing the AFM image pixel-by-pixel, without relying on an image being completed and can adjust the scanning parameters to initiate further probing at the identified areas of interest. Pixel-by-pixel recognition enables adaptive adjustment of scanning parameters without any user intervention [84]. Following Huang et al.'s AI-AFM, Krull et al proposed DeepSPM, which is explained as "artificial-intelligence-driven scanning probe microscopy." Their proposed method is capable of continuous SPM data acquisition, where all the imaging steps are

initiated and controlled by the AI system without any human intervention. Their proposed method combines all the previously presented methods: autonomous parameter selection and optimization, image analysis in real time, and adaptive probing and automatic detection and conditioning of the tip state. DeepSPM operates in a control loop, where it selects an appropriate scanning region and determined the scanning area and after completing a scan it evaluates both the imaging parameters and the selected area; this continues until the system completes a "good" image. If the classifier CNN thinks that the probe is bad, the system uses a deep RL agent to condition the tip. While the authors demonstrated this method with STM, they conclude that DeepSPM can be generalized to most SPM techniques with their open-source code [92].

There have also been advances in automated image analysis for SPM images. Postexperiment analysis of SPM images conventionally relied upon on researcher's manual interpretation, which includes selection of necessary filtering techniques, detection and quantitative analysis of subjects of interest. This process may require multiple software packages, is very time consuming, and open to human error and subjectivity when it comes to detection and identification of target measures. Automation of SPM imaging can provide fast and consistent postprocessing alternatives in comparison to conventional methods.

One of the first examples in this area was showed by Randall et al., where automated detection and identification of atomic terraces, dimer rows, vacancies, dangling rows, and surface contamination of STM images of Si (100) 2×1 H surfaces were performed by identifying all the dimer rows and deciding what each specific atom position corresponded to based on published knowledge about Si surfaces [93]. Open-source toolbox development is another approach toward automated data analysis of SPM images. Two such examples can be seen in Stirling et al. and Gudinas et al.'s works; they developed open-source toolboxes for automated SPM data analysis, which can be applied to a wide range of SPM datasets and can perform adaptive masking and flatting, step edge detection, atom and molecule recognition and generate image statistics [94,95]. Gudinas et al. state that the toolbox they developed is limited by the quality of the STM images, and high background noise can result in poor results for isolating defects. One of the most impressive examples of automated data analysis of SPM images was proposed by two different groups. Zhang et al. showed that they were able to extract buried information from experimental SPM data by using an artificial neural network (Fig. 5C) [85], which was not possible for humans to detect, and similarly, Alldritt et al. showed that buried information from experimental AFM data can be extracted reliably and rapidly using CNNs [96].

The combination of automated platform technologies for nanoscience with active learning algorithms presents innumerable opportunities for the discovery and development of new materials (Fig. 5D). While prior work has developed and utilized closed-loop systems involving microfluidics, SPM, and STM, ample opportunity exists

for the application of AE systems for regression, classification, and discovery applications in these areas. The increasing availability of open-source software packages for active learning lowers the barrier to entry for combining nanoscience platform technologies and active learning algorithms to select subsequent experiments using available experimental data.

5. Conclusions and future directions

Nanoscale systems have many unique features, but their unifying characteristic is that they are small. This simple feature can translate to a major advantage when considering experimentation in that each system under study occupies a small volume, providing a potential path to extraordinary rate of experimentation. Unfortunately, the nanoscale dimensions of such systems are often also a challenge as the analytical systems needed to interrogate nanoscale systems are often quite large. Thus, the path to realizing general purpose AE systems for studying nanoscale materials must find a balance between leveraging scalability while maintaining a consistent and clear window into the nanoscale world. The examples of AE systems that have been realized to date show the promise of this new experimental paradigm in accelerating research, but challenges remain. The overwhelming combinations of materials, processing conditions, and properties mean that human intuition and oversight will remain a critical factor, if only to select the scope and objective of AE campaigns. Further, advances from other communities on how best to leverage AE systems to employ simulation and other types of prior knowledge will be indispensable in realizing the potential of AE systems. Taken together, we show how AE as a physical embodiment as the merging of data science and physical experimentation can help advance the field of nanoscience.

References

[1] B.P. MacLeod, et al., Self-driving laboratory for accelerated discovery of thin-film materials, Sci. Adv. 6 (2020), https://doi.org/10.1126/sciadv.aaz8867.
[2] M.M. Flores-Leonar, et al., Materials acceleration platforms: on the way to autonomous experimentation, Curr. Opin. Green Sustain. Chem. 25 (2020), 100370.
[3] R.D. King, et al., Functional genomic hypothesis generation and experimentation by a robot scientist, Nature 427 (2004) 247–252.
[4] R.D. King, et al., The automation of science, Science 324 (2009) 85–89.
[5] K. Williams, et al., Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases, J. R. Soc. Interface 12 (2015), 20141289.
[6] P. Nikolaev, et al., Autonomy in materials research: a case study in carbon nanotube growth, npj Comput. Mater. 2 (2016), https://doi.org/10.1038/npjcompumats.2016.31.
[7] A. Sparkes, et al., Towards robot scientists for autonomous scientific discovery, Autom. Exp. 2 (2010) 1–11.
[8] S.N. Deming, H.L. Pardue, An automated instrumental system for fundamental characterization of chemical reactions, Anal. Chem. 43 (1971) 192–200.
[9] L.M. Mayr, D. Bojanic, Novel trends in high-throughput screening, Curr. Opin. Pharmacol. 9 (2009) 580–588.

[10] D. Perera, et al., A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow, Science 359 (2018) 429–434.
[11] C.W. Coley, N.S. Eyke, K.F. Jensen, Autonomous discovery in the chemical sciences part I: progress, Angew. Chem. Int. Ed. 59 (2019) 2–38.
[12] M. Legrand, A. Foucard, Automation on the laboratory bench, J. Chem. Educ. 55 (1978) 767–771.
[13] A. Bédard, et al., Reconfigurable system for automated optimization of diverse chemical reactions, Science 361 (2018) 1220–1225.
[14] D. Fan, et al., A robotic intelligent towing tank for learning complex fluid-structure dynamics, Sci. Robot. 4 (2019) 1–13.
[15] B. Burger, et al., A mobile robotic chemist, Nature 583 (2020) 237–241.
[16] L.M. Roch, et al., ChemOS: an orchestration software to democratize autonomous discovery, PLoS One 15 (2020) 1–18.
[17] A.E. Gongora, et al., Using simulation to accelerate autonomous experimentation (AE): a case study using mechanics, SSRN Electron. J. (2021), https://doi.org/10.2139/ssrn.3751791.
[18] A. Jain, et al., Commentary: the materials project: a materials genome approach to accelerating materials innovation, APL Mater. 1 (2013), https://doi.org/10.1063/1.4812323.
[19] B.G. Buchanan, et al., Applications of artificial intelligence for chemical inference. 22. Automatic rule formation in mass spectrometry by means of the Meta-DENDRAL program, J. Am. Chem. Soc. 98 (1976) 6168–6178.
[20] P. Langley, G.L. Bradshaw, H.A. Simon, Bac0n5: the discovery of conservation laws, IJCAI (U S) (1981) 121–126.
[21] B.C. Falkenhainer, R.S. Michalski, Integrating quantitative and qualitative discovery: the ABACUS system, Mach. Learn. 1 (1986) 367–401.
[22] D. Kulkarni, H.A. Simon, The processes of scientific discovery: the strategy of experimentation, Cogn. Sci. 12 (1988) 139–175.
[23] J.M. Zytkow, J. Zhu, R. Zembowicz, Operational definition refinement: a discovery process, in: Proceedings of the Tenth National Conference on Artificial Intelligence, 1992, pp. 76–81.
[24] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (2009) 81–86.
[25] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, E.T. Mueller, Watson: beyond jeopardy! Artif. Intell. 199–200 (2013) 93–105.
[26] H. Kitano, Artificial intelligence to win the Nobel prize and beyond: creating the engine for scientific discovery, AI Mag. 37 (2016) 39–49.
[27] D.A. Cohn, Z. Ghahramani, M.I. Jordan, Active learning with statistical models, J. Artif. Intell. Res. 887 (1996) 705–712.
[28] P.I. Frazier, J. Wang, Bayesian optimization for materials design, Springer Ser. Mater. Sci. 225 (2015) 45–75.
[29] F. Häse, L.M. Roch, C. Kreisbeck, A. Aspuru-Guzik, Phoenics: a Bayesian optimizer for chemistry, ACS Cent. Sci. 4 (2018) 1134–1145.
[30] C. Qin, D. Klabjan, D. Russo, Improving the expected improvement algorithm, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1–11.
[31] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. De Freitas, Taking the human out of the loop: a review of Bayesian optimization, Proc. IEEE 104 (2016) 148–175.
[32] L.M. Roch, et al., ChemOS: orchestrating autonomous experimentation, Sci. Robot. 3 (2018) 19.
[33] V. Duros, et al., Human versus robots in the discovery and crystallization of gigantic polyoxometalates, Angew. Chem. Int. Ed. 56 (2017) 10815–10820.
[34] A.E. Gongora, et al., A Bayesian experimental autonomous researcher for mechanical design, Sci. Adv. 6 (2020), https://doi.org/10.1126/sciadv.aaz1708.
[35] N.M.F.S.A. Cerqueira, et al., Receptor-based virtual screening protocol for drug discovery, Arch. Biochem. Biophys. 582 (2015) 56–67.
[36] R. Gómez-Bombarelli, et al., Automatic chemical design using a data-driven continuous representation of molecules, ACS Cent. Sci. 4 (2018) 268–276.
[37] A.G. Kusne, et al., On-the-fly closed-loop materials discovery via Bayesian active learning, Nat. Commun. 11 (2020) 1–11.

[38] J.M. Granda, L. Donina, V. Dragone, D.L. Long, L. Cronin, Controlling an organic synthesis robot with machine learning to search for new reactivity, Nature 559 (2018) 377–381.
[39] M.M. Noack, et al., A kriging-based approach to autonomous experimentation with applications to X-ray scattering, Sci. Rep. 9 (2019) 1–19.
[40] L. Porwol, et al., An autonomous chemical robot discovers the rules of inorganic coordination chemistry without prior knowledge, Angew. Chem. Int. Ed. 59 (2020) 11256–11261.
[41] R.W. Epps, et al., Artificial chemist: an autonomous quantum dot synthesis bot, Adv. Mater. 32 (2020), https://doi.org/10.1002/adma.202001626.
[42] A.G. Mamalis, L.O.G. Vogtländer, A. Markopoulos, Nanotechnology and nanostructured materials: trends in carbon nanotubes, Precis. Eng. 28 (2004) 16–30.
[43] S. Polizu, O. Savadogo, P. Poulin, L. Yahia, Applications of carbon nanotubes-based biomaterials in biomedical nanotechnology, J. Nanosci. Nanotechnol. 6 (2006) 1883–1904.
[44] R.H. Baughman, A.A. Zakhidov, W.A. De Heer, Carbon nanotubes: the route toward applications, Science 297 (2002) 787–792.
[45] S. Kumar, R. Rani, N. Dilbaghi, K. Tankeshwar, K.H. Kim, Carbon nanotubes: a novel material for multifaceted applications in human healthcare, Chem. Soc. Rev. 46 (2017) 158–196.
[46] I. Sumio, Helical microtubules of graphitic carbon, Nature 354 (1991) 56–58.
[47] D. Tasis, N. Tagmatarchis, A. Bianco, M. Prato, Chemistry of carbon nanotubes, Chem. Rev. 106 (2006) 1105–1136.
[48] N. Anzar, R. Hasan, M. Tyagi, N. Yadav, J. Narang, Carbon nanotube – a review on synthesis, properties and plethora of applications in the field of biomedical science, Sensors Int. 1 (2020), 100003.
[49] S.S.M. Saleh, H.M. Akil, M.H. Abdul Kudus, Fillers and Reinforcements for Advanced Nanocomposites, Elsevier, 2015, pp. 82–83.
[50] R. Rao, D. Liptak, T. Cherukuri, B.I. Yakobson, B. Maruyama, In situ evidence for chirality-dependent growth rates of individual carbon nanotubes, Nat. Mater. 11 (2012) 213–216.
[51] R. Rao, et al., Revealing the impact of catalyst phase transition on carbon nanotube growth by in situ raman spectroscopy, ACS Nano 7 (2013) 1100–1107.
[52] C.R. Oliver, et al., Statistical analysis of variation in laboratory growth of carbon nanotube forests and recommendations for improved consistency, ACS Nano 7 (2013) 3565–3580.
[53] C.R. Oliver, et al., Robofurnace: a semi-automated laboratory chemical vapor deposition system for high-throughput nanomaterial synthesis and process discovery, Rev. Sci. Instrum. 84 (2013) 1–14.
[54] P. Nikolaev, D. Hooper, N. Perea-López, M. Terrones, B. Maruyama, Discovery of wall-selective carbon nanotube growth conditions via automated experimentation, ACS Nano 8 (2014) 10214–10222.
[55] K. Abdel-Latif, et al., Self-driven multistep quantum dot synthesis enabled by autonomous robotic experimentation in flow, Adv. Intell. Syst. 3 (2020), 2000245.
[56] J. Chang, et al., Efficient closed-loop maximization of carbon nanotube growth rate using Bayesian optimization, Sci. Rep. 10 (2020) 1–9.
[57] N. Kaiser, Review of the fundamentals of thin-film growth, Appl. Opt. 41 (2002) 3053.
[58] S. Biswas, O. Shalev, M. Shtein, Thin-film growth and patterning techniques for small molecular organic compounds used in optoelectronic device applications, Annu. Rev. Chem. Biomol. Eng. 4 (2013) 289–317.
[59] O. Ostroverkhova, Organic optoelectronic materials: mechanisms and applications, Chem. Rev. 116 (2016) 13279–13412.
[60] S.R. Forrest, M.E. Thompson, Introduction: organic electronics and optoelectronics, Chem. Rev. 107 (2007) 923–925.
[61] M. Leskelä, M. Mattinen, M. Ritala, Review article: atomic layer deposition of optoelectronic materials, J. Vac. Sci. Technol. B 37 (2019), 030801.
[62] L. Meng, et al., Organic and solution-processed tandem solar cells with 17.3% efficiency, Science 361 (2018) 1094–1098.
[63] T.D. Lee, A.U. Ebong, A review of thin film solar cell technologies and challenges, Renew. Sust. Energ. Rev. 70 (2017) 1286–1297.
[64] S. Sun, et al., Accelerated development of perovskite-inspired materials via high-throughput synthesis and machine-learning diagnosis, Joule 3 (2019) 1437–1451.

[65] A.A. Volk, R.W. Epps, M. Abolhasani, Accelerated development of colloidal nanomaterials enabled by modular microfluidic reactors: toward autonomous robotic experimentation, Adv. Mater. (2020), https://doi.org/10.1002/adma.202004495.
[66] K. Abdel-Latif, F. Bateni, S. Crouse, M. Abolhasani, Flow synthesis of metal halide perovskite quantum dots: from rapid parameter space mapping to AI-guided modular manufacturing, Matter 3 (2020) 1053–1086.
[67] V. Kumar, et al., Continuous flow synthesis of anisotropic cadmium selenide and zinc selenide nanoparticles, ChemNanoMat 3 (2017) 204–211.
[68] S. Kubendhiran, Z. Bao, K. Dave, R.S. Liu, Microfluidic synthesis of semiconducting colloidal quantum dots and their applications, ACS Appl. Nano Mater. 2 (2019) 1773–1790.
[69] J. Li, et al., Autonomous discovery of optically active chiral inorganic perovskite nanocrystals through an intelligent cloud lab, Nat. Commun. 11 (2020) 1–10.
[70] J. Wang, et al., A chip-to-chip nanoliter microfluidic dispenser, Lab Chip 9 (2009) 1831–1835.
[71] X. Zhang, B. Lies, H. Lyu, H. Qin, In-situ monitoring of electrohydrodynamic inkjet printing via scalar diffraction for printed droplets, J. Manuf. Syst. 53 (2019) 1–10.
[72] J. Huang, et al., Unsupervised learning for the droplet evolution prediction and process dynamics understanding in inkjet printing, Addit. Manuf. 35 (2020), 101197.
[73] W. Bin Du, Q. Fang, Z.L. Fang, Microfluidic sequential injection analysis in a short capillary, Anal. Chem. 78 (2006) 6404–6410.
[74] W. Bin Du, M. Sun, S.Q. Gu, Y. Zhu, Q. Fang, Automated microfluidic screening assay platform based on DropLab, Anal. Chem. 82 (2010) 9941–9947.
[75] R.C. Lagoy, D.R. Albrecht, Automated fluid delivery from multiwell plates to microfluidic devices for high-throughput experiments and microscopy, Sci. Rep. 8 (2018) 1–10.
[76] S. Krishnadasan, R.J.C. Brown, A.J. DeMello, J.C. DeMello, Intelligent routes to the controlled synthesis of nanoparticles, Lab Chip 7 (2007) 1434–1441.
[77] R.M. Maceiczyk, A.J. Demello, Fast and reliable metamodeling of complex reaction spaces using universal kriging, J. Phys. Chem. C 118 (2014) 20026–20033.
[78] L. Bezinge, R.M. Maceiczyk, I. Lignos, M.V. Kovalenko, A.J. Demello, Pick a color MARIA: adaptive sampling enables the rapid identification of complex perovskite nanocrystal compositions with defined emission characteristics, ACS Appl. Mater. Interfaces 10 (2018) 18869–18878.
[79] D.J. Eichelsdoerfer, K.A. Brown, C.A. Mirkin, Capillary bridge rupture in dip-pen nanolithography, Soft Matter 10 (2014) 5603–5608.
[80] H. Nakashima, M.J. Higgins, C. O'Connell, K. Torimitsu, G.G. Wallace, Liquid deposition patterning of conducting polymer ink onto hard and soft flexible substrates via dip-pen nanolithography, Langmuir 28 (2012) 804–811.
[81] C.D. O'Connell, M.J. Higgins, R.P. Sullivan, S.E. Moulton, G.G. Wallace, Ink-on-probe hydrodynamics in atomic force microscope deposition of liquid inks, Small 10 (2014) 3717–3728.
[82] C.D. O'Connell, M.J. Higgins, D. Marusic, S.E. Moulton, G.G. Wallace, Liquid ink deposition from an atomic force microscope tip: deposition monitoring and control of feature size, Langmuir 30 (2014) 2712–2721.
[83] V. Saygin, B. Xu, S.B. Andersson, K.A. Brown, Closed-loop nanopatterning of liquids with dip-pen nanolithography, ACS Appl. Mater. Interfaces (2021), https://doi.org/10.1021/acsami.1c00095.
[84] B. Huang, Z. Li, J. Li, An artificial intelligence atomic force microscope enabled by machine learning, Nanoscale 10 (2018) 21320–21326.
[85] Y. Zhang, et al., Machine learning in electronic-quantum-matter imaging experiments, Nature 570 (2019) 484–490.
[86] M. Rashidi, R.A. Wolkow, Autonomous scanning probe microscopy in situ tip conditioning through machine learning, ACS Nano 12 (2018) 5185–5189.
[87] O. Gordon, et al., Scanning tunneling state recognition with multi-class neural network ensembles, Rev. Sci. Instrum. 90 (2019), https://doi.org/10.1063/1.5099590.
[88] O.M. Gordon, F.L.Q. Junqueira, P.J. Moriarty, Embedding Human Heuristics in Machine-Learning-Enabled Probe Microscopy, arXiv, 2019, https://doi.org/10.1088/2632-2153/ab42ec.
[89] S.B. Andersson, L.Y. Pao, Non-raster sampling in atomic force microscopy: a compressed sensing approach, in: Proceedings of the American Control Conference, 2012, pp. 2485–2490.

[90] Y. Luo, S.B. Andersson, Image reconstruction for sub-sampled atomic force microscopy images using deep neural networks, Micron 130 (2020), 102814.
[91] Y. Wu, Y. Fang, C. Wang, C. Liu, Z. Fan, A high-speed atomic force microscopy with super resolution based on path planning scanning, Ultramicroscopy 213 (2020), 112991.
[92] A. Krull, P. Hirsch, C. Rother, A. Schiffrin, C. Krull, Artificial-intelligence-driven scanning probe microscopy, Commun. Phys. 3 (2020) 1–8.
[93] J.N. Randall, J.R. Von Ehr, J.B. Ballard, J.H.G. Owen, E. Fuchs, Automated scanning tunneling microscope image analysis of Si (100):H 2 × 1 surfaces, Microelectron. Eng. 98 (2012) 214–217.
[94] J. Stirling, R.A.J. Woolley, P. Moriarty, Scanning probe image wizard: a toolbox for automated scanning probe microscopy data analysis, Rev. Sci. Instrum. 84 (2013), https://doi.org/10.1063/1.4827076.
[95] A. Gudinas, J. Moscatello, S.M. Hollen, Defect identification and statistics toolbox: automated defect analysis for scanning probe microscopy images, J. Phys. Condens. Matter 33 (2020), https://doi.org/10.1088/1361-648X/abc1b2.
[96] B. Alldritt, et al., Automated Structure Discovery in Atomic Force Microscopy, arXiv, 2019, pp. 1–10.

CHAPTER 13

Nanomaterials and artificial intelligence in anti-counterfeiting

Yongfeng Lu[a,b], Hongrui Cheng[a,b], Paul S. Francis[c], and Yuanhui Zheng[a,b]
[a]College of Chemistry, Fuzhou University, Fuzhou, China
[b]Fujian Science & Technology Innovation Laboratory for Optoelectronic Information, China
[c]School of Life and Environmental Sciences, Faculty of Science, Engineering and Built Environment, Deakin University, Waurn Ponds, VIC, Australia

1. Introduction

Counterfeiting is growing worldwide and has become a serious global issue [1–6]. Over the last few decades, counterfeited and pirated goods have pervaded all levels of our modern society. They are commonly found in medicine, food, cosmetics, precious art, and almost every aspect of our daily life [1,3–6]. For example, a report by the World Health Organization showed that approximately 10% of medicines around the world may be counterfeit, and this number could be even higher in developing countries [7]. Counterfeit goods not only pose a risk to human health but also cause significant damage to the global economy. It has been reported that the global economic losses caused by fake products reached 1.82 trillion USD in 2020 [8]. Anti-counterfeiting technologies are implemented by authorities and/or manufacturers to prevent the production and/or sale of counterfeit goods (including banknotes, identity cards/documents, and commercial products). The global market size for these technologies was 51.8 billion USD in 2017, and is predicted to grow annually at a rate of 11.7% from 2018 to 2025 [9]. There is therefore a prevailing need for multidisciplinary research to devise new secure anti-counterfeiting strategies to "outsmart" counterfeiters.

Typical anti-counterfeiting systems comprise two components: patterned security tags and reliable authentication techniques. The key ingredient of a security label is the information carrier within the pattern. It is usually a nanomaterial that exhibits at least one unique physical response under external stimuli (e.g., light, heat, magnetic field, etc.). The physical responses are encoded in the patterns of the security labels that can be directly visualized by eye or read by an authentication tool such as an optical microscopes or spectrometers. The field of anti-counterfeiting has seen three major trends over the past few decades: (1) new stimuli-responsive nanomaterials are being investigated as information carriers to increase the encoding levels (see Section 3); (2) new patterning techniques are being created to enhance the patterning complexity while decreasing the fabrication cost (see Sections 3 and 4); and (3) miniaturized authentication tools

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00013-5
Copyright © 2023 Elsevier Inc.
All rights reserved.

integrated with smartphones and advanced algorithms are being developed (see Sections 5 and 6). Fundamental advances in nanotechnology and artificial intelligence are generally seen as key to the development of the next-generation of anti-counterfeiting technologies. This draws attention to the synthesis of novel functional nanomaterials with unique properties and the development of techniques in pattern extraction, recognition, and comparison.

2. Encryption mechanism of optical security labels

The commercialized anti-counterfeiting technologies mainly include radio frequency identification tags (RFIDs) and optical anti-counterfeiting tags (e.g., holographic tags, barcodes, and QR codes), which allow frequent authentication at any point of dispatch and storage from the manufacturer to the end user [10,11]. Combining intelligent light management through nanomaterials with novel printing techniques and smartphones as verification tools has the scope to create advanced optical anti-counterfeiting technologies that are less expensive, simpler, and more reliable than RFIDs.

The encryption mechanism of optical security labels relies on the optical responses of the nanomaterial under external light stimuli. The irradiation of an optical nanomaterial results in the absorption, reflection, refraction (including transmission), diffraction, and scattering of the incident light, which can give the material a specific color (i.e., structural color) and can be used as coding information in anti-counterfeiting (Fig. 1). The absorption of the incident light may also lead to the formation of excitons (i.e., electron–hole pairs) or surface plasmons (i.e., the collective oscillation of free-electron clouds in resonance with the incident electromagnetic field) depending on the types of optical nanomaterials involved [9]. The generated excitons can recombine and emit light with wavelengths different to those of the incident light. Emission that occurs immediately upon excitation is called fluorescence, whereas emission after the excitation source is removed (referred to as afterglow) can involve thermally activated delayed fluorescence

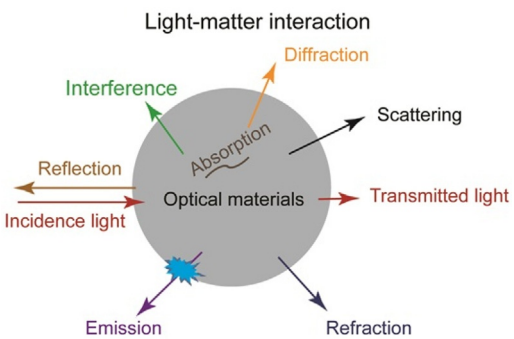


Fig. 1 Optical responses of a material interacting with light.

(TADF) or phosphorescence. Surface plasmons that are strongly confined within the metal nanoparticles or highly doped semiconductor nanoparticles can effectively absorb and scatter light, not only providing the material color but also generating high electromagnetic field intensities in the close vicinity of the surface of the nanoparticles [12–14]. The strong near-field enhancement effect can be exploited for the amplification of fluorescence and Raman signals of molecules surrounding the nanoparticles, offering additional optical responses. The multiple optical responses including scattered light, surface plasmon-enhanced fluorescence (SPEF), and surface-enhanced Raman scattering (SERS), can be assigned as multiple codes for anti-counterfeiting applications [12,13,15].

So far, the optical information in the majority of the investigated security labels is encoded in two-dimensional (2D) patterns in "collective behaviors." That is, the coding signals of two neighboring information carriers within the security labels cannot be spatially distinguished. These security labels fall into the category of traditional optical security labels produced by a deterministic process. The low complexity and high predictability of these security labels mean that they could be readily copied by counterfeiters [16]. To address this issue, physical unclonable functions (PUFs, also known as physical one-way functions) have been proposed and introduced in optical encryption and decryption, and are the foundation for unclonable anti-counterfeiting systems (referred to as PUF security labels) [17]. Recent advances in nanotechnology give researchers opportunities to encode optical information in random physical patterns with spatially distinguishable nanoscopic features, forming PUF security labels. These labels, on the macroscopic level, show features that look similar to traditional optical security labels, while on the nanoscopic level, they are completely different from each other due to the random nanoscopic features within the macroscopic pattern. These nanoscopic features are batch-to-batch dependent and prohibitively difficult to duplicate [18].

3. Advanced optical nanomaterials for anti-counterfeiting applications

Driven by potential applications in optical encryption, imaging, and displays, there has been tremendous interest in optical nanomaterials owing to their unique and fascinating properties that are superior to those of bulk materials [9,10]. Much of this interest is powered by the growing expertise in wet-chemical synthesis routes that enable the production of nanomaterials with well-controlled size, morphology, and composition [10,19] and advanced micro/nanofabrication techniques that are able to generate 2D patterns [20,21]. There are three main types of materials exploited for their unique optical properties in multiple dimensions (e.g., responsive wavelength (color), reflection or emission intensity, and/or emission lifetime): photonic crystals, luminescent materials, and plasmonic materials. Such nanomaterials carry rich optical information and are widely applied in anti-counterfeiting applications [9].

3.1 Photonic crystals

Photonic crystals (also known as structural color materials) are periodic dielectric structures that can manipulate the flow of light and have been successfully used in commercial anti-counterfeiting applications as hologram labels. Their colors mainly arise from interference, diffraction, scattering of incident light caused by the special periodic structures of the materials [22,23]. Numerous fabrication methods have been developed to achieve photonic structures at different length scales and can be categorized into self-assembly approaches, top-down routes, and their combinations [22]. Self-assembly methods take advantage of intraparticular interactions (e.g., capillary force and electrostatic interactions) [24–27] of colloidal building blocks or external forces (e.g., electric field and magnetic field) [27–29] to arrange them into ordered structures. With the help of spin coating, a wafer-scale monolayer crystal composed of non-close-packed spherical building blocks has been produced through colloidal self-assembly [30]. Top-down approaches, on the other hand, make use of advanced microfabrication techniques (e.g., direct laser writing [31], holographic lithography [32], nanoimprint lithography [33,34], etching [35], and so on) to generate periodic structures with desired separation and ordering from bulk materials. For example, Campbell et al. demonstrated the fabrication of photonic crystals with sub-micrometer periodicity by a holographic lithography that involves the exposure of a photoresist film to a three-dimensional (3D) interference pattern generated by four noncoplanar laser beams and the dissolution of unexposed photoresist [22]. Compared with self-assembly methods, the top-down approaches have illimitable pattern design capability but suffer from high fabrication cost. As a practical fabrication strategy for security labels with desired patterns, self-assembly can serve as complementary processing steps within a sequence of the top-down fabrication techniques. To this end, templated self-assembly and novel printing techniques have been developed for the fabrication of patterned photonic crystals [23,36].

The optical response (λ) of photonic crystals comprising spherical particles is determined by their structures and can be expressed by the following equation:

$$m\lambda = \sqrt{\frac{8}{3}}D\left(\sum_i n_i^2 V_i - \sin^2\phi\right)^{1/2} \quad (1)$$

where D is the lattice constant (i.e., center-to-center distance between two nearest spheres), n_i and V_i are the refractive index and volume fraction of each component, and ϕ is the angle between the incident light and the sample normal [26]. The lattice constant of a photonic crystal can be adjusted by controlling the size of the building blocks [21], filling the voids with a chemical that is able to swell and shrink [23], or mechanical pressing or stretching [37], which allow researchers to manipulate their optical properties in the visible region. According to Eq. 1, the change of the compositions (i.e., refractive index, n_i) of the building blocks and the filling materials in the voids also

allows for a shift in their spectra. When the building blocks are magnetic particles, the lattice constant and ordering of the photonic crystals can be further tuned by the external magnetic field [25]. It has been demonstrated by the group of Yin that magnetic Fe_3O_4 colloids are able to assemble into photonic crystals that show vibrant colors from red to blue when increasing the strength of the applied magnetic field [25].

3.2 Luminescent materials

Luminescent materials (also known as phosphors) are substances including organic and inorganic compounds that emit light under excitation. Although a large number of luminescent materials have been discovered, here we will mainly focus on perovskite quantum dots (QDs), rare-earth fluoride nanocrystals, and carbon dots, as they have received much attention as information carriers for anti-counterfeiting applications.

Liquid phase synthesis is the most popular approach to prepare inorganic QDs due to its simplicity and potential for mass production. The hot-injection method, involving injection of a metal–organic precursor into a high-boiling solvent containing ligands and the subsequent decomposition or reaction with other precursors in the solvent at an elevated temperature, is widely used for the synthesis of nanocrystals ranging from II-VI semiconductors to rare-earth fluorides and halide perovskites [38–43]. Their size, shape, composition, and heterostructures (especially core–shell structures) can be easily engineered by controlling the reaction time and temperature as well as the ratios of the precursors and ligands [39,44]. Monodispersed rare-earth fluoride and halide perovskite nanocrystals with high crystallinity and desired morphology have been achieved through hot-injection method. However, this method also suffers from two major drawbacks. Firstly, it requires harsh reaction conditions, such as a high temperature and oxygen/water-free environment, to obtain high-quality nanocrystals. Secondly, the produced nanocrystals show poor water solubility because of the presence of hydrophobic ligands on their surface.

The reprecipitation method, which takes advantage of the solubility variations of precursors when mixing a polar solvent containing highly concentrated precursors with a poor solvent to reprecipitate the desired product, is another facile method for synthesizing colloidal perovskite QDs [45]. The synthesis is initiated by the dropwise addition of the concentrated precursor solution into a poor solvent to induce reprecipitation of perovskite QDs at room temperature, due to the supersaturation of the precursors in the mixed solvents [46]. The presence of ligands in the poor solvent can narrow the size distribution of QDs and increase their resistance to moisture in air [46,47], which are important for the fabrication of bright and stable fluorescent security labels. Similar to the reprecipitation method, chemical co-precipitation is a widely adopted method for the synthesis of rare-earth fluoride nanocrystals, which makes use of the precipitation of target products out of the precursor solution at a relatively low temperature. Carbon

dots, a metal-free luminescent material, were discovered as a by-product of single-walled carbon nanotubes synthesized by arc discharge [48]. The wet-chemical synthesis of carbon dots exploits the polymerization or carbonization of organic molecules at an elevated temperature [48,49]. Commonly used wet-chemical methods include the hydrothermal/solvothermal approach, chemical oxidation method, and microwave/ultrasound-assisted synthesis route [48].

These luminescent materials are particularly interesting for optical anti-counterfeiting applications due to their rich optical properties and solution processability. For example, halide perovskite QDs show tunable photoluminescence wavelengths with ultra-high brightness, while rare-earth fluoride nanocrystals exhibit upconversion properties with tunable emission lifetimes [42,50–52]. Carbon dots, as an environmentally friendly and low-cost material, can exhibit the attractive optical properties found in halide perovskite QDs and rare-earth fluoride nanocrystals noted above [53], which makes it possible to encode optical information on space or time scales.

3.3 Plasmonic materials

Plasmonics is a subfield of photonics that deals with surface plasmons and the optical properties resulting from light interacting with metal thin films and nanostructures. A critical prerequisite for anti-counterfeiting applications of plasmonic nanostructures is the controllable synthesis of high-quality metallic nanoparticles with desired size and shapes to tailor their distinct plasmonic signatures and near-field enhancements. This can be done by careful optimization of synthesis conditions. Hitherto, a number of wet-chemical methods with well-understood growth mechanisms have been developed for the synthesis of colloidal metal nanoparticles with controllable size and morphology [54–57] since the discovery of gold colloids by Faraday. One of the most commonly used methods is citrate reduction, in which sodium citrate is used as a reductant and stabilizer for the synthesis of spherical gold or silver nanoparticles with average diameters ranging from 8 to 150 nm [19,55,58,59]. Modification of the citrate reduction method by using polyvinylpyrrolidone (PVP) as a capping agent and hydrogen peroxide as an oxidative etchant (called the oxidative etching method) allows the synthesis of high-yield of silver nanoplates [59]. This strategy can also be used to prepare gold triangular nanoplates through the selection of suitable reductant, stabilizer, and oxidative etching agent [60]. Polyol reduction is another well-established method for the synthesis of metal colloids with controllable particle shapes (e.g., quasi nanospheres, nanocubes, nanowires, etc.), in which polyol acts not only as a reducing agent but also a solvent and PVP is usually used as stabilizing agent [19,55,58]. A "seed-mediated" route has been developed for the synthesis of gold nanorods, using sodium borohydride to produce small seeds and ascorbic acid to reduce the metal salt in the growth solution [61]. Surfactant cetyltrimethylammonium bromide (CTAB) can alter the surface energies of the seeds for

different crystallographic planes. The group led by Murray discovered that the introduction of aromatic additives in the seed-mediated synthesis method can dramatically improve the uniformity of gold nanorods [62]. By controlling the size, shape, and composition of the colloidal metal nanoparticles, their surface plasmon resonance peaks can be tailored over the entire visible to mid-infrared region of the electromagnetic spectrum [19,55,58]. The seed-mediated route has been extended to synthesize metallic nanoparticles with complex nanostructures such as core–shell structures with and without a small interior gap [63–67]. The former is actually a nanomatryoshka structure, while the latter is the conventional core–shell structure. To fabricate nanomatryoshka structures, the core nanoparticles are modified with an insulation layer (e.g., DNA, polymer, silica dioxide, and dithiol molecules) and then used as seeds for the shell growth. The insulation layer allows the embedding of Raman-active molecules in the nanogap junction, where the electromagnetic field is localized and enormously magnified. With the built-in hot spots, their SERS signals are enhanced. The concepts of colloidal self-assembly and lithographic fabrication techniques described in Section 3.1 have also been widely adopted for the fabrication of complicated metal nanostructures [12,19,29,56,58,66,68–76].

4. Advanced optical anti-counterfeiting labels

With access to the well-established fabrication techniques presented above, an immense variety of optical nanomaterials with controlled nanostructures (e.g., size, shape, and even periodicity) have been successfully fabricated. These nanomaterials often bear unique and tunable optical properties that are particularly interesting to anti-counterfeiting applications. To harness these properties in a security label, the positioning of the nanostructures to form a 2D pattern on a supporting substrate is a prerequisite. Although top-down lithography techniques can be employed for this purpose, a more cost-effective and widely used technique is printing, in which the nanoparticles synthesized through wet-chemical methods are used as inks [12,18,77]. Another cost-effective approach for the fabrication of 2D security labels is hybrid nanofabrication, which involves the combination of lithographic patterning techniques with colloidal self-assembly methods [25,78]. In this section, selected important encryption mechanisms for anti-counterfeiting applications and corresponding examples of optical security labels fabricated by the above-mentioned techniques are highlighted.

4.1 Conventional optical anti-counterfeiting labels

4.1.1 Structural color-based anti-counterfeiting labels

Structural colors in nature have been known for centuries. With the development of microscopes, especially electron microscopes, the physical mechanisms underlying the brilliant colors of creatures like butterflies [79–81], chameleons [82], and beetles [83] are now well understood. Inspired by nature, materials with artificial structural colors

have been fabricated and applied in the anti-counterfeiting field. Many physical phenomena are responsible for the colors, which can be broadly categorized into optical absorption, scattering, and interference mechanisms. The absorption mechanism mainly involves small metal (plasmonic) nanoparticles and will be discussed in the section of plasmonic anti-counterfeiting. Here we will focus on structural colors originated from scattering and interference.

A typical example of structural color based on scattering was demonstrated by Cao et al. [84], who exploited the resonant light-scattering properties of individual silicon nanowires, as shown in Fig. 2A. Each silicon nanowire acts as a cylindrical cavity resonator that can trap light in circulating orbits by multiple total internal reflections from the periphery, generating a high-field intensity within the nanowire when the wavelength of incident light matches one of the optical resonances supported by the nanowire. Simultaneously, the incident light at the resonant wavelength is strongly scattered back to the far field by the nanowire and can be easily detected by a dark-field microscope (Fig. 2A, left). The wavelength of the light scattered by a silicon nanowire (λ) can be given by the following equation:

$$p(\lambda/n) \approx 2\pi r \tag{2}$$

where p is a positive integer, n is the refractive index of the silicon nanowire, and r is its radius. By simply positioning the silicon nanowires with controllable diameters into a defined pattern on a flat surface through conventional lithography techniques, the wavelength of the scattered light can be tuned over the entire visible light region and an anti-counterfeiting tag with a wide range of vivid colors can be achieved (Fig. 2A, right). A similar resonant light-scattering phenomenon has been reported for single spherical silicon nanoparticles [89], where their observed resonances in the visible spectral range correspond to the excitation of magnetic and electric dipole modes.

Optical interference is the most common mechanism for the generation of structural color. To date, four different types of structures that allow for optical interference have been reported, as summarized in Fig. 2B. Thin films composed of a dielectric layer on a reflective surface (Type I) are one of the most understood interference structures [85,87,90]. As shown in the left panel of Type I, when white light is incident on the thin films, there are two reflection processes at the interfaces of the materials differing by their refractive indices. At each interface, the incident light splits into two components (i.e., transmitted and reflected). The reflected light beams from the two interfaces create a phase-matched condition depending on the thickness of the dielectric layer. The thickness of the layer can be tuned for constructive interference between the two reflected light beams to give a monochromatic color that is discernable by the naked eye. Yakovlev et al. demonstrated optical interference thin films as structure units for color image printing using TiO_2 sol–gel as an ink [85]. Colored interference layers with patterned structures were produced by controlling the thickness of highly refractive layers on the order

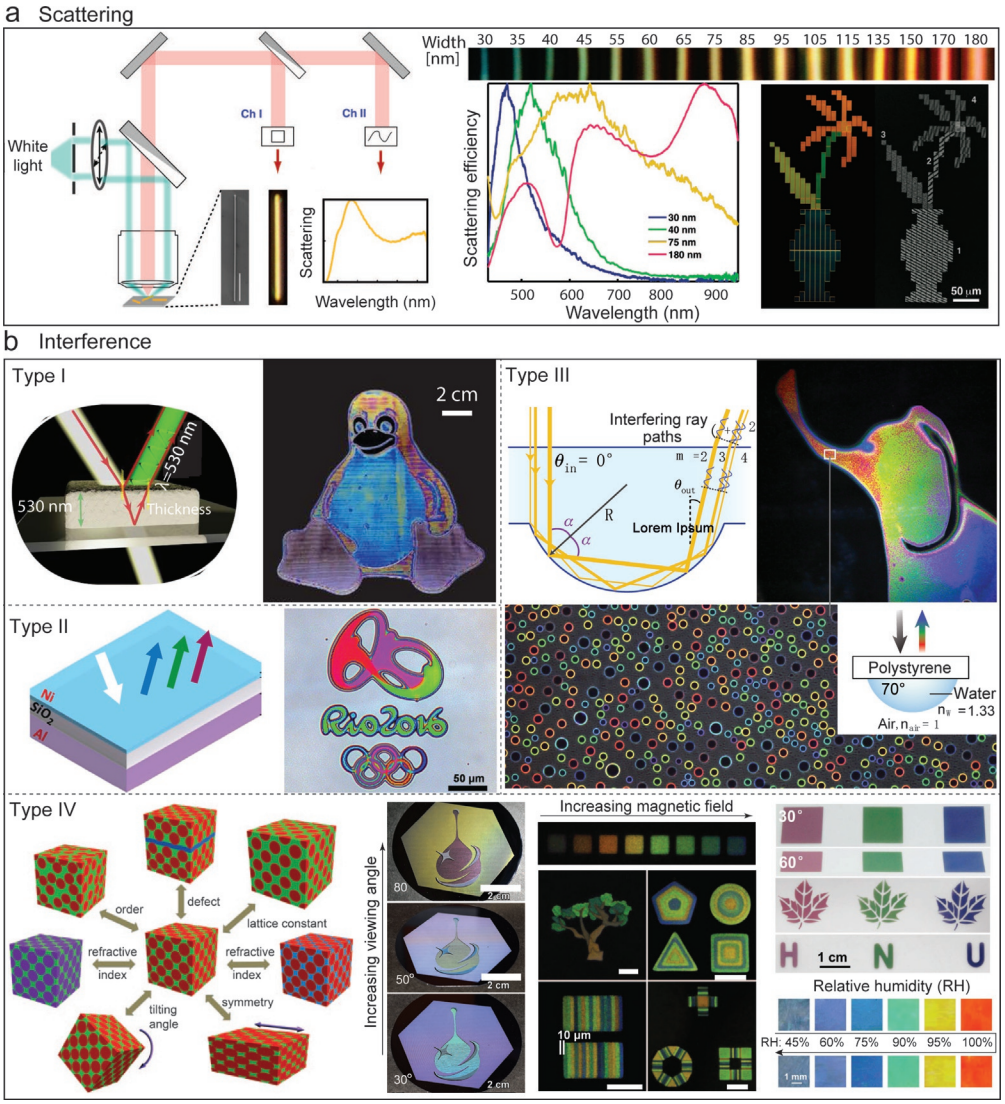


Fig. 2 Optical encryption mechanisms of photonic crystals and examples of the structural colors from corresponding encryption mechanisms: (A) scattering [84], Copyright 2010, American Chemical Society and (B) interference mechanisms: (Type I) a dielectric layer on a reflective surface [85], Copyright 2016, American Chemical Society, (Type II) a dielectric layer sandwiched between two metal reflectors (also called Fabry-Perot-type interference) [75], Copyright 2016, Wiley-VCH, (Type III) Microscale concave structures [86], Copyright 2019, Springer Nature, (Type IV) self-assembled microspheres [25,26,87,88], Copyright 2011, Wiley-VCH, Copyright 2018, Wiley-VCH, Copyright 2009, Springer Nature and Copyright 2018, Wiley-VCH.

of the wavelength through inkjet deposition (Type I, right). Kats et al. demonstrated an ultrathin optical interference coating comprising highly absorbing films of a few to tens of nanometers in thickness on a metal surface, which can be used to produce color images insensitive to surface roughness using lithographic techniques [70,91]. Different from lossless dielectric layers discussed above, the reflection and transmission phase shifts at the interface between the highly absorbing film and air are not limited to 0 or π. These nontrivial interface phase shifts allow the total phase accumulation at the boundary and within the absorbing dielectric film to reach approximately 0 (modulo 2π) for certain films with thicknesses far below $\lambda/4n$.

Fabry-Perot-type interference is found in structures comprising a dielectric layer sandwiched between two metal reflectors (Type II, Fig. 2B), which have been investigated for decades for color filtering [75,92]. Recently, Yang et al. reported an asymmetric Fabry-Perot interference structure comprising a thin layer of nickel film and a thick aluminum film, separated by a lossless SiO_2 dielectric layer (Type II, left) and demonstrated the feasibility of using such structures for high-resolution color printing (Type II, right) [75]. They showed that full color printing with high brightness, saturation, and resolution (>50,000 dpi) could be achieved using Fabry-Perot-type elements as the pigments. The replacement of the lossless dielectric spacing layer with ultrathin amorphous silicon (*a*-Si) in a similar asymmetric Fabry-Perot interference structure enabled the printing of 2D colored images with wide gamut, angle insensitivity, high resolution (up to 150,000 dpi), and enlarged viewing angle [92].

It has been recently reported that optical interference can also occur at microscale concave interfaces with adjacent volumes of high- and low-refractive-index media (Type III, Fig. 2B). As demonstrated by Goodling et al., incident light traveling along different trajectories and undergoing multiple total internal reflections at these concave interfaces allows the total phase accumulated along each trajectory to realize constructive interference (Type III, left top) [86]. The intensity of the total internal reflections from the concave interfaces (I) as a function of wavelength of all incident light paths for fixed angles of light incidence (θ_{in}) and observation (θ_{out}) can be given by the following equation:

$$I = \left| \left(\sum\nolimits_{m,all\ possilbe\ paths} A_m(\alpha_m) \right) \left[r_m\left(\alpha_m, \frac{n1}{n2}\right) \right]^m \exp\left(\frac{2\pi i n_1}{\lambda_0} l_m(\alpha_m) \right) \right|^2 \tag{3}$$

where the index of summation m represents the number of reflections that occur for each trajectory, A_m is an amplitude factor, $\alpha_m = \frac{\pi}{2} - \frac{\pi - \theta_{in} + \theta_{0ut}}{2m}$ is the local angle of incidence, r_m is the complex Fresnel reflection coefficient, n_1 and n_2 are the refractive indices of the medium above and below the interface, respectively, λ_0 is the wavelength of light in air, and $l_m = 2mR\cos(\alpha_m)$ is the physical path length of each trajectory (R: the optical interface's radius of curvature). This was the first report to unveil the mechanism behind

the optical phenomenon that light was only observed near the solid–liquid contact line and this effect is dependent on sessile-droplet diameter and contact angle. Typical examples of concave structures include multiphase droplets, 3D patterned polymer surfaces, and solid microparticles. By taking advantage of this unique optical phenomenon, colorful images with defined patterns (e.g., a patterned shape of an elephant) composed of concave structures have been produced (Type III, right top). Following the work of Goodling et al., Fan et al. fabricated an array of concave structures by the transfer of a partially embedded polymeric microsphere monolayer onto the sticky surface of a transparent tape and demonstrated the potential application of such structures for smart traffic safety and advertisement displays at night [30].

Among the existing interference systems, the most widely studied is photonic crystals that comprise self-assembled microspheres (Type IV, Fig. 2B). The structural colors of such photonic crystals are attributed to interference and reflection, which can be described by the Bragg-Snell law [26]. As discussed earlier, this law can be approximately expressed as Eq. (1). Based on this equation, the parameters that affect the optical properties of photonic crystals are schematically summarized in the first panel of Type IV. They are lattice constant, effective refractive index, crystal structure, incidence/viewing angle (or titling angle of photonic crystals when the direction of incident light is fixed), the degree of order and defects in photonic structures. The lattice constant is the parameter that has been the most intensively employed to control the color of the photonic crystals. This parameter can be tailored through mechanical deformation, swelling and deswelling, heat-induced polymer volume or phase transition, and charge-induced expansion and contraction. Since colloidally stable inks of microspheres can be easily prepared using appropriate solvents as dispersing media, inkjet printing has been widely adopted for the fabrication of 2D and 3D photonic crystal-based anti-counterfeiting labels [21,36,37,87,93,94]. For example, Keller et al. fabricated stable inks for inkjet printing of short-range ordered 2D photonic crystals with controllable layer numbers on various substrates (e.g., Si wafer, glass slide, and glossy photo paper) without any surface treatment [87]. The printed opal holography on a silicon wafer shows multiple bright colors with a wide angle dependence in daylight (Type IV, column 2) [87]. With the help of infiltration, full-spectrum noniridescent structural color patterns can be rapidly printed on liquid-permeable substrates on a large scale [27,95]. It has been reported by Nam et al. that when the thickness of well-ordered 2D photonic crystal patterns is controlled down to a single layer, the resulting anti-counterfeiting labels can achieve covert-overt transformation by simply changing the background and/or illumination intensities [36]. Kim et al. developed a magnetically tunable and lithographically fixable material called "M-ink" and used this material to create high-resolution patterns with brilliant colors through a maskless lithographic technique (Type IV, column 3) [25]. The degree of order in "M-ink" and consequently its colors can be reversibly controlled with an external magnetic field. Photochemical fixing of the magnetically tunable "M-ink" in a

photocurable resin was employed to generate multicolored patterns. Such a robust and scalable structural color printing technique together with low-cost "M-ink" material offers a great opportunity to manufacture colorful security labels for general consumer goods. In addition, defects can be easily created in photonic crystals by doping and used to tune their optical properties. Zhang et al. reported brilliant noniridescent structural colors with enhanced brightness by doping amorphous photonic structures with graphene nanosheets containing a fraction of graphene quantum dots (GQDs) [88]. The doped amorphous photonic structures showed vivid structural colors with high color contrast and saturation but low dependence to viewing angle (Type IV in Fig. 2B, the upper part of column 4). It is believed that the uniform light absorption ability of grapheme nanosheets in the whole visible spectrum and the remarkable excitation tunable emission of GQDs are responsible for the high color saturation and brightness. When further filling the voids of these photonic structures with a humidity-sensitive hydrogel, their colors can be dynamically tuned by changing the humidity surrounding the photonic structures (Type IV, the bottom part of column 4), which allows for the development of dynamic anti-counterfeiting labels. The dynamic color change originates from the variation of the lattice constant due to the swelling and deswelling effect.

The structures of photonic crystals such as the symmetry and the shapes of the building blocks also have a dramatic effect on their optical properties, as they can cause anisotropic lattice constants. Except for colloidal self-assembly approaches, top-down techniques, for example holographic lithography, standing-wave lithography, and two-photon polymerization lithography, enable the fabrication of 3D photonic crystals with controllable lattice that shows a continuous hue [22,97]. Ito et al. reported the production of brilliant structural colors by creating periodic microfibrillation in glassy polymer films through standing-wave lithography (Fig. 3A) [96]. As shown in the top panel of Fig. 3A, a standing-wave pattern can be formed in a photosensitive polymer film when the reflected wave from the resist–substrate interface interferes with the incident wave. This causes the selective crosslinking of the photoresist in alternating layers in the film, leading to the generation of residual stresses across the non-crosslinked layers of the photoresist. Upon a weak solvent exposure, periodic microfibril layers with random porous features can be formed, which is triggered by the stress applied to the non-crosslinked layers. Combined with conventional photolithography, various colorful patterns can be produced (Fig. 3A, bottom). Direct laser writing by multiphoton polymerization of a photoresist has been developed for the rapid, cheap, and flexible fabrication of photonic structures with lattice constants in the micrometer scale and thereby the optical response in the infrared spectral region [97]. Liu et al. developed an annealing strategy to shrink the photonic crystals fabricated by the direct laser writing technique to reduce the lattice constant (Fig. 3B) [23]. Woodpile photonic crystals with lattice constants in the range of 305–672 nm have been realized. The resulting structures show optical responses in visible spectral region originating from Bragg diffraction. Any 3D microscopic models with woodpile structures of desired colors can be generated (e.g., the Eiffel Tower shown

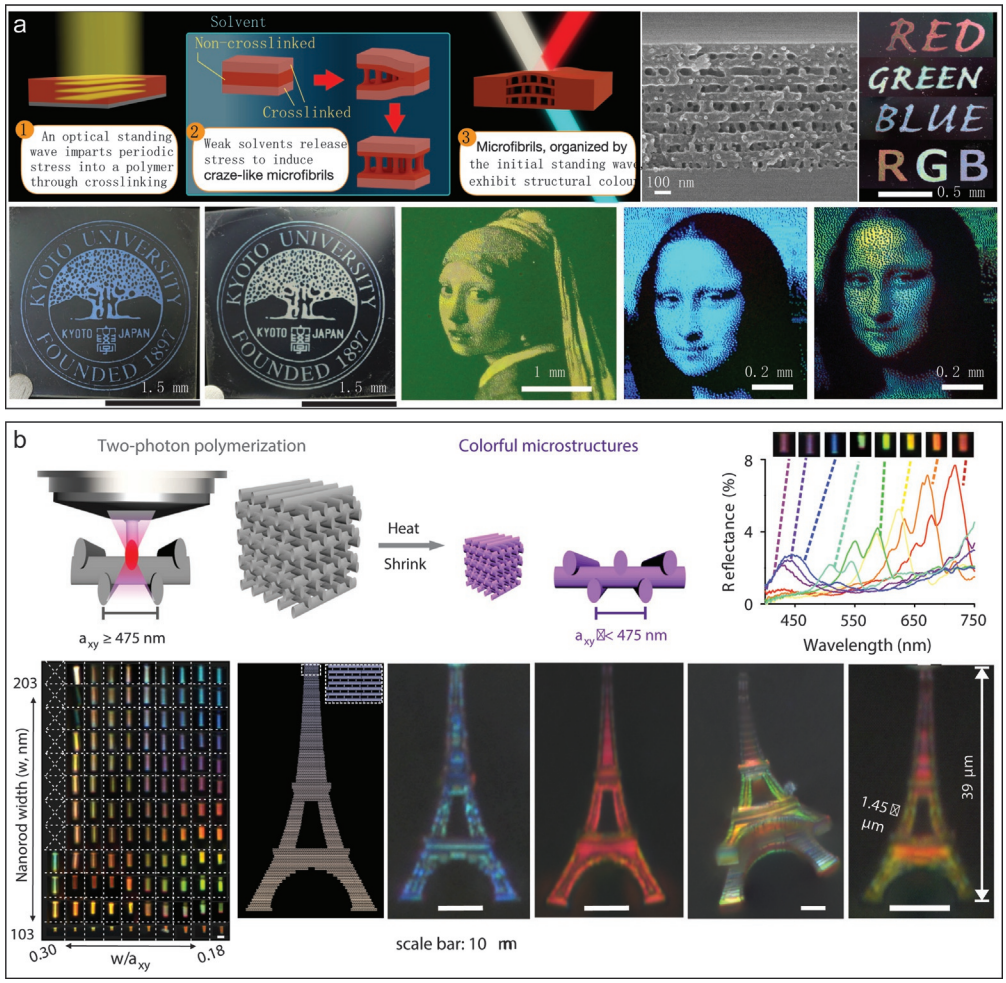


Fig. 3 Structural colors from 3D photonic crystals. (A) Periodic microfibrillation in glassy polymer films fabricated through standing-wave lithography [96], Copyright 2019, Springer Nature and (B) lattice constant shrinkable woodpile photonic crystals fabricated by direct laser writing technique [23], Copyright 2019, Springer Nature.

in Fig. 3B). It should be noted that both standing-wave lithography and heating-assisted two-photon polymerization lithography could induce nonuniform anisotropic lattice constants within the produced photonic crystals and cause local deviation of colors, which could be potentially used in unclonable anti-counterfeiting.

4.1.2 Luminescent anti-counterfeiting labels

Lanthanide-doped nanoparticles are known for their multidimensional emission properties, including emitting color, ratio-metric intensities from multiple emission bands, and luminescent lifetimes, originating from the lanthanide ions. Such ions possess a wealth of

electronic transitions within their 4f electron shells, showing characteristic "ladder-like" multiple excited states (Fig. 4A, top left). One of the merits of lanthanide-doped upconversion nanoparticles for anti-counterfeiting applications is that their optical responses are tunable by ion doping. As illustrated in Fig. 4A, the doped ions can be categorized into sensitizer and activator, which facilitate light absorption and emission, respectively. The absorption of the sensitizer mainly determines the excitation wavelength of lanthanide-doped nanoparticles. The most commonly used sensitizers are lanthanide ions having strong absorption in the near infrared (NIR) region such as Yb^{3+} and Nd^{3+}. NIR lasers with wavelengths of 808 and 980 nm are typical excitation sources for downconversion (converting a high-energy photon to two lower-energy photons) [104] and upconversion (the opposite process of downconversion) [105,106] to occur. Other sensitizers like Eu^{3+} and Ce^{3+} can efficiently absorb ultraviolet (UV) light, allowing for downshifting (converting a high-energy photon to a lower-energy one) [104,106]. In addition, some lanthanide-doped nanoparticles are also scintillating materials that can convert X-ray to visible light [107].

Lanthanide-doped nanocrystals have been widely studied in X-ray luminescence [108]. For example, Ou et al. fabricated lanthanide-doped $NaLuF_4$ nanoparticles with ultralong-lived X-ray luminescence (lifetime up to 30 days) [98], which can be potentially applied in the anti-counterfeiting field (Fig. 4A, right). Moreover, the phenomena of downshifting and upconversion have been used as dual optical responses that are excited under different conditions to increase the anti-counterfeiting levels [105]. Recently, Liu et al. demonstrated the simultaneous downshifting/upconversion from lanthanide-doped core/shell fluoride nanoparticles for multimode anti-counterfeiting [109]. They chose $LiLuF_4/LiYF_4$ core/shell nanoparticles as a host material and doped two types of lanthanide-ion pairs in the shell. The first type of doping ion pairs are Eu^{3+}/Ce^{3+} and Tb^{3+}/Ce^{3+} that enable downshifting fluorescence, and the second type of doping ion pairs are $Er^{3+/}Yb^{3+}$, Ho^{3+}/Yb^{3+}, and Tm^{3+}/Yb^{3+} that allow for upconversion fluorescence. It should be noted that the lanthanide-ion pairs for upconversion nanoparticles generally act as sensitizing and emitting centers that absorb the energy of NIR photons to emit light in the visible ranges. The resulting materials can change color from green to yellow to orange to red or from violet to white to green when varying the excitation sources and their intensities. In other work by Xie et al., novel lanthanide-doped core–shell nanoparticles with dual downshifting and upconversion luminescence were designed and fabricated. The nanoparticles were used as environmentally, friendly luminescent inks to produce dual-mode emission patterns on papers by printing [100]. As shown in Fig. 4A, the printed picture of a temple shows different multicolored patterns under the excitation of a 980-nm laser and 254 nm UV light. Full-color features can also be achieved from multi-shell upconversion nanoparticles under excitation of two-pulsed lasers with different wavelengths [106].

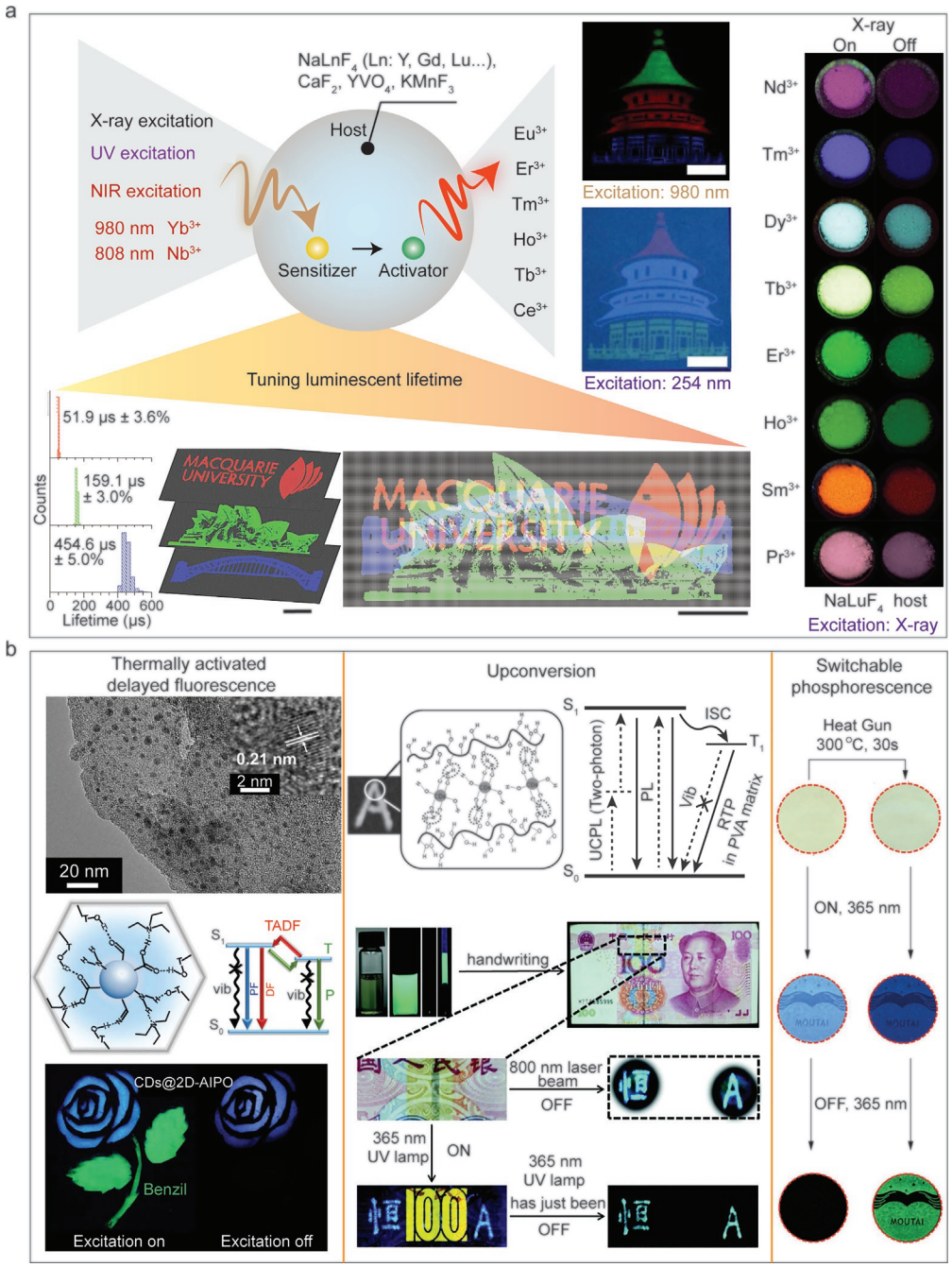


Fig. 4 Luminescent encryption using (A) lanthanide-doped nanoparticles [9,98–100], Copyright 2019, Wiley-VCH, Copyright 2021, Springer Nature, Copyright 2013, Springer Nature, Copyright 2019, The Royal Society of Chemistry and (B) carbon dots for anti-counterfeiting applications [101–103], Copyright 2017, the Authors. Published by American Association for the Advancement of Science (AAAS), Copyright 2016, Wiley-VCH, Copyright 2018, Wiley-VCH.

It has been reported that the emission colors of lanthanide-doped upconversion nanoparticles can be modulated by temperature [110]. Heat-induced luminescence quenching is a well-known phenomenon due to the non-radiative combination of photoexcited carriers. As expected, such thermal quenching behavior has been commonly observed in $NaYF_4$: Yb^{3+}, Er^{3+} upconversion nanoparticles [111]. Recently, an anomalous heat-induced luminescence enhancement (also called inverse thermal quenching effect) was reported by the group of Shao for $NaYF_4$: Yb^{3+}, Er^{3+} upconversion nanoparticles smaller than 30 nm [50]. Since then, several kinds of small-sized upconversion nanoparticles with a thermal luminescence enhancement effect have been reported [40,50]. Interestingly, this abnormal thermo-enhanced luminescence becomes more significant as the size of the nanoparticles decreases [40] and can be further magnified by low-valence ion doping [112,113]. A similar phenomenon was also found in Na_3ZrF_7: Yb^{3+}, Er^{3+}/Ho^{3+}/Tm^{3+} upconversion nanoparticles [114]. Several mechanisms have been proposed to explain the inverse thermal quenching effect: (1) phonon-promoted energy transfer from Yb^{3+} sensitizers to activating ions [115]; (2) the removal of luminescent quencher—water molecules physically adsorbed onto Yb^{3+} on the nanoparticle surface—at an elevated temperature [116]; and (3) defect-assisted electron capture from the excited states of Yb^{3+} sensitizers and then the release of the trapped electrons to the adjacent excited states of activators gradually with a rise in temperature [78]. By taking advantage of this inverse thermal quenching effect, anti-counterfeiting labels that can recover hidden luminescent information [99] or dynamically switch between two luminescent colors [117] through modulation of temperatures have been developed.

Apart from emission colors, the luminescent lifetime of upconversion nanoparticles can also be finely tuned by lanthanide-ion doping [99], which makes it possible to encrypt optical information on time scale. Lu et al. reported the tuning of the luminescent lifetimes of upconversion nanoparticles from a few microseconds to several hundreds of microseconds by controlling the concentration of activator ions and demonstrated lifetime multiplexing using the upconversion nanoparticles as information carriers (Fig. 4A, bottom left) [99]. It is well known that the luminescence of upconversion nanoparticles relies on energy transfer from the sensitizer to the emitter ions. Manipulating the sensitizer-emitter distances allows control over their luminescent lifetime. As proof-of-concept, three layers of patterns made of upconversion nanoparticles with different luminescent lifetimes have been produced. The hidden information in the covert patterns was unlocked by monitoring the lifetime values from each pixel of each layer. To increase coding capacity, it is straightforward to combine emission colors and luminescent lifetimes for optical encryption from a single type of upconversion nanoparticles [116]. This was achieved by the design of core/multi-shell upconversion nanoparticles with controllable energy relay paths [106]. For example, Liu et al. synthesized hexagonal $NaGdF_4$:Mn@$NaYF_4$:Yb/Tm-$NaYF_4$ core@shell–shell nanoparticles that can be excited under both 808 and 980 nm irradiation [112]. It was believed that such a design

could effectively separate Yb^{3+} and Mn^{2+} ions, thus preventing the back-energy transfer from Mn^{2+} to Yb^{3+}. This is of importance to realize a long-lived Mn^{2+} luminescence and short-lived lanthanide upconversion emission in the same particles.

Quantum dots (QDs) are extremely small semiconductor nanoparticles that can emit light of specific wavelength with high quantum yield. Widely studied QDs include toxic II–VI and perovskite QDs, and environmentally, friendly carbon dots. The former is a good ink material for the fabrication of microscopic security labels [18], but not macroscopic ones due to their high toxicity. In this section, we will highlight the key optical characteristics, such as downshifting, upconversion, and phosphorescence, of carbon dots toward their anti-counterfeiting applications [53]. Low toxicity and biocompatible carbon dots with different emission colors have been synthesized and used as inks for luminescent pattern generation [53,118,119]. Lou et al. reported that the aggregation of carbon dots within predefined patterns caused luminescence quenching [49]. Upon exposure to water, the luminescence recovered due to the redispersion of the carbon dots in the fiber network of the paper. The afterglow phenomenon of carbon dots including TADF and phosphorescence has also been observed when they are embedded in nanoconfined spaces of zeolitic crystalline matrices (Fig. 4B, left). As reported by Liu et al., the carbon dots@zeolite composites showed a high quantum yield of ~52% and a long lifetime of 350 ms [101]. The authors demonstrated fluorescence and afterglow dual-responsive security labels using these carbon dots@zeolite composites as information carriers. By simply replacing zeolite with boric acid [120] or silica [121], ultralong lifetime and efficient room temperature phosphorescent carbon dots have been achieved. The group of Lin reported that N-doped and N, P-codoped carbon dots themselves showed strong and long lifetime phosphorescence in an ambient environment [122]. Luminescent inks composed of these carbon dots and polyvinyl alcohol (PVA) have been prepared for the printing of optical patterns (Fig. 4B, middle and right) [102]. Fluorescence, upconversion, and phosphorescence triple-mode security labels have been realized for N-doped carbon dots embedded in PVA [102]. Phosphorescence is very sensitive to oxygen and moisture, and the exposure of phosphorescent materials to highly humid environments can cause phosphorescence quenching. By simply drying the security patterns by heating, however, the phosphorescence was easily recovered (Fig. 4B, right), which allows for dynamic anti-counterfeiting [103].

4.1.3 Plasmonic anti-counterfeiting labels

Metallic nanostructures, including colloidal and lithographically defined nanoparticles, support localized surface plasmon resonances, which give rise to color. The colors are largely determined by the structures of the metal nanoparticles, such as the size, morphology, interparticle separation, and assembly configuration [56]. The use of metal nanostructures to generate plasmonic colors has a long history. In ancient Rome, gold nanoparticles were used to stain glass for aesthetic purposes. A famous example of the

aesthetic use of gold nanoparticles is the beautiful Lycurgus Cup that appears green in reflected light and red in transmitted light. There is a growing interest in plasmonics for color-based applications driven by their advantages over conventional pigments, which include rich everlasting colors without photobleaching, with high contrast and spatial resolution [73,123–126].

Gold nanorods are particularly interesting for color-based applications as their optical properties are highly sensitive to their aspect ratio, allowing their plasmon absorption peak to be tuned throughout the entire visible spectrum [19,42]. Advances in nanofabrication techniques allow the precise control over nanorod assembly onto nanopatterned substrates [127]. For example, Kuemin et al. demonstrated the successful self-assembly of gold nanorods into the recessed regions of a patterned substrate in an oriented fashion through capillary assembly (Fig. 5A) [127]. As shown in Fig. 5A, as the three-phase contact line moves over the surface of a patterned substrate with recessed features, gold nanorods are deposited in the desired regions, while the excess nanorods are dragged back to the colloidal solution by the interfacial capillary forces. The recessed holes within any lithographically defined 2D patterns (e.g., WALK Ampelmann) acted as trapping sites for aligned assembly of the nanorods on the single-particle level. Each pixel was composed of a single nanorod that changes color from green to red when rotating the polarizer by 90 degree. Thai et al. extended this strategy to fabricate vertically aligned gold nanorod arrays on patterned substrates [129]. These standing gold nanorod arrays offer large SERS enhancement and can be potentially used for SERS-based anti-counterfeiting applications.

It is well known that gold nanorods can transform into spherical nanoparticles upon photothermal treatment and display completely different plasmon absorption profiles. Based on this phenomenon, Zijlstra et al. developed a new technique integrating of wavelength, polarization, and spatial dimension to produce multiplexed images [130], which can be readily used in optical patterning, encryption, and anti-counterfeiting. The encoded information density can easily go beyond 10^{12} bits per cm^3 (1 T bit cm^{-3}), ensuring high security level for anti-counterfeiting applications. In contrast to periodic plasmonic nanoparticle arrays shown in Fig. 5A, disordered systems have also shown color tunability. For example, Mao et al. recently demonstrated the tunable reflection colors of disordered plasmonic nanoparticles by controlling the coupling to an external cavity [131]. The cavity was created by sequential deposition of a silver film, LiF spacer, and disordered silver nanoclusters on a solid substrate. By tuning the thickness of the cavity between the silver mirror and the layer of disordered silver nanoclusters, the system exhibiting broadband adsorption underwent a phase transition to one with finite reflection bands. These plasmonic color patterns could serve as security labels due to their unique far-field optical appearance.

Top-down lithographic methods are able to produce periodic plasmonic nanostructures that can generate colors in high resolution for color-based devices in combination

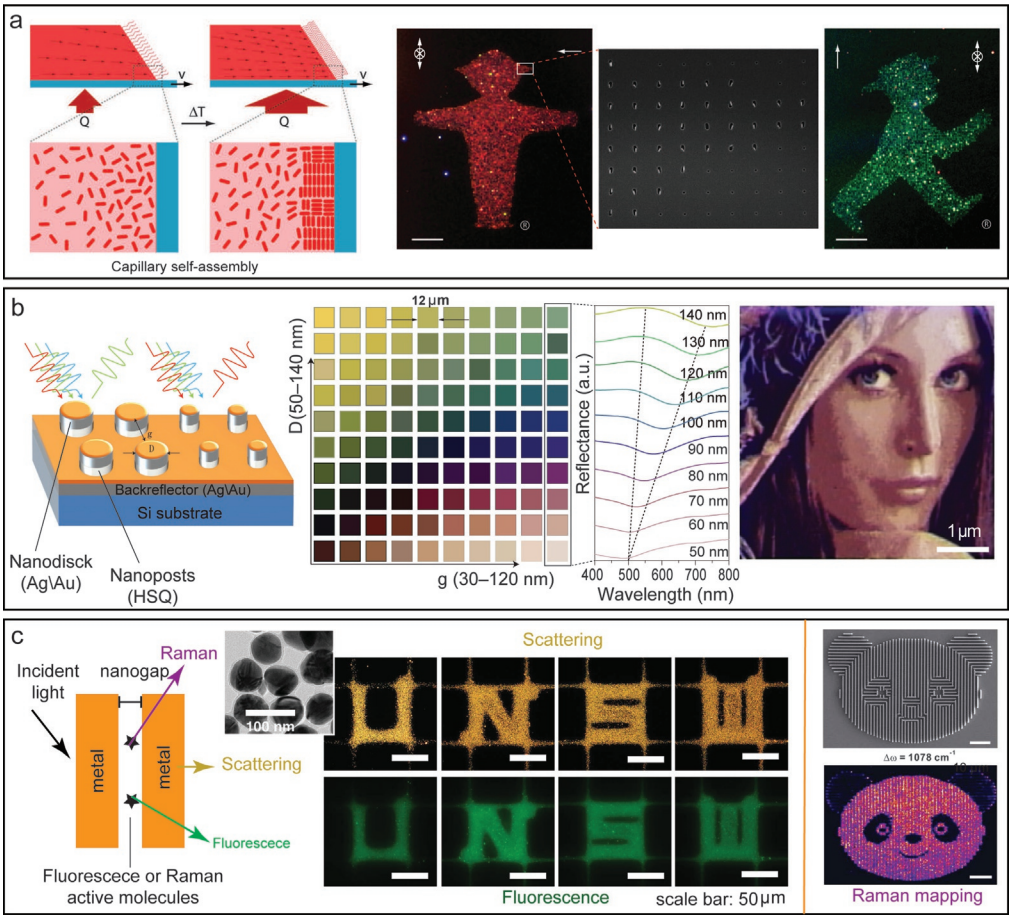


Fig. 5 Examples of static plasmonic colors, fluorescence image, and Raman mapping from metal nanostructures. (A) Oriented gold nanorod arrays fabricated by template-assisted capillary self-assembly [127], Copyright 2012, Wiley-VCH, (B) gold\silver nanodisc arrays with controlled diameter and interparticle distance fabricated by electron-beam lithography technique [128], Copyright 2012, Springer Nature, and (C) dye-embedded silver@silica core–shell spherical nanoparticle arrays fabricated by shadow-mask-lithography-assisted self-assembly (*left*) [12] Copyright, 2016, Wiley-VCH, and silver nanowire patterns with sandwich structures fabricated by direct laser writing technique (*right*) [124], Copyright 2015, The Royal Society of Chemistry.

with thin-film deposition techniques [73,76,132]. The plasmonic colors generated through lithographic methods overcome the limitations of conventional color printing such as insufficient resolution and the fading of dye pigments and lack of multicolor in self-assembled nanoparticles. Aluminum nanorods with tunable dimensions and separation have been lithographically patterned to form periodic arrays on a flat substrate [133]. The color of the nanorod arrays is determined by their length, aspect ratio, and

interparticle distance. Both the longitudinal plasmon resonance and far-field diffractive coupling are responsible for the vivid colors. When the nanorod arrays were placed on an elastic substrate, the interparticle spacing can be tuned by stretching the substrate in different directions [134]. Consequently, the scattering color can be modified to blue or red from the initial color of the at-rest structure. Well-ordered arrays of other metal nanostructures (e.g., metal nanodisks [69], nanoholes [135,136], anisotropic cross-shaped antenna [137], and nanoaperture structures [136]) have also been lithographically fabricated. The scattering strength of the nanoantenna resonators can be enhanced by hybridization with a thin metal film (also called metal back reflector). For example, Kumar et al. demonstrated printing color at the optical diffraction limit using hybridized nanodisk arrays (i.e., nanodisk array sitting above a metal back reflector separated by a spacer) as color pixels (Fig. 5B) [128]. Since the color of the hybridized nanodisks is sensitive to their dimensions, the full palette of colors can be achieved by tuning the parameters of the nanodisks, such as their diameter and interparticle spacing. This allows for the generation of bright-field color prints with resolutions up to ~100,000 dpi. The color print can be used as anti-counterfeiting marks as it carries a large amount of optical information that is difficult to duplicate.

The interaction between light and metal nanostructures not only gives rise to color but also generates a strong near field in close vicinity to the metal surface. Integration of plasmonic nanostructures with both fluorescence and Raman-active molecules enables the system to simultaneously show far-field scattering, fluorescence, and Raman triple optical signals, as illustrated in the left panel of Fig. 5C. The fluorescence and Raman signals are particularly strong when the molecules are located in the small gaps (i.e., hot spots) of two adjacent nanoparticles [12]. The multiple optical signals of the emitter coupled plasmonic nanostructures can be assigned as spectral codes for anti-counterfeiting applications. We have developed a multiplexed security label that carries triple graphical and spectral information in the same device [12]. SERS encryption alone has also been widely studied for anti-counterfeiting applications [68,124,138]. A notable example is the SERS-based nanodisk codes reported by Qin et al., which are composed of a graphic Raman "barcode" and a spectral Raman fingerprint [69]. The encoding elements are made of dispersible arrays of gold nanodisks that are prepared by on-wire lithography. Raman-active chromophores—methylene blue and *para*-mercaptoaniline—are loaded in the small nanogaps of two neighboring gold nanodisks. Other metal nanostructures such as silver nanowires, sandwich structured nanowires, and plasmene nanosheets (2D plasmonic nanomaterials) have been developed as plasmonic nanoantenna for SERS-based anti-counterfeiting applications [68].

The realization of dynamic color generation is of importance for high security level anti-counterfeiting. So far, several concepts for the generation of dynamic plasmonic colors have been proposed that involve the reversible modification of the size and interparticle distance of metal nanoparticles, dielectric property of surrounding medium, and

the phase of the materials [29]. The most straightforward method for reversible control of the size of metal nanoparticles is electrochemical deposition, which allows a precise and controlled growth of nanometer-thick layers of metals on predefined metal islands [65,141]. Wang et al. applied the electrochemical deposition technique to generate highly ordered plasmonic nanodomes in real time that permit dynamic color modulation (Fig. 6A) [82]. The authors used an anodized aluminum oxide template on 50-nm-thick SiO_2-coated conductive ITO substrate as an etching mask to produce a SiO_2 nanohole array and then deposited a thin layer of gold into the nanoholes, to form an array of gold nanodomes. Subsequently, a silver shell with controllable thickness was electrochemically deposited onto the nanodomes from a silver-ion containing electrolyte and then dissolved as demanded. The color of the electrode is determined by the thickness of silver shell and can be reversibly switched from blue to red. A mechanical chameleon was used as a model to demonstrate the dynamic color change based on the well-ordered gold/silver nanodome arrays (Fig. 6A, right).

An alternative way to generate dynamic plasmonic colors is the reversible tuning of the interparticle distance of metal nanoparticle assemblies [131], which can be achieved by altering the electrostatic interaction between the adjacent nanoparticles or the length of thermoresponsive surface ligands. For example, Liu et al. reported a hydrolysis-induced electrostatic strategy to dynamically control the spacing between strongly coupled nanoparticles within a silver nanoparticle film and therefore the plasmonic colors (Fig. 6B) [139]. The metal nanoparticle film was fabricated by positioning poly(acrylic acid) (PAA)-coated silver nanoparticles on top of a layer of sodium borate ($Na_2B_4O_7$) residing on a glass substrate. Upon exposure to moisture, the sodium borate hydrolyzed into H_3BO_3. The released OH^- ions deprotonated the carboxyl groups on the capping agent (i.e., PAA), increasing the electrostatic repulsion between the silver nanoparticles in the film owing to the increase of the surface charges. Consequently, the nanoparticles were pushed away from each other and their coupling strength weakened. This led to the gradual change of the extinction peak of the silver nanoparticle film from 526 nm (pink) to 423 nm (yellow) in 220 ms. Even a small change in distance resulted in large blue shifts in the order of 100 nm. This process is highly reversible, and the initial state could be returned in 640 ms, with outstanding repeatability in color switching over 1000 cycles. This hydrolysis-induced electrostatic strategy for dynamic color control is attractive for anti-counterfeiting applications, owing to its sustained reversibility, ease of scalable fabrication, and fast switching times.

A reversible change of refractive index of the medium surrounding nanostructured metal surfaces is another important strategy to dynamically generate plasmonic colors. Liquid crystal (LC) is a functional optical material that allows electrical switching of its orientation and anisotropic refractive index. Based on this phenomenon, Franklin et al. developed a LC-plasmonic coupled system by integrating the LCs with arrays of shallow aluminum nanowells (Fig. 6C) [125]. The nanowells were imprinted on a

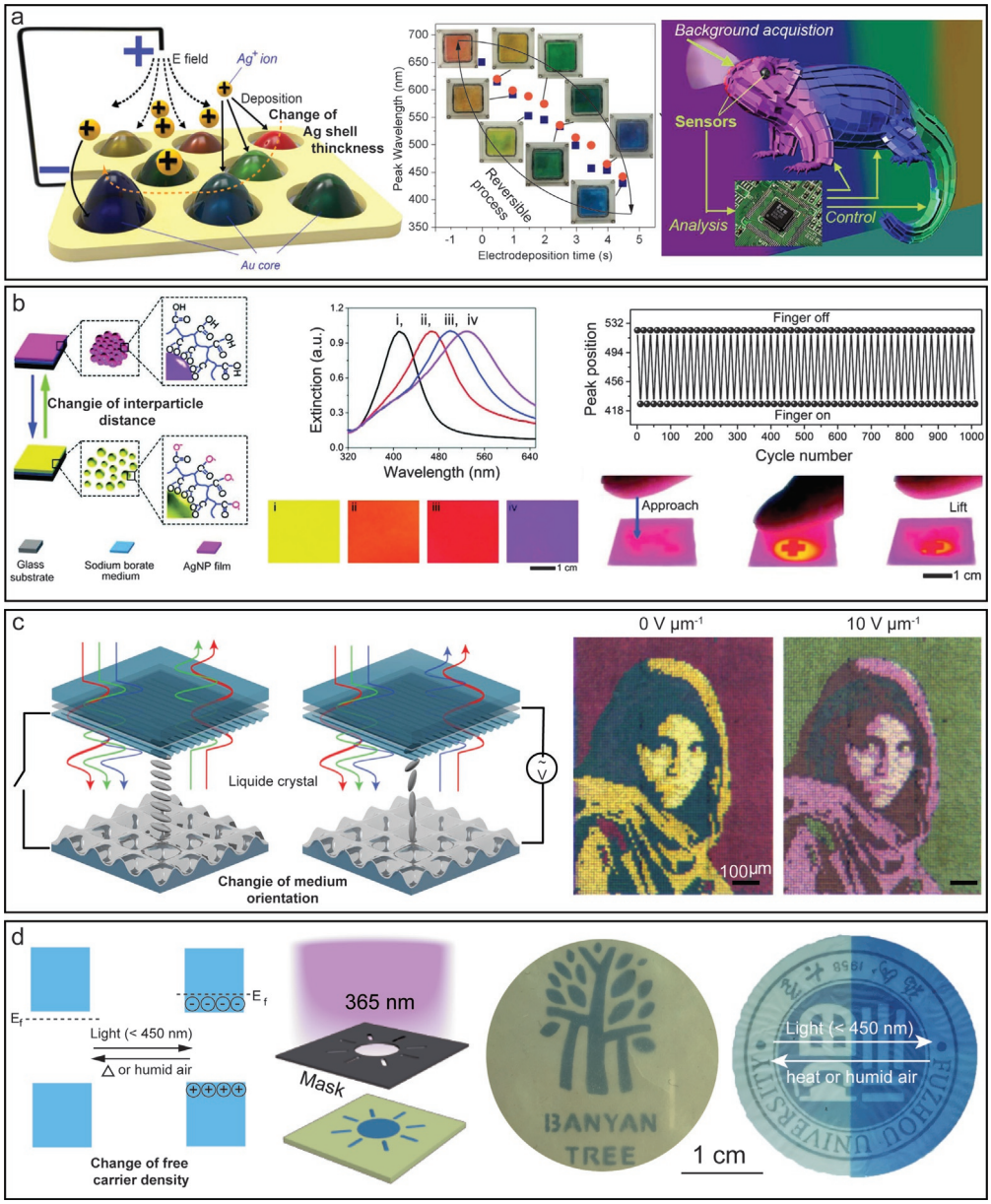


Fig. 6 Examples of dynamic plasmonic colors originated from the reversible modification of (A) the size [82], Copyright 2016, American Chemical Society, (B) interparticle distance of metal nanoparticles [139], Copyright 2019, Wiley-VCH, (C) dielectric property of surrounding medium [125], Copyright 2015, Springer Nature, and (D) the phase of the materials [140], Copyright 2021, Elsevier.

polymer and covered with a smooth layer of aluminum and used as the bottom electrode. An indium tin oxide (ITO)-coated glass substrate was used as the counter electrode. A high birefringence (Hi-Bi) LC was filled in the cell in direct contact with the aluminum surface. When an external voltage was applied, the LC molecules began to vertically rotate from their initial state, until they all aligned along the electrical field between the two electrodes. After the removal of the voltage, the LC molecules returned to their original state. The potential-driven reorientation of LCs changed the surrounding refractive index. The surface plasmon resonance frequency is highly sensitive to this parameter, resulting in a change in the observed color. Based on this mechanism, the authors fabricated a plasmonic color micro-image of the Afghan girl (Fig. 6C, right), which could be electrically modulated [125]. The group took this approach a step further to produce single plasmonic pixels for full RGB color generation by exploiting the polarization rotation of LC in addition to the anisotropic refractive index [76]. By combining these two effects, the reflective color of a single plasmonic pixel was dynamically tuned over the full visible spectral range as a function of the external voltage applied.

According to the Drude–Lorentz free electron model, a metal is considered a plasma consisting of positive ions fixed at the crystal lattice and free electrons with equal number of opposite charges. Under light radiation, the free electrons within the metal coherently oscillate at a frequency of ω_p relative to the lattice of positive ions. For bulk metals, ω_p can be expressed as the following equation:

$$\omega_p = \sqrt{\frac{N \cdot e^2}{\varepsilon_0 \cdot m_e}} \tag{4}$$

where N is the density of free electrons, ε_0 is the dielectric constant of a vacuum, e is the charge of an electron, and m_e is the effective mass of an electron. For a spherical metal nanoparticle, the theoretical frequency of its localized surface plasmon can be approximated to $\omega_p/(1+2\varepsilon_m)^{1/2}$ (note that ε_m is the dielectric constant of the surrounding medium). In other words, the electron density N of a metal nanoparticle determines its plasma frequency ω_p and thus the plasmonic resonance frequency of the metal nanostructure. Reversible phase change materials are substances that can undergo phase transition from one type of material to another under external stimuli, (e.g., light, heat, magnetic field, and so on). These materials may offer an opportunity to dynamically adjust the electron density N. Recently, we reported a highly crystalline carbon nitride that can undergo phase change from semiconductor to metalloid upon UV irradiation in an inert environment (Fig. 6D) [140]. When it was wrapped by cellulose nanofibers, the material changed color from yellow to blue upon UV illumination and reversibly switched back to its original color upon exposure to heat or moisture. The color change was ascribed to hot electron-induced surface plasmon resonance, and the hot electrons were found to exist for 5 days after the removal of the UV source. The photochromic

paper was applied to color displays and dynamic security labels. Compared with electrical modulation, optical modulation is a simpler way to encrypt optical information and is compatible with the existing optical anti-counterfeiting techniques. Other phase transition materials, like magnesium hydride, have also been explored for dynamic plasmonic color generation [32]. Upon hydrogenation and dehydrogenation, magnesium nanostructures alter their crystallographic and electronic structure, resulting in significant changes in optical properties along with the metal to insulator transition.

4.2 Physical unclonable function (PUF)-based optical anti-counterfeiting labels

All the security labels discussed above that use the "collective behavior" of optical nanomaterials for information encryption represent the traditional approach to anti-counterfeiting. The graphical patterns of these traditional security labels can be authenticated either on visual inspection or using optical microscopes. The anti-counterfeiting level of these security labels is determined by the complexity of the ink formulation, the limited access to the chemical components used in the ink, and/or lack of access to expensive lithographic fabrication techniques. Although the traditional security labels pose technical barriers for counterfeiting, they can still be duplicated due to their predicable and deterministic decoding mechanisms. The PUF-based encryption strategy is an ideal solution to the drawback of clonability of the traditional security labels.

Chaos, especially the disorder or randomness of a system, is a common phenomenon in nature. For instance, no two identical natural objects or creatures can be found. Taking advantage of the disorder or randomness of a system as a PUF for information encryption enables the creation of unique and unclonable security labels. This concept was pioneered in 2002 by Pappu et al., who used a laser beam to scan polymer block containing disordered glass spheres to generate randomly fluctuated scattering signals [17]. The collection of the scattered light with a charge-coupled device (CCD) camera produced a speckle pattern, which was then transformed to 1D or 2D PUF keys. Any rough surface (e.g., the surface of a plastic card or a piece of paper) can also produce a speckle pattern [13,142]. The security of this technique ultimately relied on the unlimited distribution possibility of the scatterers in the medium or on the surface and the difficulty of recreating their microstructure down to atomic length scales. Since the first report on PUF encryption, the field has received much interest, devoted to the development of random structures with specific optical responses [17]. All the optical responses (i.e., scattering/reflection, luminescence, and SERS) that are used in the conventional anti-counterfeiting labels have been exploited for PUF-based encryption and anti-counterfeiting [13,18,140,143].

Artificial fingerprints made by polymer wrinkle films are also a random scattering structure and have been widely investigated for PUF-based anti-counterfeiting

[144,145,147,148]. In 2015, Bae et al. reported artificial fingerprints made by polymeric particles with stochastically generated silica film wrinkles (Fig. 7A, left) [145]. The wrinkle surface can sufficiently scatter light and be read by an objective equipped smartphone (also called a smartphone microscope). Embedding luminescent materials in the wrinkles makes them detectable with fluorescence microscopes. Similar to human fingerprints, the

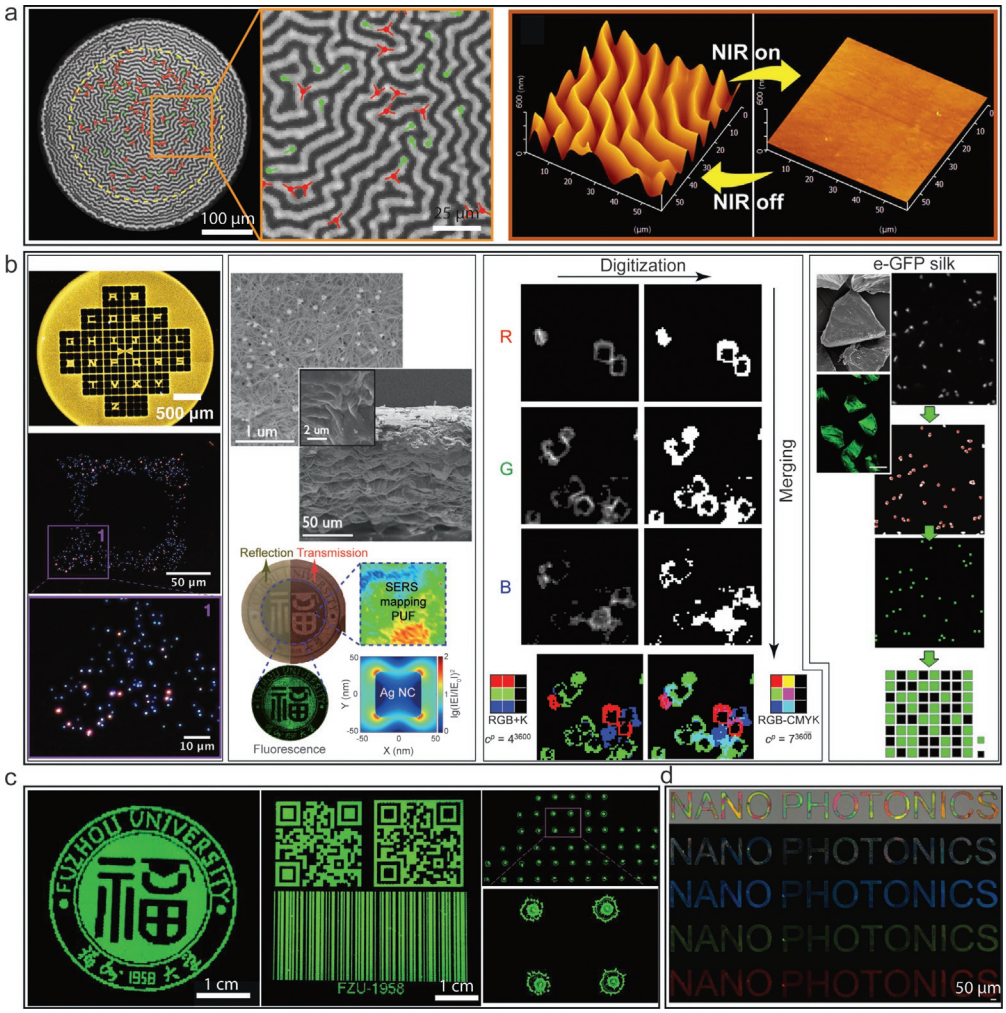


Fig. 7 Typical disordered physical features of PUF-based security labels. (A) Artificial fingerprints [144,145], Copyright 2018, Science, Copyright 2015, Wiley-VCH, (B) self-assembled stochastic particle arrays [7,12,13,146], Copyright 2016, Wiley-VCH, Copyright 2020, The Royal Society of Chemistry, Copyright 2018, AAAS, Copyright 2020, Springer Nature, (C) inkjet-printed random QD patterns [18], Copyright 2019, Springer Nature, and (D) optical interference films [143], Copyright 2019, The Royal Society of Chemistry.

artificial fingerprints are composed of random patterns of ridges and valleys (Fig. 7A, left), which ensure their unclonability. Their uniqueness is determined by the local ridge characteristics (i.e., the ridge ending and the ridge bifurcation, called minutiae points) and their relationships. By controlling the composition of polymer film, dynamic wrinkle patterns can be achieved. For instance, the group of Jiang reported NIR-light-responsive dynamic wrinkles based on a bilayer system that used functional polymers as the top stiff layers and carbon nanotube (CNT)/poly(dimethylsiloxane) (PDMS) elastomer as the bottom supporting substrate (Fig. 7A, right) [144]. Upon NIR light irradiation, the wrinkle surface was flattened by the thermal expansion of the elastic CNT-PDMS substrate, caused by the conversion of photons to thermal energy. Once the NIR light was turned off, the surface switched back to its wrinkle state. The group also showed that, with a copolymer containing pyridine (P4VP-nBA-S) and hydroxyl distyrylpyridine (DSP-OH) as the skin layer, the wrinkle surface became smooth under irradiation with visible light (450 nm) or acid treatment [148]. The dynamic regulation of the wrinkle can be used as an additional control to increase the level of security.

Another widely studied physical function is self-assembled stochastic nanoparticle arrays [15,20,146,149,150]. Plasmonic nanoparticles have shown an extraordinary capability to manipulate light by scattering it out to the far field [12,15]. This remarkable characteristic makes it possible to image a single nanoparticle with an optical microscope. Single-nanoparticle imaging by a dark-field microscope has been intensively used for their optical characterization in order to correlate their spectra with their size and shape [15,149,151]. The excitation of surface plasmons results in a strong near-field intensity in the close vicinity of a metal surface that is orders of magnitude higher than that of the incident field [13,152]. This enables the amplification of fluorescence and Raman signals emitted by a surface-confined probe molecule [12,152,153]. Taking advantage of the rich optical properties of plasmonic nanoparticles, we used fluorescein-doped silver-silica core–shell nanoparticles as information carriers to generate multiplex security labels [12]. The coding information can be read not only as color images by dark-field and fluorescence microscopes but also as a spectral fingerprint by a Raman spectrometer, thus increasing their security level. When the particle density within the security labels reaches single-particle level, they show a random pattern consisting of discrete nanoparticles (Fig. 7B, column 1). This feature of randomness nondeterministically encodes graphical information in the nanoparticle patterns, guaranteeing intrinsically unreplicable code outputs. We further extended this concept to a plasmonic nanopaper system, in which plasmonic silver nanocubes were embedded in cellulose nanofibers [13]. The nanopaper was encoded with SERS and fluorescence signals by immobilization of a benzenethiol Raman reporter onto the silver nanocubes and printing fluorescein molecules on the plasmonic nanopaper. The random location of silver nanocubes resulted in the random distribution of Raman intensity across the whole nanopaper. The obtained security label carried multiplexed optical signals including surface plasmon resonance, fluorescence,

and SERS. The localized surface plasmon resonance and fluorescence information is used as the first layer of security and can be easily verified by the naked eyes, while the SERS mapping with random Raman intensity distribution is used as the second layer of security and can be readily read out by Raman microscopy. Surface plasmon resonance and SERS have also been exploited as sole encoding signals for PUF-based security labels [13]. Luminescent particles are also considered as a promising information carrier for PUF encryption [154]. Carro-Temboury et al. demonstrated the use of lanthanide-doped zeolites in polymer films as taggants and developed multicolor fluorescent PUF encryption systems (Fig. 7B, column 3) [146]. The zeolites doped with three different lanthanide (III) ions emitted primary RGB colors by selective excitation of the individual zeolites, producing the three raw fluorescence images. The images were then digitized into three binary images. By merging the digitized images with 60×60 pixels through an RGB + K or RGB-CMYK approach, a final multi-response digital key with a maximal encoding capacity of 4^{3600} or 7^{3600} was generated. The authors demonstrated that the encoding capacity is dramatically enhanced using the multicolor taggants for PUF encryption and the high-encoding capacity enables unbreakable encryption. The fluorescence materials can be easily extended from inorganic materials to biomaterials. For example, Leem et al. reported protein-based edible PUFs using silk protein (i.e., fibroin) and fluorescent proteins as bio-taggants (Fig. 7B, column 4) [7]. Such edible PUFs shared the same encoding and decoding mechanism as the inorganic ones reported by Carro-Temboury and co-workers. The edible cryptographic primitives have great potential in pharmaceutical anti-counterfeiting.

Other physical functions including inkjet-printed random QD patterns and optical interference films have also been developed [18,143]. Recently, we reported inkjet printing of PUF-based luminescent QD security labels made of physically unclonable flower-like dot patterns using II–VI semiconductor core–shell QDs as information carriers (Fig. 7C) [18]. The surface modification of the print substrates with randomly distributed poly(methyl methacrylate) (PMMA) nanoparticles is the key step for the successful inkjet printing of the flower-like dot patterns. The PMMA nanoparticles served as stochastic pinning points for the QDs to dwell during the solvent evaporation of the ink droplets. This nondeterministic pattern formation process determines the individuality of the flower-like dot patterns. Just like the flowers in nature, every flower-like dot pattern is unique. Any 2D macroscale patterns composed of thousands of flower-like dots (e.g., the Fuzhou University logo, QR codes, and bar codes shown in Fig. 7C) can be generated using the inkjet-printing technique. Colorful PUF-based luminescent security labels can be achieved using the RGB emission core–shell QDs as inks. These covert security labels can be visualized by the naked eye when exposed to UV light, which offers a simple preliminary verification. The flower-like dot patterns within the security labels are used as the second layer of security and can be readily read out by a smartphone microscope with UV light excitation. Random colorful patterns are also commonly seen on

soap bubbles, originating from optical interference. Inspired by this phenomenon, we demonstrated an unclonable anti-counterfeiting system that comprises an optical interference film made of an uneven coordination polymer layer on a flat gold reflector (Fig. 7D) [143]. The random colorful pattern similar to that of soap bubbles from the uneven coordination polymer film is used as the physical function. The colorful PUF-based security labels were fabricated by fast crystallization of the coordination polymer—cupric bromide complex of pyrazine 1,4-dioxide—on lithographically defined gold micropatterns. The fast crystallization results in the formation of the coordination polymer with different particle sizes and irregular particle shapes, which allows the optical interference to occur. The gold layer is of vital importance for the creation of colors as it absorbs blue light but reflects most of the other visible light. The fabricated PUF-based security labels exhibit multiple high-contrast colors under bright/dark-field modes and a single color under fluorescence mode (Fig. 7D). The encryption with colors on a 50 mm square produces unique PUF codes with a maximal encoding capacity of 8^{2500} (the number 8 represents the distinguishable visible colors).

5. Artificial intelligence-based authentication

As discussed above, advanced optical anti-counterfeiting labels carry two layers of security—macroscopic patterns and nanoscale PUFs [155,156]. The former can be visually verified, while the latter requires advanced readout tools and algorithms for accurate authentication. Generally, a physical mark on the PUF-based security label is required to define the positions of all the pixels within the physical functions for their authentication [143]. Otherwise, imaging the rotated physical functions will lead to a completely different PUF code [146]. A typical authentication procedure of PUFs includes the registration of the physical key by the manufacturer, reading out the uncertain physical key by the users and then the comparison of the uncertain physical key with the registered ones in the database. There are two well-established strategies to authenticate PUF functions. One involves the conversion of the physical function into a digital key and point-by-point comparison of the digital keys with those in the database [7,157], while the other one relies on a similar character extraction, recognition, and comparison analysis to those used to authenticate real fingerprints [13,18,143]. The computer vision community has developed many algorithms for pattern recognition, enhancement, and identification. The point-by-point comparison gives rise to high-encoding capacity but suffers from being time consuming. The character extraction method can significantly reduce authentication times but decreases encoding capacity as only partial of pixels within the security label are used for verification. The verification accuracy of both strategies can be affected by the image quality of the security labels, which is determined by many factors, such as the imaging equipment, conditions, and personnel habits of users. Dusts on the pattern and poor image quality cause false positives of up to 20% using conventional algorithms [10].

To address this problem, we developed a deep learning (artificial intelligence) and computer vision technique to validate PUF-based security labels. Fig. 8A illustrates a typical authentication procedure through deep learning. Firstly, all the physical functions on the commercial products are readout by the manufacturer (Fig. 8A, step 1) and then sent

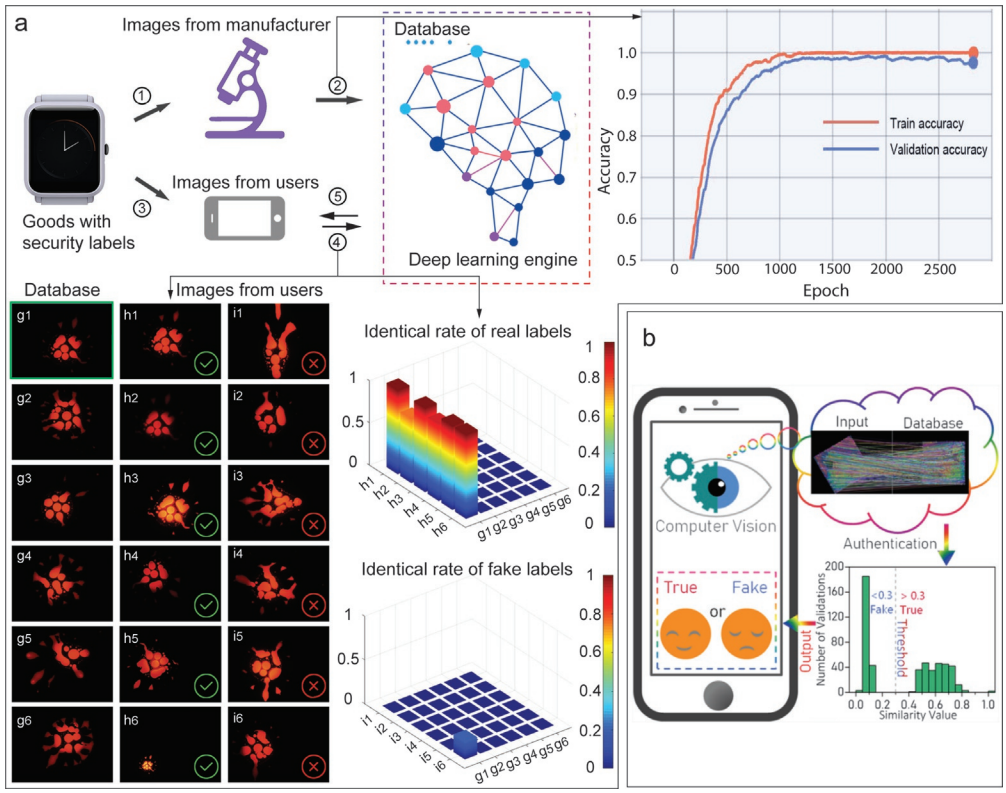


Fig. 8 (A) Deep learning decoding mechanism [18], Copyright 2019, Springer Nature. (*Top left*) Schematic illustrating the authentication process: (1) image capture and database generation from the goods by the manufacturers, (2) image learning by deep learning engine, (3) image capture from the goods by the consumers using their smart phones, (4) image recognition and comparing by deep learning engine, and (5) authentication outcome feedback to the consumers. (*Top right*) Image learning: Plots of train and validation accuracy as a function of the number of training epochs. (*Bottom left*) A library of six single dot pattern security labels (g_{1-6}), six fluorescence images taken from g_1 (referred to as genuine product) with different brightness, sharpness, rotation angles, magnifications, and the mixture of the above-mentioned factors (h_{1-6}), six fluorescence images from security labels that are not in the database (referred to as fake products) (i_{1-6}), and recognition rates of labels (h_{1-6}) and (i_{1-6}) by the authenticating way of deep learning. (B) A convenient and reliable authentication strategy based on the computer vision technique to verify the colorful PUF patterns with a different definition, rotation angle, brightness, magnification, and a mixture of these factors. A high correct validation rate of 100% is achieved using a match threshold of 0.3 [143], Copyright 2019, The Royal Society of Chemistry.

to neural networks for training (step 2). Note that only one image is taken from one security label and rotated to create training images. During the training phase, the neural networks learn the characteristic features of the security label. This is carried out by sending 80% of training images for learning and 20% for validation. The training is finished when the training and validation accuracy is above 97%. When the consumers receive the products, they image the physical functions with their personal devices (step 3), which are automatically sent to the neural networks for validation (step 4). The neural networks learn the characteristic features of the test physical functions and compare them with those stored in the database, producing a matching score. By comparing the matching scores with the threshold, the neural networks immediately provide the authentication results (real or fake) to the users (step 5). The selection of an appropriate threshold ensures low-false-positive and false-negative rates. Since the first report on using deep learning for PUF authentication [18], this approach has been adopted and further developed for the verification of various PUFs [79,158]. To omit the training process required for the deep learning technique, a computer vision-based authentication has been developed, in which only key characteristic features of the security labels are extracted and compared (Fig. 8B). Since the extracted feature vectors are invariant to location, scaling, and rotation, the computer vision-based verification technique allows us to accurately verify the PUF-based security labels with a different image scaling, focusing, and rotation in the absence of a physical mark and image training. Moreover, the approach is also robust to changes in illumination and noise on the security labels. The main drawback is that it requires large storage space to save the images of the security labels taken by the manufacturer. It appears that the integration of advanced artificial intelligence with computer vision techniques may lead to more powerful and robust authentication techniques.

6. Summary and outlook

We have described the optical encoding mechanisms, summarized the widely studied optical nanomaterials including their synthetic methods and optical properties, and highlighted the recent advances in optical anti-counterfeiting technologies. Firstly, the synthesis of various nanomaterials that can generate unique optical properties was presented. These nanomaterials can be broadly categorized as nanophotonic (including plasmonic) materials and luminescent materials. We discussed the construction of optical nanomaterials synthesized by wet-chemical methods into security labels with 2D patterns through self-assembly or inkjet printing. The direct fabrication and engineering of nanostructures via thin-film deposition, lithographic, and imprinting methods were also discussed. These nanostructures are able to create structural colors due to their optical absorption, scattering, diffraction, or interference. The structural colors can be dynamically tuned by controlling the size, shape, and composition of the components, lattice constant (or periodic distance), interparticle coupling distance and refractive index,

etc. Furthermore, examples of luminescent nanomaterials that produce fluorescence, TADF, upconversion and/or phosphorescence upon light excitation were presented. Secondly, two types of optical security labels have been developed using these nanomaterials as basic ingredients, which are referred to as conventional and PUF-based security labels. The former involves the use of collective behaviors of the optical nanomaterials for optical encryption. The main advantage of such security labels is that they can be easily verified by the naked eyes. However, the problem is that they are clonable due to their predicable and deterministic decoding mechanisms. In contrast, PUF-based security labels that make use of random nanoscopic physical features for optical encryption are difficult to be copied. Yet, they generally need an advanced and costly instrument to readout the encoded information. Lastly, a number of authentication techniques based on conventional image analysis, machine learning, deep learning or computer vision have been developed. Both deep learning and computer vision authentication approaches provide a solution to efficient and accurate authentication of PUF-based security labels with variations in sharpness, brightness, rotations, amplifications, and a mixture of these parameters. Compared with the other existing authentication algorithms, deep learning does not require any storage space for the security label images and is more efficient and robust.

Ideal anti-counterfeiting techniques require the security labels to be inexpensive, mass-producible, unclonable, and convenient for authentication with a simple and portable device. Optical PUF encryption is considered to be the next-generation anti-counterfeiting technique. However, there is still a long way toward their practical implementation. It is therefore necessary to improve existing PUF anti-counterfeiting techniques and develop new PUF techniques. To realize their practical application, future strategic directions are presented below: (1) the development of inexpensive, environmentally friendly (ideally edible) nanomaterials with rich optical properties that are dynamically tunable to increase the encoding levels, which allows for the development of dynamic and multiplex PUF security labels; (2) the development of patterning techniques that are able to integrate PUF-based security labels with the conventional ones into a single device while decreasing fabrication cost; and (3) the development of low-cost, miniaturized and nondestructive readout tools and advanced artificial intelligence authentication algorithms that ensure the simple, rapid, and accurate authentication of PUF-based security labels. It is believed that merging state-of-the-art nanotechnology and artificial intelligence techniques will be crucial to realize the commercialization of PUF-based security labels.

References

[1] P. Aldhous, Murder by medicine counterfeit, Nature 434 (2005) 132–136.
[2] Economics F, Estimating the global economic and social impacts of counterfeiting and piracy. A report commissioned by business action to stop counterfeiting and piracy (BASCAP), Suite A Lanham, An ICC Initiative, Institute for Creative Community Initiatives, 2011, pp. 20706.

[3] World Health Organization, IMPACT. Counterfeit Medicines: And Update on Estimates, WHO, 2006. http://www.who.int/medicines/services/counterfeit/impact/TheNewEstimatesCounterfeit.pdf.
[4] R.R. Gosline, http://forbes.com/2010/02/11/luxury-goodscounterfeit-fakes-chanel-gucci-cmo-network-renee-richardson-gosline.html, Forbes, 2010.
[5] J. Rogin, http://thedailybeast.com/articles/2014/02/12/the-u-s-tells-you-where-to-buy-the-best-fake-designer-handbags-pirated-movies-andmore.html., The Daily Beast, 2014.
[6] U.S. Food and Drug Administration, Counterfeit Medicine, U.S. Department of Health and Human Services, 2016, http://fda.gov/Drugs/ResourcesForYou/Consumers/BuyingUsingMedicineSafely/CounterfeitMedicine/default.htm., 2016.
[7] J.W. Leem, M.S. Kim, S.H. Choi, S.R. Kim, S.W. Kim, Y.M. Song, R.J. Young, Y.L. Kim, Edible unclonable functions, Nat. Commun. 11 (2020) 328.
[8] S. Wu, D. Ling, L. Jing, Application of Blockchain in Military Supply Chain Management, Logistics Technology, 2018.
[9] W. Ren, G. Lin, C. Clarke, J. Zhou, D. Jin, Optical nanomaterials and enabling technologies for high-security-level anticounterfeiting, Adv. Mater. 32 (2020), e1901430.
[10] R. Arppe, T.J. Sørensen, Physical unclonable functions generated through chemical methods for anti-counterfeiting, Nat. Rev. Chem. 1 (2017) 0031.
[11] Y.C. Chen, W.L. Wang, M.S. Hwang, RFID authentication protocol for anti-counterfeiting and privacy protection, in: International Conference on Advanced Communication Technology, 2007.
[12] Y. Zheng, C. Jiang, S.H. Ng, Y. Lu, F. Han, U. Bach, J.J. Gooding, Unclonable plasmonic security labels achieved by shadow-mask-lithography-assisted self-assembly, Adv. Mater. 28 (2016) 2330–2336.
[13] H. Cheng, Y. Lu, D. Zhu, L. Rosa, F. Han, M. Ma, W. Su, P.S. Francis, Y. Zheng, Plasmonic nanopapers: flexible, stable and sensitive multiplex PUF tags for unclonable anti-counterfeiting applications, Nanoscale 12 (2020) 9471–9480.
[14] Q. Zhu, S. Jiang, K. Ye, W. Hu, J. Zhang, X. Niu, Y. Lin, S. Chen, L. Song, Q. Zhang, J. Jiang, Y. Luo, Hydrogen-doping-induced metal-like ultrahigh free-carrier concentration in metal-oxide material for giant and tunable plasmon resonance, Adv. Mater. 32 (2020), e2004059.
[15] A.F. Smith, P. Patton, S.E. Skrabalak, Plasmonic nanoparticles as a physically unclonable function for responsive anti-counterfeit nanofingerprints, Adv. Funct. Mater. 26 (2016) 1315–1321.
[16] Y. Liu, C. Zhang, X. Li, D. Wu, A self-monitored fluorescence DNA anti-counterfeiting system based on silica coated SYBR green I/DNA gelatin nanoparticles, J. Mater. Chem. C 5 (2017) 5939–5948.
[17] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, Science 297 (2002) 2026–2030.
[18] Y. Liu, F. Han, F. Li, Y. Zhao, M. Chen, Z. Xu, X. Zheng, H. Hu, J. Yao, T. Guo, W. Lin, Y. Zheng, B. You, P. Liu, Y. Li, L. Qian, Inkjet-printed unclonable quantum dot fluorescent anti-counterfeiting labels with artificial intelligence authentication, Nat. Commun. 10 (2019) 2409.
[19] F. Kretschmer, S. Mühlig, S. Hoeppener, A. Winter, M.D. Hager, C. Rockstuhl, T. Pertsch, U.S. Schubert, Survey of plasmonic nanoparticles: from synthesis to application, Part. Part. Syst. Charact. 31 (2014) 721–744.
[20] A.F. Smith, S.E. Skrabalak, Metal nanomaterials for optical anti-counterfeit labels, J. Mater. Chem. C 5 (2017) 3207–3215.
[21] S. Wu, B. Liu, X. Su, S. Zhang, Structural color patterns on paper fabricated by inkjet printer and their application in anticounterfeiting, J. Phys. Chem. Lett. 8 (2017) 2835–2841.
[22] M. Campbell, N.D. Sharp, Fabrication of photonic crystals for the visible spectrum by holographic lithography (cover story), Nature 404 (2000) 53–56.
[23] Y. Liu, H. Wang, J. Ho, R.C. Ng, R.J.H. Ng, V.H. Hall-Chen, E.H.H. Koay, Z. Dong, H. Liu, C.-W. Qiu, J.R. Greer, J.K.W. Yang, Structural color three-dimensional printing by shrinking photonic crystals, Nat. Commun. 10 (2019) 4340.
[24] E.K. Davis, B.W. Russell, Disorder-to-order transition in settling suspensions of colloidal silica: X-ray measurements, Science 245 (1989) 507.

[25] H. Kim, J. Ge, J. Kim, S.-e. Choi, H. Lee, H. Lee, W. Park, Y. Yin, S. Kwon, Structural colour printing using a magnetically tunable and lithographically fixable photonic crystal, Nat. Photonics 3 (2009) 534–540.
[26] J. Ge, Y. Yin, Responsive photonic crystals, Angew. Chem. Int. Ed. 50 (2011) 1492–1522.
[27] W. Hong, Z. Yuan, X. Chen, Structural color materials for optical anticounterfeiting, Small 16 (2020), e1907626.
[28] Z. Xuan, J. Li, Q. Liu, F. Yi, S. Wang, W. Lu, Artificial structural colors and applications, Innovation 2 (2021) 100081.
[29] A. Kristensen, J. Yang, S.I. Bozhevolnyi, S. Link, P. Nordlander, N.J. Halas, N.A. Mortensen, Plasmonic colour generation, Nat. Rev. Mater. 2 (2016) 16088.
[30] W. Fan, J. Zeng, Q. Gan, D. Ji, L. Wu, Iridescence-controlled and flexibly tunable retroreflective structural color film for smart displays, Sci. Adv. 5 (2019), eaaw8755.
[31] J. Chen, Y. Wu, X. Li, F. Cao, Y. Gu, K. Liu, X. Liu, Y. Dong, J. Ji, H. Zeng, Simple and fast patterning process by laser direct writing for perovskite quantum dots, Adv. Mater. Technol. 2 (2017) 1700132.
[32] J. Li, Y. Chen, Y. Hu, H. Duan, N. Liu, Magnesium-based metasurfaces for dual-function switching between dynamic holography and dynamic color display, ACS Nano 14 (2020) 7892–7898.
[33] S.H. Ahn, L.J. Guo, Large-area roll-to-roll and roll-to-plate nanoimprint lithography: a step toward high-throughput application of continuous nanoimprinting, ACS Nano 3 (2009) 2304–2310.
[34] H. Jiang, B. Kaminska, Scalable inkjet-based structural color printing by molding transparent gratings on multilayer nanostructured surfaces, ACS Nano 12 (2018) 3112–3125.
[35] S.W. Wang, M. Li, C.S. Xia, H.Q. Wang, X.S. Chen, W. Lu, 128 channels of integrated filter array rapidly fabricated by using the combinatorial deposition technique, Appl. Phys. B Lasers Opt. 88 (2007) 281–284.
[36] H. Nam, K. Song, D. Ha, T. Kim, Inkjet printing based mono-layered photonic crystal patterning for anti-counterfeiting structural colors, Sci. Rep. 6 (2016) 30885.
[37] J.Y. Wang, Y. Cao, Y. Feng, F. Yin, J.P. Gao, Multiresponsive inverse-opal hydrogels, Adv. Mater. 19 (2007) 3865–3871.
[38] J. Yuan, A. Hazarika, Q. Zhao, X. Ling, T. Moot, W. Ma, J.M. Luther, Metal halide perovskites in quantum dot solar cells: progress and prospects, Joule 4 (2020) 1160–1185.
[39] D. Bera, L. Qian, T.-K. Tseng, P.H. Holloway, Quantum dots and their multimodal applications: a review, Materials 3 (2010) 2260–2345.
[40] D. Li, Q. Shao, Y. Dong, J. Jiang, Anomalous temperature-dependent upconversion luminescence of small-sized NaYF4:Yb3+, Er3+ nanoparticles, J. Phys. Chem. C 118 (2014) 22807–22813.
[41] Q. Liu, W. Feng, F. Li, Water-soluble lanthanide upconversion nanophosphors: synthesis and bioimaging applications in vivo, Coord. Chem. Rev. 273-274 (2014) 100–110.
[42] X. Zhu, J. Zhang, J. Liu, Y. Zhang, Recent progress of rare-earth doped upconversion nanoparticles: synthesis, optimization, and applications, Adv. Sci. 6 (2019) 1901358.
[43] Y. Jang, A. Shapiro, M. Isarov, A. Rubin-Brusilovski, A. Safran, A.K. Budniak, F. Horani, J. Dehnel, A. Sashchiuk, E. Lifshitz, Interface control of electronic and optical properties in IV-VI and II-VI core/shell colloidal quantum dots: a review, Chem. Commun. 53 (2017) 1002–1024.
[44] Z. Pang, J. Zhang, W. Cao, X. Kong, X. Peng, Partitioning surface ligands on nanocrystals for maximal solubility, Nat. Commun. 10 (2019) 2454.
[45] T.Y. Li, X. Xu, C.H. Lin, X. Guan, W.H. Hsu, M.L. Tsai, X. Fang, T. Wu, J.H. He, Highly UV resistant inch-scale hybrid perovskite quantum dot papers, Adv. Sci. 7 (2020) 1902439.
[46] X.F. Liu, L. Zou, C. Yang, W. Zhao, X.Y. Li, B. Sun, C.X. Hu, Y. Yu, Q. Wang, Q. Zhao, H.L. Zhang, Fluorescence lifetime-tunable water-resistant perovskite quantum dots for multidimensional encryption, ACS Appl. Mater. Interfaces 12 (2020) 43073–43082.
[47] D.E. Lee, S.Y. Kim, H.W. Jang, Lead-free all-inorganic halide perovskite quantum dots: review and outlook, J. Korean Ceram. Soc. 57 (2020) 455–479.
[48] Y. Wang, A. Hu, Carbon quantum dots: synthesis, properties and applications, J. Mater. Chem. C 2 (2014) 6921–6939.

[49] Q. Lou, S. Qu, P. Jing, W. Ji, D. Li, J. Cao, H. Zhang, L. Liu, J. Zhao, D. Shen, Water-triggered luminescent "nano-bombs" based on supra-(carbon nanodots), Adv. Mater. 27 (2015) 1389–1394.
[50] J. Zhou, S. Wen, J. Liao, C. Clarke, S.A. Tawfik, W. Ren, C. Mi, F. Wang, D. Jin, Activation of the surface dark-layer to enhance upconversion in a thermal field, Nat. Photonics 12 (2018) 154–158.
[51] W. Zheng, P. Huang, Z. Gong, D. Tu, J. Xu, Q. Zou, R. Li, W. You, J.G. Bunzli, X. Chen, Near-infrared-triggered photon upconversion tuning in all-inorganic cesium lead halide perovskite quantum dots, Nat. Commun. 9 (2018) 3462.
[52] H. Dong, L.D. Sun, W. Feng, Y. Gu, F. Li, C.H. Yan, Versatile spectral and lifetime multiplexing nanoplatform with excitation orthogonalized upconversion luminescence, ACS Nano 11 (2017) 3289–3297.
[53] Y. Sun, S. Liu, L. Sun, S. Wu, G. Hu, X. Pang, A.T. Smith, C. Hu, S. Zeng, W. Wang, Y. Liu, M. Zheng, Ultralong lifetime and efficient room temperature phosphorescent carbon dots through multi-confinement structure design, Nat. Commun. 11 (2020) 5591.
[54] L. Wang, M. Hasanzadeh Kafshgari, M. Meunier, Optical properties and applications of plasmonic-metal nanoparticles, Adv. Funct. Mater. 30 (2020) 2005400.
[55] X. Lu, M. Rycenga, S.E. Skrabalak, B. Wiley, Y. Xia, Chemical synthesis of novel plasmonic nanoparticles, Annu. Rev. Phys. Chem. 60 (2009) 167–192.
[56] X.M. Goh, Y. Zheng, S.J. Tan, L. Zhang, K. Kumar, C.W. Qiu, J.K. Yang, Three-dimensional plasmonic stereoscopic prints in full colour, Nat. Commun. 5 (2014) 5361.
[57] M. Faraday, The Bakerian Lecture: experimental relations of gold (and other metals) to light, Phil. Trans. R. Soc. London 147 (1857) 145–181.
[58] P. Yang, J. Zheng, Y. Xu, Q. Zhang, L. Jiang, Colloidal synthesis and applications of plasmonic metal nanoparticles, Adv. Mater. 28 (2016) 10508–10517.
[59] Q. Zhang, N. Li, J. Goebl, Z. Lu, Y. Yin, A systematic study of the synthesis of silver nanoplates: is citrate a "magic" reagent? J. Am. Chem. Soc. 133 (2011) 18931–18939.
[60] L. Chen, F. Ji, Y. Xu, L. He, Y. Mi, F. Bao, B. Sun, X. Zhang, Q. Zhang, High-yield seedless synthesis of triangular gold nanoplates through oxidative etching, Nano Lett. 14 (2014) 7201–7206.
[61] K.W. Hu, T.M. Liu, K.Y. Chung, K.S. Huang, C.T. Hsieh, C.K. Sun, C.S. Yeh, Efficient near-IR hyperthermia and intense nonlinear optical imaging contrast on the gold nanorod-in-shell nanostructures, J. Am. Chem. Soc. 131 (2009) 14186–14187.
[62] X. Ye, L. Jin, H. Caglayan, J. Chen, G. Xing, C. Zheng, V. Doan-Nguyen, Y. Kang, N. Engheta, C.-R. Kagan, Improved size-tunable synthesis of monodisperse gold nanorods through the use of aromatic additives, ACS Nano 6 (2012) 2804–2817.
[63] C. Ayala-Orozco, J.G. Liu, M.W. Knight, Y. Wang, J.K. Day, P. Nordlander, N.J. Halas, Fluorescence enhancement of molecules inside a gold nanomatryoshka, Nano Lett. 14 (2014) 2926–2933.
[64] A.K. Samal, L. Polavarapu, S. Rodal-Cedeira, L.M. Liz-Marzan, J. Perez-Juste, I. Pastoriza-Santos, Size tunable Au@Ag core-shell nanoparticles: synthesis and surface-enhanced Raman scattering properties, Langmuir 29 (2013) 15076–15082.
[65] Z. Feng, C. Jiang, Y. He, S. Chu, G. Chu, R. Peng, D. Li, Widely adjustable and quasi-reversible electrochromic device based on core-shell Au-Ag plasmonic nanoparticles, Adv. Opt. Mater. 2 (2014) 1174–1180.
[66] J. Song, B. Duan, C. Wang, J. Zhou, L. Pu, Z. Fang, P. Wang, T.T. Lim, H. Duan, SERS-encoded nanogapped plasmonic nanoparticles: growth of metallic nanoshell by templating redox-active polymer brushes, J. Am. Chem. Soc. 136 (2014) 6838–6841.
[67] L. Lin, H. Gu, J. Ye, Plasmonic multi-shell nanomatryoshka particles as highly tunable SERS tags with built-in reporters, Chem. Commun. 51 (2015) 17740–17743.
[68] K.J. Si, D. Sikdar, L.W. Yap, J.K.K. Foo, P. Guo, Q. Shi, M. Premaratne, W. Cheng, Dual-coded plasmene nanosheets as next-generation anticounterfeit security labels, Adv. Opt. Mater. 3 (2015) 1710–1717.
[69] L. Qin, M.J. Banholzer, J.E. Millstone, C. Mirkin, Nanodisk codes, Nano Lett. 7 (2007) 3849–3853.
[70] M.A. Kats, R. Blanchard, P. Genevet, F. Capasso, Nanometre optical coatings based on strong interference effects in highly absorbing media, Nat. Mater. 12 (2013) 20–24.
[71] J. Kim, J.M. Yun, J. Jung, H. Song, J.B. Kim, H. Ihee, Anti-counterfeit nanoscale fingerprints based on randomly distributed nanowires, Nanotechnology 25 (2014) 155303.

[72] K.-T. Lee, S. Seo, J. Yong Lee, L. Jay Guo, Ultrathin metal-semiconductor-metal resonator for angle invariant visible band transmission filters, Appl. Phys. Lett. 104 (2014), 231112.
[73] J. Xue, Z.K. Zhou, Z. Wei, R. Su, J. Lai, J. Li, C. Li, T. Zhang, X.H. Wang, Scalable, full-colour and controllable chromotropic plasmonic printing, Nat. Commun. 6 (2015) 8906.
[74] K.E. Chong, L. Wang, I. Staude, A.R. James, J. Dominguez, S. Liu, G.S. Subramania, M. Decker, D.-N. Neshev, I. Brener, Y.S. Kivshar, Efficient polarization-insensitive complex wavefront control using Huygens' metasurfaces based on dielectric resonant meta-atoms, ACS Photonics 3 (2016) 514–519.
[75] Z. Yang, Y. Zhou, Y. Chen, Y. Wang, P. Dai, Z. Zhang, H. Duan, Reflective color filters and monolithic color printing based on asymmetric fabry-Perot cavities using nickel as a broadband absorber, Adv. Opt. Mater. 4 (2016) 1196–1202.
[76] D. Franklin, R. Frank, S.T. Wu, D. Chanda, Actively addressed single pixel full-colour plasmonic display, Nat. Commun. 8 (2017) 15209.
[77] C. Sun, S. Su, Z. Gao, H. Liu, H. Wu, X. Shen, W. Bi, Stimuli-responsive inks based on perovskite quantum dots for advanced full-color information encryption and decryption, ACS Appl. Mater. Interfaces 11 (2019) 8210–8216.
[78] X. Zhang, R.F. Ali, J.C. Boyer, N.R. Branda, B.D. Gates, Direct photolithographic deposition of color-coded anti-counterfeit patterns with titania encapsulated upconverting nanoparticles, Adv. Opt. Mater. 8 (2020) 2000664.
[79] X. He, Y. Gu, B. Yu, Z. Liu, K. Zhu, N. Wu, X. Zhao, Y. Wei, J. Zhou, Y. Song, Multi-mode structural-color anti-counterfeiting labels based on physically unclonable amorphous photonic structures with convenient artificial intelligence authentication, J. Mater. Chem. C 7 (2019) 14069–14074.
[80] Y. Zhao, Y. Zhao, S. Hu, J. Lv, Y. Ying, G. Gervinskas, G. Si, Artificial structural color pixels: a review, Materials 10 (2017) 944.
[81] H. Tada, S. Mann, I. Miaoulis, P. Wong, Effects of a butterfly scale microstructure on the iridescent color observed at different angles, Opt. Express 5 (1999) 87–92.
[82] G. Wang, X. Chen, S. Liu, C. Wong, S. Chu, Mechanical chameleon through dynamic real-time plasmonic tuning, ACS Nano 10 (2016) 1788–1794.
[83] M. Srinivasarao, Nano-optics in the biological world:? Beetles, butterflies, birds, and moths, Chem. Rev. 99 (1999) 1935–1962.
[84] L. Cao, P. Fan, E.S. Barnard, A.M. Brown, M.L. Brongersma, Tuning the color of silicon nanostructures, Nano Lett. 10 (2010) 2649–2654.
[85] A.V. Yakovlev, V.A. Milichko, V.V. Vinogradov, A.V. Vinogradov, Inkjet color printing by interference nanostructures, ACS Nano 10 (2016) 3078–3086.
[86] A.E. Goodling, S. Nagelberg, B. Kaehr, C.H. Meredith, S.I. Cheon, A.P. Saunders, M. Kolle, L.D. Zarzar, Colouration by total internal reflection and interference at microscale concave interfaces, Nature 566 (2019) 523–527.
[87] K. Keller, A.V. Yakovlev, E.V. Grachova, A.V. Vinogradov, Inkjet printing of multicolor daylight visible opal holography, Adv. Funct. Mater. 28 (2018) 1706903.
[88] Y. Zhang, P. Han, H. Zhou, N. Wu, Y. Wei, X. Yao, J. Zhou, Y. Song, Highly brilliant noniridescent structural colors enabled by graphene Nanosheets containing graphene quantum dots, Adv. Funct. Mater. 28 (2018) 1802585.
[89] A.B. Evlyukhin, S.M. Novikov, U. Zywietz, R.L. Eriksen, C. Reinhardt, S.I. Bozhevolnyi, B.N. Chichkov, Demonstration of magnetic dipole resonances of dielectric nanospheres in the visible region, Nano Lett. 12 (2012) 3749–3755.
[90] A.V. Yakovlev, V.A. Milichko, V.V. Vinogradov, A.V. Vinogradov, Sol-gel assisted inkjet hologram patterning, Adv. Funct. Mater. 25 (2015) 7375–7380.
[91] M.A. Kats, F. Capasso, Optical absorbers based on strong interference in ultra-thin films, Laser Photonics Rev. 10 (2016) 735–749.
[92] J. Zhao, M. Qiu, X. Yu, X. Yang, W. Jin, D. Lei, Y. Yu, Defining deep-subwavelength-resolution, wide-color-gamut, and large-viewing-angle flexible subtractive colors with an ultrathin asymmetric Fabry-Perot Lossy cavity, Adv. Opt. Mater. 7 (2019) 1900646.
[93] C.I. Aguirre, E. Reguera, A. Stein, Tunable colors in opals and inverse opal photonic crystals, Adv. Funct. Mater. 20 (2010) 2565–2578.

[94] S. Andreas, F. Li, N.R. Denny, Morphological control in colloidal crystal templating of inverse opals, hierarchical structures, and shaped particles, Chem. Mater. 20 (2008) 649–666.
[95] L. Bai, V.C. Mai, Y. Lim, S. Hou, H. Mohwald, H. Duan, Large-scale noniridescent structural color printing enabled by infiltration-driven nonequilibrium colloidal assembly, Adv. Mater. 30 (2018) 1705667.
[96] M.M. Ito, A.H. Gibbons, D. Qin, D. Yamamoto, H. Jiang, D. Yamaguchi, K. Tanaka, E. Sivaniah, Structural colour using organized microfibrillation in glassy polymer films, Nature 570 (2019) 363–367.
[97] M. Deubel, G. von Freymann, M. Wegener, S. Pereira, K. Busch, C.M. Soukoulis, Direct laser writing of three-dimensional photonic-crystal templates for telecommunications, Nat. Mater. 3 (2004) 444–447.
[98] X. Ou, X. Qin, B. Huang, J. Zan, Q. Wu, Z. Hong, L. Xie, H. Bian, Z. Yi, X. Chen, Y. Wu, X. Song, J. Li, Q. Chen, H. Yang, X. Liu, High-resolution X-ray luminescence extension imaging, Nature 590 (2021) 410–415.
[99] Y. Lu, J. Zhao, R. Zhang, Y. Liu, D. Liu, E.M. Goldys, X. Yang, P. Xi, A. Sunna, J. Lu, Y. Shi, R.C. Leif, Y. Huo, J. Shen, J.A. Piper, J.P. Robinson, D. Jin, Tunable lifetime multiplexing using luminescent nanocrystals, Nat. Photonics 8 (2013) 32–36.
[100] S. Xie, G. Gong, Y. Song, H. Tan, C. Zhang, N. Li, Y. Zhang, L. Xu, J. Xu, J. Zheng, Design of novel lanthanide-doped core–shell nanocrystals with dual up-conversion and down-conversion luminescence for anti-counterfeiting printing, Dalton Trans. 48 (2019) 6971–6983.
[101] J. Liu, N. Wang, Y. Yu, Y. Yan, J. Yu, Carbon dots in zeolites: a new class of thermally activated delayed fluorescence materials with ultralong lifetimes, Sci. Adv. 3 (2017), e1603171.
[102] K. Jiang, L. Zhang, J. Lu, C. Xu, C. Cai, H. Lin, Triple-mode emission of carbon dots: applications for advanced anti-counterfeiting, Angew. Chem. Int. Ed. 55 (2016) 7231–7235.
[103] K. Jiang, Y. Wang, C. Cai, H. Lin, Conversion of carbon dots from fluorescence to ultralong room-temperature phosphorescence by heating for security applications, Adv. Mater. 30 (2018), e1800783.
[104] B. Zhou, L. Tao, Y. Chai, S.P. Lau, Q. Zhang, Y.H. Tsang, Constructing interfacial energy transfer for photon up- and down-conversion from lanthanides in a core-shell nanostructure, Angew. Chem. Int. Ed. 55 (2016) 12356–12360.
[105] W. Yao, Q. Tian, B. Tian, M. Li, H. Wang, P. Zeng, L. Liu, H. Zheng, W. Wu, Dual upconversion nanophotoswitch for security encoding, Sci. China Mater. 62 (2018) 368–378.
[106] R. Deng, F. Qin, R. Chen, W. Huang, M. Hong, X. Liu, Temporal full-colour tuning through non-steady-state upconversion, Nat. Nanotechnol. 10 (2015) 237–242.
[107] Q. Chen, J. Wu, X. Ou, B. Huang, J. Almutlaq, A.A. Zhumekenov, X. Guan, S. Han, L. Liang, Z. Yi, J. Li, X. Xie, Y. Wang, Y. Li, D. Fan, D.B.L. Teh, A.H. All, O.F. Mohammed, O.M. Bakr, T. Wu, M. Bettinelli, H. Yang, W. Huang, X. Liu, All-inorganic perovskite nanocrystal scintillators, Nature 561 (2018) 88–93.
[108] Z. Zeng, B. Huang, X. Wang, L. Lu, Q. Lu, M. Sun, T. Wu, T. Ma, J. Xu, Y. Xu, S. Wang, Y. Du, C.H. Yan, Multimodal luminescent Yb(3+)/Er(3+)/Bi(3+)-doped perovskite single crystals for X-ray detection and anti-counterfeiting, Adv. Mater. 32 (2020), e2004506.
[109] J. Liu, H. Rijckaert, M. Zeng, K. Haustraete, B. Laforce, L. Vincze, I. Van Driessche, A.M. Kaczmarek, R. Van Deun, Simultaneously excited downshifting/upconversion luminescence from lanthanide-doped core/shell fluoride nanoparticles for multimode anticounterfeiting, Adv. Funct. Mater. 28 (2018) 1707365.
[110] M. Haase, H. Schafer, Upconverting nanoparticles, Angew. Chem. Int. Ed. 50 (2011) 5808–5829.
[111] N.J. Johnson, S. He, S. Diao, E.M. Chan, H. Dai, A. Almutairi, Direct evidence for coupled surface and concentration quenching dynamics in lanthanide-doped nanocrystals, J. Am. Chem. Soc. 139 (2017) 3275–3282.
[112] X. Liu, Y. Wang, X. Li, Z. Yi, R. Deng, L. Liang, X. Xie, D.T.B. Loong, S. Song, D. Fan, A.H. All, H. Zhang, L. Huang, X. Liu, Binary temporal upconversion codes of Mn(2+)-activated nanoparticles for multilevel anti-counterfeiting, Nat. Commun. 8 (2017) 899.
[113] L. Lei, J. Xia, Y. Cheng, Y. Wang, G. Bai, H. Xia, S. Xu, Enhancing negative thermal quenching effect via low-valence doping in two-dimensional confined core–shell upconversion nanocrystals, J. Mater. Chem. C 6 (2018) 11587–11592.
[114] L. Lei, D. Chen, C. Li, F. Huang, J. Zhang, S. Xu, Inverse thermal quenching effect in lanthanide-doped upconversion nanocrystals for anti-counterfeiting, J. Mater. Chem. C 6 (2018) 5427–5433.

[115] F. Yuan, T. Yuan, L. Sui, Z. Wang, Z. Xi, Y. Li, X. Li, L. Fan, Z. Tan, A. Chen, M. Jin, S. Yang, Engineering triangular carbon quantum dots with unprecedented narrow bandwidth emission for multicolored LEDs, Nat. Commun. 9 (2018) 2249.
[116] Q. Shao, G. Zhang, L. Ouyang, Y. Hu, Y. Dong, J. Jiang, Emission color tuning of core/shell upconversion nanoparticles through modulation of laser power or temperature, Nanoscale 9 (2017) 12132–12141.
[117] A. Abdollahi, H. Roghani-Mamaqani, B. Razavi, M. Salami-Kalajahi, Photoluminescent and chromic nanomaterials for anticounterfeiting technologies: recent advances and future challenges, ACS Nano 14 (2020) 14417–14492.
[118] J. Wang, C.F. Wang, S. Chen, Amphiphilic egg-derived carbon dots: rapid plasma fabrication, pyrolysis process, and multicolor printing patterns, Angew. Chem. Int. Ed. 51 (2012) 9297–9301.
[119] S. Kalytchuk, Y. Wang, K. Polakova, R. Zboril, Carbon dot fluorescence-lifetime-encoded anticounterfeiting, ACS Appl. Mater. Interfaces 10 (2018) 29902–29908.
[120] W. Li, W. Zhou, Z. Zhou, H. Zhang, X. Zhang, J. Zhuang, Y. Liu, B. Lei, C. Hu, A universal strategy for activating the multicolor room-temperature afterglow of carbon dots in a boric acid matrix, Angew. Chem. Int. Ed. 58 (2019) 7278–7283.
[121] Y. Sun, J. Liu, X. Pang, X. Zhang, J. Zhuang, H. Zhang, C. Hu, M. Zheng, B. Lei, Y. Liu, Temperature-responsive conversion of thermally activated delayed fluorescence and room-temperature phosphorescence of carbon dots in silica, J. Mater. Chem. C 8 (2020) 5744–5751.
[122] K. Jiang, Y. Wang, X. Gao, C. Cai, H. Lin, Facile, quick, and gram-scale synthesis of ultralong-lifetime room-temperature-phosphorescent carbon dots by microwave irradiation, Angew. Chem. Int. Ed. 57 (2018) 6216–6220.
[123] J. Olson, A. Manjavacas, L. Liu, W.S. Chang, B. Foerster, N.S. King, M.W. Knight, P. Nordlander, N.J. Halas, S. Link, Vivid, full-color aluminum plasmonic pixels, Proc. Natl. Acad. Sci. U. S. A. 111 (2014) 14348–14353.
[124] Y. Cui, I.Y. Phang, Y.H. Lee, M.R. Lee, Q. Zhang, X.Y. Ling, Multiplex plasmonic anticounterfeiting security labels based on surface-enhanced Raman scattering, Chem. Commun. 51 (2015) 5363–5366.
[125] D. Franklin, Y. Chen, A. Vazquez-Guardado, S. Modak, J. Boroumand, D. Xu, S.T. Wu, D. Chanda, Polarization-independent actively tunable colour generation on imprinted plasmonic surfaces, Nat. Commun. 6 (2015) 7337.
[126] Z. Li, S. Butun, K. Aydin, Large-area, lithography-free super absorbers and color filters at visible frequencies using ultrathin metallic films, ACS Photonics 2 (2015) 183–188.
[127] C. Kuemin, L. Nowack, L. Bozano, N.D. Spencer, H. Wolf, Oriented assembly of gold nanorods on the single-particle level, Adv. Funct. Mater. 22 (2012) 702–708.
[128] K. Kumar, H. Duan, R.S. Hegde, S.C. Koh, J.N. Wei, J.K. Yang, Printing colour at the optical diffraction limit, Nat. Nanotechnol. 7 (2012) 557–561.
[129] T. Thai, Y. Zheng, S.H. Ng, S. Mudie, M. Altissimo, U. Bach, Self-assembly of vertically aligned gold nanorod arrays on patterned substrates, Angew. Chem. Int. Ed. 51 (2012) 8732–8735.
[130] P. Zijlstra, J.W. Chon, M. Gu, Five-dimensional optical recording mediated by surface plasmons in gold nanorods, Nature 459 (2009) 410–413.
[131] P. Mao, C. Liu, F. Song, M. Han, S.A. Maier, S. Zhang, Manipulating disordered plasmonic systems by external cavity with transition from broadband absorption to reconfigurable reflection, Nat. Commun. 11 (2020) 1538.
[132] L. Shao, X. Zhuo, J. Wang, Advanced plasmonic materials for dynamic color display, Adv. Mater. 30 (2018), e1704338.
[133] M.L. Tseng, J. Yang, M. Semmlinger, C. Zhang, P. Nordlander, N.J. Halas, Two-dimensional active tuning of an aluminum plasmonic array for full-spectrum response, Nano Lett. 17 (2017) 6034–6039.
[134] Y. Shen, V. Rinnerbauer, I. Wang, V. Stelmakh, J.D. Joannopoulos, M. Soljačić, Structural colors from Fano resonances, ACS Photonics 2 (2015) 27–32.
[135] H. Jiang, B. Kaminska, H. Porras, M. Raymond, T. Kapus, Microlens arrays above interlaced plasmonic pixels for optical security devices with high-resolution multicolor motion effects, Adv. Opt. Mater. 7 (2019) 1900237.
[136] A. Dhawan, M.D. Gerhold, J.F. Muth, Plasmonic structures based on subwavelength apertures for chemical and biological sensing applications, IEEE Sensors J. 8 (2008) 942–950.

[137] Y. Yu, L. Wen, S. Song, Q. Chen, Transmissive/reflective structural color filters: theory and applications, J. Nanomater. 2014 (2014) 1–17.
[138] Y. Cui, R.S. Hegde, I.Y. Phang, H.K. Lee, X.Y. Ling, Encoding molecular information in plasmonic nanostructures for anti-counterfeiting applications, Nanoscale 6 (2014) 282–288.
[139] L. Liu, R. Aleisa, Y. Zhang, J. Feng, Y. Yin, Dynamic color-switching of plasmonic nanoparticle films, Angew. Chem. Int. Ed. 58 (2019) 16307–16313.
[140] H. Cheng, W. Sun, Y. Lu, H. Li, W. Su, J. Zhang, T. Guo, F. Li, P.S. Francis, Y. Zheng, Hot electrons in carbon nitride with ultralong lifetime and their application in reversible dynamic color displays, Cell Rep. Phys. Sci. 2 (2021) 100516.
[141] R. Yu, P. Mazumder, N.F. Borrelli, A. Carrilero, D.S. Ghosh, R.A. Maniyara, D. Baker, F.J. García de Abajo, V. Pruneri, Structural coloring of glass using dewetted nanoparticles and ultrathin films of metals, ACS Photonics 3 (2016) 1194–1201.
[142] C.N. Chong, D. Jiang, J. Zhang, L. Guo, Anti-counterfeiting with a random pattern, in: 2008 Second International Conference on Emerging Security Information, Systems and Technologies, 2008, pp. 146–153.
[143] F. Han, Y. Liu, F. Li, Y. Lu, H. Cheng, Y. Lin, T. Zhao, S.H. Ng, U. Bach, Y. Zheng, Self-assembly of coordination polymers on plasmonic surfaces for computer vision decodable, unclonable and colorful security labels, J. Mater. Chem. C 7 (2019) 13040–13046.
[144] F. Li, H. Hou, J. Yin, X. Jiang, Near-infrared light-responsive dynamic wrinkle patterns, Sci. Adv. 4 (2018), eaar5762.
[145] H.J. Bae, S. Bae, C. Park, S. Han, J. Kim, L.N. Kim, K. Kim, S.H. Song, W. Park, S. Kwon, Biomimetic microfingerprints for anti-counterfeiting strategies, Adv. Mater. 27 (2015) 2083–2089.
[146] M.R. Carro-Temboury, R. Arppe, T. Vosch, T.J. Srensen, An optical authentication system based on imaging of excitation-selected lanthanide luminescence, Sci. Adv. 4 (2018), e1701384.
[147] G. Chen, Y. Weng, W. Wang, D. Hong, L. Zhou, X. Zhou, C. Wu, Y. Zhang, Q. Yan, J. Yao, T. Guo, Spontaneous formation of random wrinkles by atomic layer infiltration for anticounterfeiting, ACS Appl. Mater. Interfaces 13 (2021) 27548–27556.
[148] T. Ma, T. Li, L. Zhou, X. Ma, J. Yin, X. Jiang, Dynamic wrinkling pattern exhibiting tunable fluorescence for anticounterfeiting applications, Nat. Commun. 11 (2020) 1811.
[149] J.D. Smith, M.A. Reza, N.L. Smith, J. Gu, M. Ibrar, D.J. Crandall, S.E. Skrabalak, Plasmonic anticounterfeit tags with high encoding capacity rapidly authenticated with deep machine learning, ACS Nano 15 (2021) 2901–2910.
[150] R. Arppe-Tabbara, M. Tabbara, T.J. Sorensen, Versatile and validated optical authentication system based on physical unclonable functions, ACS Appl. Mater. Interfaces 11 (2019) 6475–6482.
[151] Q. Li, F. Chen, J. Kang, J. Su, F. Huang, P. Wang, X. Yang, Y. Hou, Physical unclonable anticounterfeiting electrodes enabled by spontaneously formed plasmonic core–shell nanoparticles for traceable electronics, Adv. Funct. Mater. 31 (2021) 2010537.
[152] Y. Zheng, A.H. Soeriyadi, L. Rosa, S.H. Ng, U. Bach, J. Justin Gooding, Reversible gating of smart plasmonic molecular traps using thermoresponsive polymers for single-molecule detection, Nat. Commun. 6 (2015) 8797.
[153] X. Zhang, Y. Zheng, X. Liu, W. Lu, J. Dai, D.Y. Lei, D.R. MacFarlane, Hierarchical porous plasmonic metamaterials for reproducible ultrasensitive surface-enhanced Raman spectroscopy, Adv. Mater. 27 (2015) 1090–1096.
[154] P. Kumar, S. Singh, B.K. Gupta, Future prospects of luminescent nanomaterial based security inks: from synthesis to anti-counterfeiting applications, Nanoscale 8 (2016) 14297–14340.
[155] J. Sang, J. Zhou, J. Zhang, H. Zhou, H. Li, Z. Ci, S. Peng, Z. Wang, Multilevel static-dynamic anticounterfeiting based on stimuli-responsive luminescence in a niobate structure, ACS Appl. Mater. Interfaces 11 (2019) 20150–20156.
[156] L. Shi, L. Meng, F. Jiang, Y. Ge, F. Li, X.G. Wu, H. Zhong, In situ inkjet printing strategy for fabricating perovskite quantum dot patterns, Adv. Funct. Mater. 29 (2019), 1903648.
[157] Y. Gu, C. He, Y. Zhang, L. Lin, B.D. Thackray, J. Ye, Gap-enhanced Raman tags for physically unclonable anticounterfeiting labels, Nat. Commun. 11 (2020) 516.
[158] W. Liang, S. Xie, D. Zhang, X. Li, K.-c. Li, A mutual security authentication method for RFID-PUF circuit based on deep learning, ACM Trans. Internet Technol. 22 (2022) 1–20.

CHAPTER 14

Machine learning data processing as a bridge between microscopy and the brain

Yijun Bao[a] **and Yiyang Gong**[a,b]
[a]Department of Biomedical Engineering, Duke University, Durham, NC, United States
[b]Department of Neurobiology, Duke University, Durham, NC, United States

1. Introduction

A brain has many neurons that together can convey complex emotions and behaviors. Neuroscientists want to understand the activities of neurons, including the relationship between the activity of neurons and animal behavior. Electrophysiology has been a useful tool to quantify neural activities associated with the dynamic neuron membrane potential, especially the action potential, or "spike," voltage waveform that acts as the fundamental unit of communication between neurons. However, electrophysiology with cellular resolution typically requires invasively inserting an electrode into the brain; the technique can only sample one or more neurons simultaneously spread over large volumes of tissues. Optical microscopy can image many neurons in a large field of view simultaneously, chronically, and in vivo. Genetically encoded calcium indicators (GECI) [1–4] and genetically encoded voltage indicators (GEVI) [5–8] were two types of genetically encoded tools that can translate changes in intracellular calcium concentration or membrane voltage associated with action potentials into changes in fluorescence intensity. Fluorescence microscopy can image these indicators by exciting them at one wavelength and capturing emission photons at a different wavelength. Generally, the classes of GECIs and GEVIs have trended toward larger responses and higher brightness, and have enabled more sensitive recordings of neural activity limited by shot noise. Depending on the wavelength of the excitation light, the indicators needed to absorb one or more photons to be excited, requiring the one-photon, or multiphoton microscopy [9]. In parallel with sensor development, microscopy development has improved optical sectioning, capture efficiency, and resolutions. Combining genetically encoded indicators and advanced microscopy techniques, neuroscientists can now record the neural activities of hundreds to thousands of neurons in multiple animal models at high speed.

The fluorescence imaging data, although very informative, was not at cellular resolution in raw form as with electrophysiological data. Neuroscientists currently perform a

Intelligent Nanotechnology
https://doi.org/10.1016/B978-0-323-85796-3.00014-7
Copyright © 2023 Elsevier Inc.
All rights reserved.

multistep video processing pipeline to extract the neural activities from individual neurons (Fig. 1). The pipeline includes correcting motion artifacts, segmenting spatial footprints of neurons, extracting temporal traces, and inferring activity spikes [10–12]: First, motion artifacts caused movement of animals during in vivo experiments should be corrected so that the imaged neurons in different frames are aligned; second, active neurons with dynamic activity should be segmented; third, for each neuron, the temporal activity trace should be calculated; and finally, the spikes of that neuron can be inferred from the activity trace. In the past, many neuroscientists manually drew masks for the neurons and automated the remaining steps. However, advances in imaging techniques have brought large increases in the recording size (from several GB to several hundreds of GB); ambitious neuroscience experiments sometimes measure neural activity from multiple planes in a brain simultaneously, over multiple animals and multiple days. All these factors make

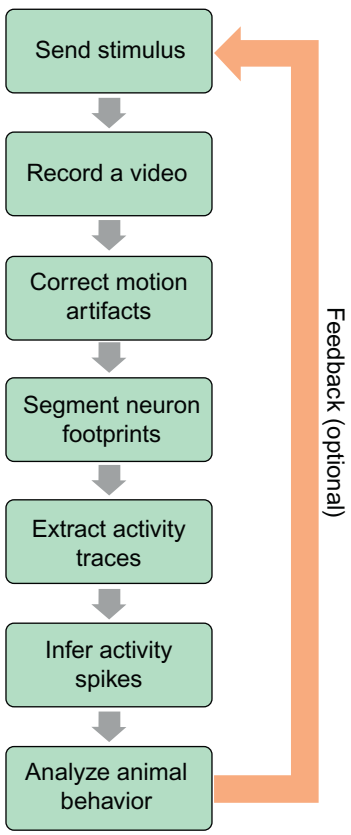


Fig. 1 Typical flow chart of data processing steps in neuroscience experiments using genetically encoded indicators and fluorescence microscopy. The optional orange feedback loop is added in feedback closed-loop experiments.

manual neuron segmentation very time-consuming. In addition, manual segmentation lacks consistency; multiple labelers or even the same labeler can mark the same neuron differently over multiple instances. Automated neuron segmentation techniques can potentially offer better consistency, scalability, and speed, and thus can advance neuroscience research based on fluorescent imaging microscopy. Researchers have developed many automated neuron segmentation algorithms based on machine learning techniques. The accuracy and speed of these algorithms have continually improved over time, and the state-of-the-art algorithms now approach the accuracy of human labelers but significantly outperform humans in speed [10,13,14]. Some neuroscience experiments seek to perform closed-loop, real-time behavioral or neural manipulation in response to the recorded neural activity [15,16] (Fig. 1). These real-time investigations require an online video processing pipeline, which adds the following requirements: the processing speed must be faster than the recording speed of the microscope, and the processing results must be updated shortly after the image collection without waiting for future frames. Researchers have also developed some online methods using machine learning, although these methods generally lag the performance of batch algorithms that process large portions of videos together.

In this chapter, we will give an overview of existing neuron segmentation methods based on machine learning, focusing mostly on processing calcium imaging data from two-photon microscopy. Because current GECIs have a better signal-to-noise ratio (SNR) than GEVIs, and current two-photon microscopy has better SNR and optical sectioning than one-photon microscopy in scattering tissues, many data processing methods targeted GECI and two-photon microscopy data first, with some of them later generalizing to GEVI or one-photon microscopy. We will also briefly discuss spike inference after neuron segmentation.

2. Machine learning

Machine learning is the study of computer algorithms that improves automatically through experience and by the use of data [17]. It is a branch of artificial intelligence that can learn from data, identify informative patterns, and make decisions with minimal human intervention. This is particularly helpful when there is no deterministic step-by-step guide for the desired tasks. The tasks of identifying neurons and neural spikes from the imaging videos fall into the class of feature extraction and pattern recognition problems. Many solutions to this class of problems come from two major categories of machine learning techniques in image and video processing: unsupervised learning and supervised learning. Unsupervised learning methods learn from unlabeled data by extracting meaningful features. They do not require human intervention in principle. Supervised learning methods learn to predict the output from labeled data, which includes the desired outputs of the inputs. These labels are prepared using other means, often with human labeling. The

segmentation algorithms are trained using pairs of raw input data and output labels, with the goal of finding general rules that map the raw input data to output labels.

Widely used unsupervised learning methods in neuron segmentation include clustering, dimensionality reduction, and dictionary learning. Clustering sought to group the pixels in the video according to their similarity (Fig. 2A). Graphically, a clustering algorithm defined a distance between any two pixels and sought to divide the pixels into several groups, so that any two pixels belonging to the same group had a small distance (and were thus similar), and any two pixels not belonging to the same group had a large distance (and were thus different) [18]. In neuron segmentation, the similarity was often related to the temporally synchronized activity [19,20]. Dimensionality reduction sought to decompose the calcium imaging video into a product of two matrices, and the decomposed matrices had much smaller dimensions and ranks than the original video [21]. The most widely used decomposition was spatial and temporal decomposition, where one decomposed matrix represented the spatial footprints of neurons and the other represented the temporal activities of neurons [12,22–24] (Fig. 2B). Matrix factorization algorithms often added constraints or penalties with the decomposition, so that the factorized matrices had some desired features present in neurons and calcium dynamics. Dictionary learning, also known as sparse coding, sought to find a finite number of basis functions and used a sparse linear combination of the basis functions to represent the video. In neuron segmentation, the basis functions were often the spatial footprints of individual neurons [25–27], temporal activities of individual neurons [28], or a combination of the spatial and temporal components [29].

Supervised learning often built artificial neural networks that originated from mimicking how the neurons in animal brains process signals. The constructed neural networks often had multiple layers, each containing several artificial neurons and performing a specific operation. The most widely used neural network in image and video processing was the convolutional neural network (CNN), which applied convolution layers to integrate the information from groups of pixels across the entire image (Fig. 2C). There were often multiple pooling layers that downscaled the data, so that higher-level features can be computed and learned. If the output was just a single number that classified the input, a fully connect layer was usually used as the final layer of the neural network to produce a single output indicator. These CNNs were known as classifiers. If the output value was binary, they can classify whether the neuron candidates were truly neurons based on their spatial properties [11,30]. When the allowed output had a finite number of options, they can represent the type of cells [31]. If the desired output should have the same size as the input, up-converting layers restored high-level features back to the original dimensionality after pooling. The outputs of this type of CNN were feature maps that corresponded to the input pixel by pixel. One traditional output is the probability map [14], which inferred the probability of each pixel being active. However, the output layer can also represent other quantities, like spatial gradient [32] or pixel-pixel affinity [19]. The

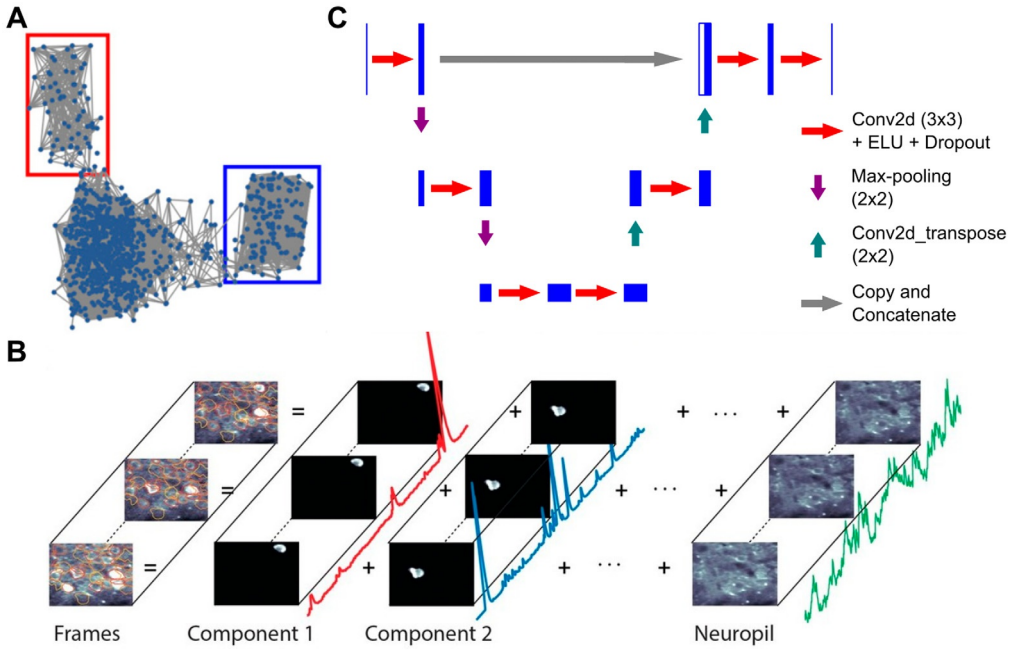


Fig. 2 Typical categories of machine learning algorithms used in neuron segmentation. *Multiple sources* (A) Clustering. This type of method constructed a graph, in which each node represented a pixel and the length of the edge between each pair of pixels represented the dissimilarity between the pair of pixels. A cluster of nodes with short distances can represent a neuron. (B) Matrix factorization. A video is modeled as the linear superposition of the contributions from various neural components, background (neuropil), and random noise; the contribution of each neuron was the tensor product of its spatial footprint and its temporal activity. Matrix factorization methods seek to find the best matrices that represent the spatial and temporal components of the neurons using some iterative optimization algorithms, so that the reconstructed video is as close to the original video as possible subject to some constraints and regularizations. (C) A convolutional neural network (CNN). A typical CNN has convolution, max-pooling, convolution transpose (up-conversion), concatenation, and dropout layers. These tensor operations can generate features from various resolution levels to help segment neurons. The output image has the same dimension as the input image. *((A) From Q. Spaen, R. Asín-Achá, S.N. Chettih, M. Minderer, C. Harvey, D.S. Hochbaum, HNCcorr: a novel combinatorial approach for cell identification in calcium-imaging movies, Eneuro 6(2) (2019), https://doi.org/10.1523/eneuro.0304-18.2019, Copyright © 2019. (B) From A. Giovannucci, J. Friedrich, P. Gunn, J. Kalfon, B.L. Brown, S.A. Koay, J. Taxidis, F. Najafi, J.L. Gauthier, P. Zhou, B.S. Khakh, D.W. Tank, D.B. Chklovskii, E.A. Pnevmatikakis, elife 8 (2019), https://doi.org/10.7554/eLife.38173, Copyright © 2019. (C) Modified from Y. Bao, S. Soltanian-Zadeh, S. Farsiu, Y. Gong, Segmentation of neurons from fluorescence calcium recordings beyond real-time, Nat. Mach. Intell. 3(7) (2021) 590–600, https://doi.org/10.1038/s42256-021-00342-x, Copyright © 2021 Springer International Publishing.)*

outputs of this type of CNN often required additional postprocessing to finally generate the segmented neuron masks. U-Net [33] was a popular CNN architecture widely used in image segmentation. In addition to convolution, pooling, and up-converting layers, it also used concatenations to connect two ends of the overall network, so that both low-level and high-level information can help generate the output. Many neuron segmentation methods used either the original U-Net [31] or the modified forms of U-Net [10,19,32,34]. In addition to CNNs, other supervised learning techniques, such as the recurrent neural network (RNN), have been applied in neuron segmentation as well [35].

Some algorithms combined the benefits of unsupervised and supervised learning techniques. For example, hybrid algorithms employed unsupervised learning to extract features (e.g., calcium sources) from the data, and employed supervised learning to evaluate the features and decided whether they are true features of interest (e.g., neurons) [11,30]. Alternatively, unsupervised learning algorithms can postprocess the outputs of CNNs, often simpler quantities than the raw data, to generate final neuron masks [19]. In addition, semisupervised learning trained algorithms in regimes between unsupervised and supervised learning, often when there was not sufficient training data [36]. These methods trained models from a dataset containing both labeled and unlabeled data; training models on the small, labeled dataset initialized the models, and guided classification and feature extraction on the large, unlabeled dataset.

Machine learning problems are closely tied with optimization problems, because the learning processes often minimize some cost functions or loss functions. In supervised learning, these functions quantify the difference between the training labels and the outputs from processing the training data. Minimizing these functions leads to a map between future test data input and algorithm output. In unsupervised learning, computers try to extract informative features from the input and reconstruct the input using the extracted features. The goal is to make the reconstructed input closely match the real input under some constraints, and their differences are usually quantified using cost functions. In practice, the cost functions often contain regularization terms to reflect the preferences or constraints. Popular optimization algorithms, such as gradient descent, stochastic gradient descent, and alternating gradient descent, are often employed in machine learning. But different from conventional optimization problems, the machine learning problems aim to minimize the equivalent loss or cost functions in data not used in training, so the trained models are generalizable and do not overfit the training data.

3. Identifying active neurons

Most GECIs report active neurons with increases in fluorescent intensity. Neurons present their footprints most clearly during their active frames, so that active neurons appear brighter than inactive neurons and background in neighboring regions. Neurons often

have near oval shapes, and the boundary pixels are often brighter than the interior pixels, because many GECIs are mainly expressed in the cell cytoplasm rather than the nucleus. The intensity of the neurons varies over time, usually in the form of a nearly constant baseline and some sparse active transients. The transients typically follow a rapid rise and slow decay waveform, and the time constants of different calcium sensors have been measured in the literature. The pixels belonging to the same neuron usually fire concurrently, so these pixels have high correlations with each other. On the other hand, background pixels dominated by noise have low correlations to each other even if they are spatially connected.

Many machine learning techniques have been developed and applied to the problem of segmenting active neurons from calcium imaging videos. Based on the dimensionality of the processed data, these methods can be categorized into three major types: segmentation from 2D summary images, segmentation from 3D videos as a whole or in blocks, and segmentation from 3D videos frame-by-frame. Based on machine learning techniques, these methods can be characterized into three major types: unsupervised learning, supervised learning, and a combination of both. Different methods have different speeds and accuracy (Fig. 3). We will categorize algorithms primarily by their dimensionality and secondarily by their core machine learning techniques.

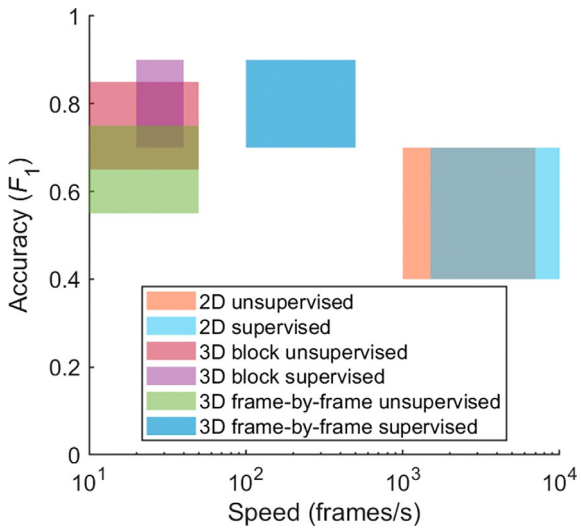


Fig. 3 Summary of accuracy and speed of various neuron segmentation technique categories. The data are mainly from Refs. [10,14]. The accuracy was quantified using the F_1 score defined in the referenced papers. The speed was quantified as the number of frames of a video divided by the total processing time. We assumed the two-photon imaging video had a relatively good quality, the lateral size was $\sim 512 \times 512$, and the computer was a standard desktop. Because the evaluation data in different papers were different, and some papers did not report quantitative speed, we filled this figure together with our best estimate to aid in qualitative comparisons.

3.1 2D segmentation methods

The first class of neuron segmentation algorithms processed small sets of 2D summary images. One 2D summary image can compress the temporal information of a 3D movie down to one quantity per pixel. Widely used summary quantities include the mean [34], maximum [37], correlation [20], or standard deviation [38]. Typically, active neurons should show higher strength in the summary images than background pixels, and image segmentation techniques can segment neurons based on these summary quantities. Particularly, the correlation image calculated the correlation of the temporal traces of each pixel to the neighboring pixels. If a pixel belonged to an active neuron, it should have high correlations with neighboring pixels, whereas if a pixel did not belong to a neuron, it should contain mostly noise and thus have low correlations with neighboring pixels. Segmenting 2D summary images was conceptually simpler than segmenting 3D video, and some methods developed for general cell segmentation can be applied directly [32].

One of the simplest 2D segmentation methods was applying a threshold on the summary image to identify active pixels and grouping active pixels into neurons. Some methods applied spatial filtering to enhance neuron shapes (CALIMA [39]). When neurons were slightly overlapped, watershed (SeNeCA [40]) or adaptive thresholding (ACSAT [37]) can separate some of them. Unsupervised learning algorithms can learn the spatial features of neurons or explore the similarity among pixels in multiple dimensions, so they can potentially identify fundamental neuron features or cluster pixels more accurately. Convolutional sparse block coding (CSBC [26]) is a dictionary learning method that modeled the mean image using convolutional sparse block coding, which inferred the most likely locations of neurons using convolutional block pursuit and learned the shapes of neurons using block K-SVD. HNCcorr [20] used Hochbaum's normalized cut to cluster pixels according to their similarities between correlation images. Local selective spectral clustering (LSSC [18]) represented the video using a 2D image of a special embedding norm and used spectral clustering to cluster pixels in the embedding space to neurons. NeuroSeg [41] processed the normalized image convolved with a generalized Laplacian of Gaussian filter by combining adaptive thresholding and clustering using agglomerative hierarchical clustering (Fig. 4A). Supervised learning algorithms were mainly based on CNNs, which can learn the spatial features of neurons after training on manually labeled examples. U-Net [33] was a widely used CNN in biomedical image processing tasks, and it has been used to directly segment the 2D summary images of fluorescent imaging videos (e.g., UNet2DS [34] and Seong et al. [31]). Cellpose [32] was a general cell segmentation technique not specific to neurons, which used a modified U-Net to predict the horizontal and vertical gradients as well as whether a pixel belonged to any cell. The three predicted maps were combined into a gradient vector field, and all the pixels that converged to the same fixed point were assigned to the same neuron mask. Some algorithms combined unsupervised and supervised learning. For example, DISCo

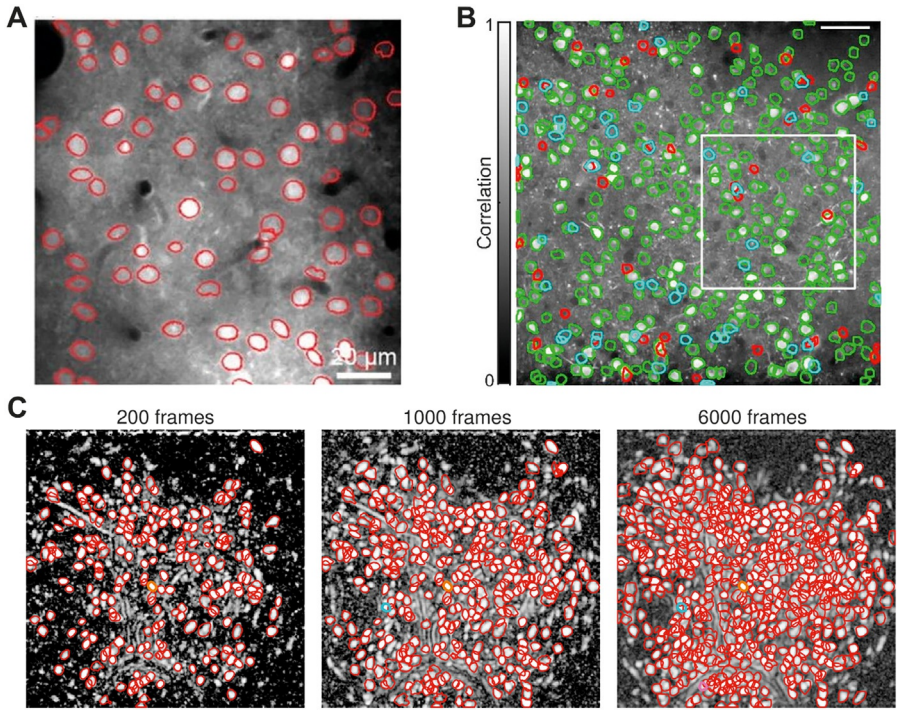


Fig. 4 Example neuron segmentation results using machine learning algorithms. *Multiple sources* (A) Result of NeuroSeg, a 2D unsupervised learning segmentation algorithm. Contours of segmented neurons are overlaid on the 2D image. No overlapping neurons were allowed. (B) Result of STNeuroNet, a 3D block supervised learning segmentation algorithm. Contours of true positive *(green)*, false positive *(red)*, false negative *(cyan)* neurons are overlaid on the correlation image. The accuracy of the segmented neurons was evaluated by comparing them with independent manual markings. Overlapping neurons were allowed in the output. (C) Result of OnACID-E, a 3D frame-by-frame unsupervised learning algorithm. Contours of segmented neurons up to a specific number of frames are overlaid on the correlation image. More neurons were segmented with increasing numbers of frames. *((A) From J. Guan, J. Li, S. Liang, R. Li, X. Li, X. Shi, C. Huang, J. Zhang, J. Pan, H. Jia, L. Zhang, X. Chen, X. Liao, NeuroSeg: automated cell detection and segmentation for in vivo two-photon Ca2+ imaging data, Brain Struct. Funct. 223(1) (2018) 519–533, https://doi.org/10.1007/s00429-017-1545-5, Copyright © 2021 Springer International Publishing. (B) From S. Soltanian-Zadeh, K. Sahingur, S. Blau, Y. Gong, S. Farsiu, Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning, Proc. Natl. Acad. Sci. 116(17) (2019) 8554–8563, https://doi.org/10.1073/pnas.1812995116, Copyright © 2019 National Academy of Sciences. (C) From J. Friedrich, A. Giovannucci, E.A. Pnevmatikakis, Online analysis of microendoscopic 1-photon calcium imaging data streams, PLoS Comput. Biol. 17(1) (2021) e1008565, https://doi.org/10.1371/journal.pcbi.1008565, Copyright: © 2021.)*

[19] applied a 3D CNN to predict the correlation image, and applied a U-Net on the concatenated mean and correlation images to predict pixel-pixel affinities and foreground-background probability; it then used graph cut to postprocess the U-Net output and clustered pixels to neurons. The multiple output layers of the U-Net enabled more sophisticated postprocessing that made use of both the activity probability of each pixel and the relationship among neighboring pixels for more accurate segmentation. ImageCN [30] detected regions of interest (ROIs) using a local adaptive thresholding algorithm, separated overlapping neurons using watershed, and finally classified ROIs as neurons or nonneurons using a CNN.

Overall, the class of algorithms processing 2D summary images was relatively fast but less accurate. 2D methods were usually very fast because of two reasons: calculating the summary images was fast, and the time processing the summary images was nearly independent of the video length. However, these algorithms struggled with separating overlapping neurons because overlapping neurons were usually connected in the summary images [14,20,34,42]; these connected regions were often incorrectly identified as a single neuron. Even when algorithms correctly identified the connected regions as containing multiple neurons, the algorithms failed to accurately mark the boundary between the neurons that was blurred in the summary image. ACSAT [37] recursively lowered the threshold to separate potentially overlapping neurons, but it could only separate overlapping neurons when the overlapping pixels were less bright than the unique pixels. SeNeCA [40] used an appropriately interrupted watershed algorithm to separate overlapping neurons, but the watershed may not preserve the natural shapes of neurons. UNet2DS [34] and Cellpose [32] removed the pixels belonging to multiple neurons when creating training labels, so that the trained CNN only learned features of nonoverlapping pixels. 2D methods can find both active and inactive neurons, because inactive neurons can also have a higher mean or maximum intensity than background pixels.

3.2 3D block segmentation methods

The second class of neuron segmentation algorithms processed the entire video or blocks of frames, and simultaneously analyzed both spatial features of neurons and temporal dynamics of neural activity. Because 3D methods extracted neurons based on their calcium dynamics, they usually only extracted active neurons. Unsupervised learning methods modeled the imaging video as independent contributions from each neuron, background, and noise, and the contribution of each neuron can be further decomposed into the spatial neuron shape and temporal dynamics (Fig. 2B). The models can be summarized as follows:

$$Y(x,t) = \sum\nolimits_i a_i(x)c_i(t) + B(x,t) + E(x,t) \quad (1)$$

where Y is the entire 3D video, a_i is the spatial component of each neuron, c_i is the temporal component of each neuron, B is the background, E is the noise, x is the 2D spatial coordinate, and t is the time. For the convenience of computation, the models were often expressed in a matrix form,

$$\boldsymbol{Y}^{(N\times T)} = \boldsymbol{A}^{(N\times K)}\,\boldsymbol{C}^{(K\times T)} + \boldsymbol{B}^{(N\times T)} + \boldsymbol{E}^{(N\times T)} \tag{2}$$

where $\mathbf{Y}$ is the entire video reshaped to a 2D matrix, $\mathbf{A}$ is the spatial components of all K neurons in the N pixels of each image, and $\mathbf{C}$ is the temporal waveforms of all K neurons over all T frames. The dimensions of the matrices are labeled as superscripts. K is usually much smaller than N and T, so the dimensionality of the representation is much smaller than the dimensionality of the raw data. Researchers have developed many unsupervised learning algorithms to extract $\mathbf{A}$ and $\mathbf{C}$ in these models, so that they can segment neuron shapes and extract their temporal traces simultaneously. These methods mainly used optimization strategies aiming to maximize or minimize a target quantity or a cost function. For example, independent component analysis (PCA/ICA [21] and ICA + SRO [43]) maximized the spatial and temporal sparsity, and Suite2p [12] minimized the noise and used a classifier to refine segmented ROIs based on several statistics related to shape and activity. The cost functions under optimization were often constructed with constraints or regularizations matching calcium activities: the calcium activities of a neuron were spatially localized, temporally sparse, and nonnegative. For example, nonnegative matrix factorization (NMF [22]) minimized the noise under a nonnegativity constraint. Constrained nonnegative matrix factorization (CNMF [23], enhanced CNMF [44], panelized matrix decomposition (PMD) [45], and CNMF-E [24]) maximized the spatial and temporal sparsity under nonnegativity and noise constraints. Adding these constraints or regularizations forced the factorized matrices to better satisfy the physical properties of calcium imaging and made the results more interpretable. Dictionary learning approaches (ADINA [25], SSDT [29], and Temporal DL [28]) modeled the neurons and calcium activities as sparse signals in the spatial and/or temporal domains with a limited number of spatial and/or temporal convolutional kernels as basis functions. EXTRACT [46] analyzed the problem mainly through noise estimation: it solved the current cell's spatial and temporal components using the one-sided Huber estimator. These methods usually updated the spatial and temporal components alternatively using some iterative optimization algorithms. Some algorithms used additional dimensional reduction techniques like principal component analysis (PCA) [21,43], singular value decomposition (SVD) [12], or penalized matrix decomposition [45] before the matrix factorization to reduce the computation load and suppress noise. Haeffele et al. [47] discussed a generalized problem formation for these structured matrix factorization problems. They proved that although these problems were nonconvex, their local minima are global minima if the rank of the factorized matrices was sufficiently large and adapted to the data. Methods

not based on matrix factorization also exist. For example, the activity-based level set method (ABLE [48]) represented the boundaries of neurons using level set functions, and updated the level set functions by solving a partial differential equation.

Supervised learning methods seek to learn implicit features of neurons from the training data and apply the knowledge to new data. Unlike supervised 2D methods that aggregated temporal information before applying CNN, supervised 3D methods also dealt with temporal information within the CNN. ConvNet [42] temporally downscaled the video, so that each frame is much longer than the typical calcium transient duration, and then applied a (2 + 1)D CNN (2D spatial kernel and 1D temporal max-pooling) to the downscaled 3D video. This method treated spatial and temporal features differently. The algorithm effectively applied a 2D segmentation method on the downsampled frames of videos and applied a maximum projection on the output images. Because the temporal domain processing was relatively simple, this algorithm showed that the (2 + 1)D CNN could only achieve accuracy scores similar to a 2D CNN. STNeuroNet [14] used a full 3D CNN (DenseVNet) that learned both the spatial features of neurons and the temporal features of neural activities (Fig. 4B). Although it also had a temporal max-pooling layer within the CNN, the layers before the temporal max-pooling treated spatial and temporal dimensions equivalently. This method divided the videos into several temporal blocks to bypass the limited graphics card memory and used postprocessing to aggregate the information from all the blocks. Because the ground truth markings were usually just a set of neurons, STNeuroNet used the ground truth markings to create a binary video indicating whether a pixel was active in each frame. The algorithm then fed blocks of these binary videos as training ground truths together with blocks of spatially filtered raw videos to train the CNN.

Some methods combined both unsupervised and supervised learning paradigms. For example, CaImAn Batch [11] first used CNMF to propose active components. It then added a quality control step to screen these components with appropriate neuron shapes based on spatial footprint consistency, peak SNR, and a CNN classifier. The quality control step removed some nonneuron calcium sources, such as axons and dendrites that were not the main research interest. On the other hand, if the study focused on dendrites, the classifier could be trained to remove neurons and keep only dendrites. MIN1PIPE [35] also used CNMF to extract the spatial footprints and temporal traces of the candidate neurons, but it refined candidate neurons according to the temporal traces. They applied RNNs with a long-short-term memory (LSTM) module as the classifier for calcium spikes.

The algorithms processing 3D inputs generally produced more accurate segmentation than the algorithms processing 2D summary images. The 3D algorithms used the additional temporal information to separate overlapping neurons according to their individual active periods, or identify clusters of pixels that had temporal dynamics similar to calcium

activity. However, the speed of 3D algorithms was slower than the speed of 2D algorithms because processing temporal information carried a larger computation load. Nevertheless, the state-of-the-art algorithms in this class could process recordings at a speed close to the video rate, ~30 frames/s [14].

3.3 3D frame-by-frame segmentation methods

In addition to processing the video after collecting the entire imaging video (named offline processing), some in vivo neuroscience experiments need online processing. These experiments wanted to respond to neural behaviors in real-time, and some closed-loop experiments wanted to send feedback to neurons or the animal according to the recorded neural activities [15,16]. Because 3D block segmentation methods processed all frames together or a large block of frames each time, these algorithms had a significant delay between collecting data and obtaining segmented neurons and thus were not ready to use in real-time in vivo neuroscience experiments. Alternative 3D segmentation algorithms can process the video in a frame-by-frame fashion. Among these methods, some processed each frame without knowing any information from future frames, and they periodically exported a set of segmented neurons up to the most recent frame with a delay of at most a few frames. These methods are called online segmentation methods, and they can fulfill the tasks in real-time in vivo neuroscience experiments.

The 3D frame-by-frame segmentation methods also included unsupervised, supervised, or combined learning paradigms. Some methods modified 3D block segmentation methods to incrementally add one frame at a time. For example, OnACID [49] and OnACID-E [50] are the online versions of CNMF [23] and CNMF-E [24]. These methods used online NMF to estimate the temporal activity at the latest frame given the existing spatial components, and marked new components in each residue frame (Fig. 4C). To model the background of one-photon imaging, OnACID-E also used a ring-shaped CNN and trained it in an unsupervised way. Alternatively, some methods applied 2D segmentation techniques on each frame and aggregated the output from all frames. These algorithms benefited from the fact that each individual frame had only a few, well separated active neurons that were easy to segment. Therefore, while any existing 2D segmentation method can be used theoretically, much simpler 2D methods can significantly speed up the processing without sacrificing accuracy. For example, the dictionary learning method SCALPEL [27] identified connected components by simple thresholding in each frame and merged these components using prototype clustering based on dissimilarity. Supervised methods applied a CNN on each frame and aggregated the output from all frames, but the CNN can be much lighter than the CNN used to process 2D summary images. For example, SUNS [10] applied a shallow U-Net to each frame, thresholded and segmented active pixels into connected regions in each frame, and merged these connected regions from all frames. Similar to STNeuroNet, SUNS also

used the ground truth markings to create a binary video indicating whether a pixel is active in each frame to assist training. The number of parameters in the shallow U-Net of SUNS was three orders of magnitude smaller than in the original U-Net, so the training and execution were much faster. The low number of trained parameters also effectively prevented overfitting by learning only the most fundamental features. SCALPEL and the batch version of SUNS preprocessed the entire video to enhance neurons before segmenting each frame, and the aggregation process aggregated all frames together. The online version of SUNS modified the preprocessing so that it only relied on frames from the recent past, and it aggregated recent frames periodically to add the newly found neurons to the set of segmented neurons. Combining unsupervised and supervised learnings is also feasible. For example, CaImAn Online [11] was based on OnACID [49], and it added a quality control step to screen the segmented neuron candidates based on peak SNR and a CNN classifier.

In general, the accuracy and speed of 3D frame-by-frame segmentation methods were not significantly different from 3D block segmentation methods (Fig. 3). SUNS had a much faster speed than other 3D methods because of its light CNN and fast computation speed of modern GPUs.

4. Spike inference

In addition to neuron segmentation, many spike inference algorithms also implemented machine learning techniques. The goal of spike inference is to infer the time and amplitude of neural activity transients from the fluorescence intensity of a neuron as a function of time, also called a fluorescence trace. After segmenting neurons, the fluorescent trace can be obtained by averaging the fluorescent intensities of all pixels within the neuron masks in all frames. However, because some neurons overlap with each other, the overlapped pixels will cause crosstalk from the activities of overlapped neurons. Some methods have been developed to unmix activity from multiple overlapping neurons and create traces that contain only the activity of each individual neuron. The goal of spike inference is to locate the time and estimate the number or amplitude of the activity transients from the noisy traces. All traces contain not only the information of neural activities but also noise induced by either shot noise or electronic noise in the microscope electronics. Similar to neuron segmentation methods, spike inference methods also employ both unsupervised learning and supervised learning. However, different natures of the problems, such as dimensionality, necessitate different approaches when processing neural trace data.

When a neuron fires an action potential, or a "spike," its fluorescence trace will display a period of high intensity, also known as a fluorescence transient. A fluorescence transient displays a stereotypical rapid increase in intensity followed by a gradual decay. Although the relationship between the spikes and fluorescence waveforms is broadly consistent for

neurons in the same brain area expressing the same types of GECIs, the waveforms may differ stochastically over individual neurons or different brain areas and could be nonlinear. Nevertheless, in the first order, the fluorescence trace can be approximated as a spike train convolved with a typical rise-decay waveform plus some noise, so the spike inference problem is also sometimes named spike deconvolution. The simplest spike inference method is setting a threshold and assigning each rising edge across the threshold as a spike. However, this method is inaccurate in the presence of noise, and it lacks the ability to separate activities from close spikes. Likewise, conventional linear deconvolution based on Fourier transform is not suitable, because it cannot provide a temporally sparse spike train. Some more sophisticated deterministic methods used Schmitt-trigger thresholding (Peeling [51]), annihilating filter (finite rate of innovation (FRI) [52]), group delay (GDSpike [53]), or Bayesian inference (particle filter smoother (PFS) [54] and Markov chain Monte-Carlo (MCMC) [55]) methods to infer the spike times.

Similar to neuron segmentation, machine learning techniques, including unsupervised learning and supervised learning, have been widely used in spike inference algorithms. Unsupervised learning methods assumed a form of calcium generative model, where the spike times were estimated by maximizing a likelihood function (e.g., maximum-a-posteriori (MAP) [56] and MLSpike [57], Fig. 5A) or minimizing a cost

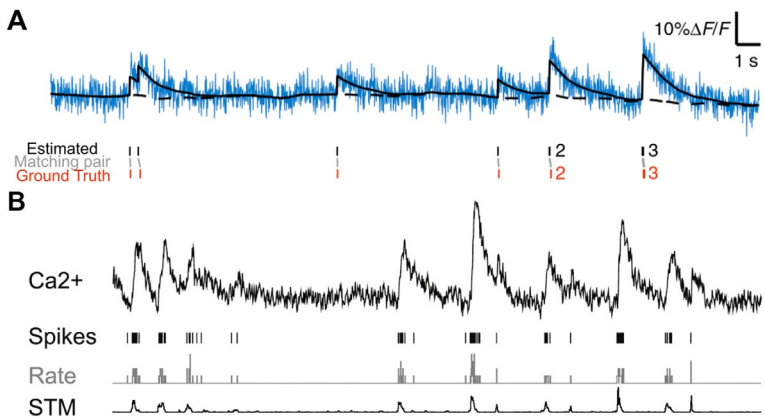


Fig. 5 Spike inference algorithms seek to find the calcium transients from fluorescent traces. *Multiple sources* (A) MLSpike is an unsupervised algorithm that predicted the time and numbers of action potentials from fluorescence traces. (B) STM is a supervised algorithm that predicted the spiking rate during a recording. *((A) Modified from T. Deneux, A. Kaszas, G. Szalay, G. Katona, T. Lakner, A. Grinvald, B. Rózsa, I. Vanzetta, Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo, Nat. Commun. 7(1) (2016) 12190, https://doi.org/10.1038/ncomms12190, Copyright © 2016. (B) From L. Theis, P. Berens, E. Froudarakis, J. Reimer, M. Román Rosón, T. Baden, T. Euler, A.S. Tolias, M. Bethge, Benchmarking spike rate inference in population calcium imaging, Neuron 90(3) (2016) 471–482, https://doi.org/10.1016/j.neuron.2016.04.014, Copyright © 2016 Elsevier Inc.)*

function (e.g., blind sparse deconvolution (BSD) [58], OASIS [59], nonnegative deconvolution (NND) [60], fast compressible state-space (FCSS) [61], and functional pruning [62]). The model parameters can be estimated directly from the recording (e.g., MLSpike [57]) or jointly with spike times (e.g., MAP [56], BSD [58], FCSS [61], and OASIS [59]). DeepSpike [63] used a recognition model consisting of a CNN and RNNs, which was independent of the generative model, but this method learned the parameters in both models simultaneously by jointly maximizing a likelihood function. By precisely fitting the spike time with the fluorescent trace, many algorithms can achieve super resolution in the temporal domain when processing high SNR recordings; these algorithms could locate the spike times more accurately than the sampling rate of the microscope [54,55,58].

Recently, many supervised learning techniques have been developed to make use of increasingly effective training data. Because humans were unable to precisely locate the spikes, the ground truth labels used to train the algorithms were usually obtained by simultaneous electrophysiology recording and two-photon microscopy [2,64,65]. Electrophysiology recordings can find all spikes and locate the times of spikes with microsecond precision. Supervised learning methods often did not assume explicit calcium generative models, but they learned the implicit relationship between the spikes and the traces by minimizing a loss function. The CNN was again one technique used in spike inference (e.g., signal-to-signal (S2S) [66] and CASCADE [67]), but it was no longer the dominant technique because of the different problem dimensionality. LSTM (e.g., DeepCINAC [68]), support vector machine (SVM) (e.g., PCA-SVM [69], HA_time [70]), and RNN (e.g., the supervised version of DeepSpike [63]) were all supervised learning techniques used in spike inference. Maximum likelihood can also be used in supervised learning; it maximized a conditional probability as the likelihood function (e.g., spike-triggered mixture (STM) [65]). Although neuroscientists desired precise discrete spike trains from optical recordings, using spike train labels directly often caused poorly trained supervised algorithms, because a slightly shifted spike significantly degraded the loss function. Therefore, many methods smoothed the spike trains in some way, such as convolving with a narrow Gaussian kernel and used the smoothed spike trains as the training labels. The outputs of these methods represented firing rates rather than discrete spike trains (Fig. 5B). Another difficulty in applying supervised learning algorithms is their low generalizability to recordings with different imaging conditions from the training data, such as frame rate and noise level. CASCADE [67] included methods to resample the original ground truth datasets to match the sampling rate and noise level to the test dataset.

The methods discussed above all processed the 1D temporal traces. Some methods can combine spike inference with previous steps in the overall calcium imaging analysis pipeline. These methods processed the entire 3D video; they combined trace unmixing with spike inference, and sometimes with neuron segmentation as well. In the neuron

segmentation methods discussed in the previous section, SSDT [29], ADINA [25], and CNMF [23] all solved neuron segmentation, trace calculation, and spike inference simultaneously. SSDT and ADINA used sparse dictionary learning, which represented the video using a sum of convolutions of basis functions and sparse spike trains. CNMF optimized the temporal sparsity of the spike trains to decompose the video into spatial and temporal components. 3D supervised learning was also possible, but it was unrealistic to create training labels by measuring the spikes of all neurons in a video using electrophysiology recording. Therefore, DeepCINAC [68] manually defined the spikes of neurons as the rising edges of the transients and excluded false transients caused by overlapped neurons by visual inspection of the video. As a result, the algorithm detected false transients caused by overlapped neurons and thus eliminated the need for trace unmixing. However, this algorithm relied on the manually labeled transients, and thus had accuracy limited by the accuracy of the labels.

5. Discussion

We focused on segmenting the body of neurons (somas), but some techniques can also segment and recover the activities of nonsomatic compartments, such as boutons, spines, axons, and dendrites [22,45,71,72]. Because human labels of dendrites and spines were lacking, existing methods of segmenting subcellular compartments were mainly unsupervised. In addition to the general methods, some methods were tailored for some specific forms of data generated by advanced microscopes. Seeded iterative demixing (SID [38]) adapted the matrix factorization approach to light-field microscopy. SCISM [73], developed for volumetric two-photon imaging of neurons using stereoscopy (vTwINS), iteratively selected neurons by greedily minimizing the residual norm. Finally, an algorithm incorporated the external stimulus as another dimension in the segmentation problem [74]; this method estimated cell shapes, spike times, and stimulus tuning curves simultaneously based on variational inference.

A complete calcium imaging analysis pipeline includes (1) motion correction, (2) neuron segmentation, (3) trace extraction, and (4) spike inference. We mainly focused on the methods for the second or fourth steps in this chapter, although many 3D block neuron segmentation methods based on unsupervised learning can also extract traces simultaneously, and some may even perform steps (2–4) together in their optimization process (e.g., SSDT [29], ADINA [25], and CNMF [23]). Some recent studies have simultaneously computed other combinations of steps. Deformable NMF (dNMF [75]) included a time-varying spatial deformation function in the matrix decomposition model, so they can perform steps (1–3) simultaneously.

In addition to calcium imaging, voltage imaging is another fluorescent imaging technique based on GEVIs. GEVIs usually have much faster kinetics than GECIs, and they can potentially resolve neural activities with an order of magnitude better temporal

resolution than that of GECIs. However, GEVIs are currently less developed than GECIs, and the current GEVIs report action potentials with smaller changes in relative fluorescence intensity than the changes of GECIs. The resulting low SNR presents challenges to the entire video processing pipeline. Nevertheless, algorithms for voltage imaging have also been developed [45,76].

In conclusion, we introduced a select set of neuron segmentation and spike inference algorithms using various machine learning techniques. These methods have played important roles in processing calcium imaging data generated by fluorescent microscopy and in retrieving information necessary for additional neuroscience analysis. Fast and accurate calcium imaging processing methods can replace labor-intensive human labeling and automate the analysis of neural activity. Fast online processing methods can potentially enable real-time closed-loop neuroscience experiments.

References

[1] J. Akerboom, N. Carreras Calderón, L. Tian, S. Wabnig, M. Prigge, J. Tolö, A. Gordus, M.B. Orger, K.E. Severi, J.J. Macklin, R. Patel, S.R. Pulver, T.J. Wardill, E. Fischer, C. Schüler, T.-W. Chen, K.S. Sarkisyan, J.S. Marvin, C.I. Bargmann, L.L. Looger, Genetically encoded calcium indicators for multi-color neural activity imaging and combination with optogenetics, Front. Mol. Neurosci. (2013), https://doi.org/10.3389/fnmol.2013.00002.

[2] T.-W. Chen, T.J. Wardill, Y. Sun, S.R. Pulver, S.L. Renninger, A. Baohan, E.R. Schreiter, R.A. Kerr, M.B. Orger, V. Jayaraman, L.L. Looger, K. Svoboda, D.S. Kim, Ultrasensitive fluorescent proteins for imaging neuronal activity, Nature 499 (7458) (2013) 295–300, https://doi.org/10.1038/nature12354.

[3] H. Dana, Y. Sun, B. Mohar, B.K. Hulse, A.M. Kerlin, J.P. Hasseman, G. Tsegaye, A. Tsang, A. Wong, R. Patel, J.J. Macklin, Y. Chen, A. Konnerth, V. Jayaraman, L.L. Looger, E.R. Schreiter, K. Svoboda, D.S. Kim, High-performance calcium sensors for imaging activity in neuronal populations and microcompartments, Nat. Methods 16 (7) (2019) 649–657, https://doi.org/10.1038/s41592-019-0435-6.

[4] M. Inoue, A. Takeuchi, S. Manita, S. Horigane, M. Sakamoto, R. Kawakami, K. Yamaguchi, K. Otomo, H. Yokoyama, R. Kim, T. Yokoyama, S. Takemoto-Kimura, M. Abe, M. Okamura, Y. Kondo, S. Quirin, C. Ramakrishnan, T. Imamura, K. Sakimura, H. Bito, Rational engineering of XCaMPs, a multicolor GECI suite for in vivo imaging of complex brain circuit dynamics, Cell 177 (5) (2019), https://doi.org/10.1016/j.cell.2019.04.007. 1346-1360.e24.

[5] A.S. Abdelfattah, T. Kawashima, A. Singh, O. Novak, H. Liu, Y. Shuai, Y.-C. Huang, L. Campagnola, S.C. Seeman, J. Yu, J. Zheng, J.B. Grimm, R. Patel, J. Friedrich, B.D. Mensh, L. Paninski, J.J. Macklin, G.J. Murphy, K. Podgorski, E.R. Schreiter, Bright and photostable chemigenetic indicators for extended in vivo voltage imaging, Science 365 (6454) (2019) 699–704, https://doi.org/10.1126/science.aav6416.

[6] Y. Gong, C. Huang, J.Z. Li, B.F. Grewe, Y. Zhang, S. Eismann, M.J. Schnitzer, High-speed recording of neural spikes in awake mice and flies with a fluorescent voltage sensor, Science 350 (6266) (2015) 1361, https://doi.org/10.1126/science.aab0810.

[7] K.D. Piatkevich, S. Bensussen, H. Tseng, S.N. Shroff, V.G. Lopez-Huerta, D. Park, E.E. Jung, O.A. Shemesh, C. Straub, H.J. Gritton, M.F. Romano, E. Costa, B.L. Sabatini, Z. Fu, E.S. Boyden, X. Han, Population imaging of neural activity in awake behaving mice, Nature 574 (7778) (2019) 413–417, https://doi.org/10.1038/s41586-019-1641-1.

[8] V. Villette, M. Chavarha, I.K. Dimov, J. Bradley, L. Pradhan, B. Mathieu, S.W. Evans, S. Chamberland, D. Shi, R. Yang, B.B. Kim, A. Ayon, A. Jalil, F. St-Pierre, M.J. Schnitzer, G. Bi, K. Toth, J. Ding, S. Dieudonné, M.Z. Lin, Ultrafast two-photon imaging of a high-gain voltage Indicator

in awake behaving mice, Cell 179 (7) (2019), https://doi.org/10.1016/j.cell.2019.11.004. 1590–1608. e23.
[9] T.R. Neu, J.R. Lawrence, One-photon versus two-photon laser scanning mic roscopy and digital image analysis of microbial biofilms, in: Methods in Microbiology, vol. 34, Academic Press, 2004, pp. 89–136, https://doi.org/10.1016/S0580-9517(04)34004-3.
[10] Y. Bao, S. Soltanian-Zadeh, S. Farsiu, Y. Gong, Segmentation of neurons from fluorescence calcium recordings beyond real-time, Nat. Mach. Intell. 3 (7) (2021) 590–600, https://doi.org/10.1038/s42256-021-00342-x.
[11] A. Giovannucci, J. Friedrich, P. Gunn, J. Kalfon, B.L. Brown, S.A. Koay, J. Taxidis, F. Najafi, J.L. Gauthier, P. Zhou, B.S. Khakh, D.W. Tank, D.B. Chklovskii, E.A. Pnevmatikakis, elife (2019) 8, https://doi.org/10.7554/eLife.38173.
[12] M. Pachitariu, C. Stringer, M. Dipoppa, S. Schröder, L.F. Rossi, H. Dalgleish, M. Carandini, K.D. Harris, Suite2p: beyond 10,000 neurons with standard two-photon microscopy, bioRxiv (2017) 061507, https://doi.org/10.1101/061507.
[13] E.A. Pnevmatikakis, Analysis pipelines for calcium imaging data, Curr. Opin. Neurobiol. 55 (2019) 15–21, https://doi.org/10.1016/j.conb.2018.11.004.
[14] S. Soltanian-Zadeh, K. Sahingur, S. Blau, Y. Gong, S. Farsiu, Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning, Proc. Natl. Acad. Sci. 116 (17) (2019) 8554–8563, https://doi.org/10.1073/pnas.1812995116.
[15] M.G. Kearney, T.L. Warren, E. Hisey, J. Qi, R. Mooney, Discrete evaluative and premotor circuits enable vocal learning in songbirds, Neuron 104 (3) (2019), https://doi.org/10.1016/j.neuron.2019.07.025. 559–575.e6.
[16] R. Sitaram, T. Ros, L. Stoeckel, S. Haller, F. Scharnowski, J. Lewis-Peacock, N. Weiskopf, M.L. Blefari, M. Rana, E. Oblak, N. Birbaumer, J. Sulzer, Closed-loop brain training: the science of neurofeedback, Nat. Rev. Neurosci. 18 (2) (2017) 86–100, https://doi.org/10.1038/nrn.2016.164.
[17] T.M. Mitchell, Machine Learning, McGraw-Hill Education, 1997. https://books.google.com/books?id=xOGAngEACAAJ.
[18] G. Mishne, R.R. Coifman, M. Lavzin, J. Schiller, Automated cellular structure extraction in biological images with applications to calcium imaging data, bioRxiv (2018) 313981.
[19] E. Kirschbaum, A. Bailoni, F.A. Hamprecht, DISCo for the CIA: deep learning, instance segmentation, and correlations for calcium imaging analysis, arXiv preprint (2020). Doi:arXiv:1908.07957.
[20] Q. Spaen, R. Asín-Achá, S.N. Chettih, M. Minderer, C. Harvey, D.S. Hochbaum, HNCcorr: a novel combinatorial approach for cell identification in calcium-imaging movies, Eneuro 6 (2) (2019), https://doi.org/10.1523/eneuro.0304-18.2019.
[21] E.A. Mukamel, A. Nimmerjahn, M.J. Schnitzer, Automated analysis of cellular signals from large-scale calcium imaging data, Neuron 63 (6) (2009) 747–760, https://doi.org/10.1016/j.neuron.2009.08.009.
[22] R. Maruyama, K. Maeda, H. Moroda, I. Kato, M. Inoue, H. Miyakawa, T. Aonishi, Detecting cells using non-negative matrix factorization on calcium imaging data, Neural Netw. 55 (2014) 11–19, https://doi.org/10.1016/j.neunet.2014.03.007.
[23] E.A. Pnevmatikakis, D. Soudry, Y. Gao, T.A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, M. Ahrens, R. Bruno, T.M. Jessell, D.S. Peterka, R. Yuste, L. Paninski, Simultaneous denoising, deconvolution, and demixing of calcium imaging data, Neuron 89 (2) (2016) 285–299, https://doi.org/10.1016/j.neuron.2015.11.037.
[24] P. Zhou, S.L. Resendez, J. Rodriguez-Romaguera, J.C. Jimenez, S.Q. Neufeld, A. Giovannucci, J. Friedrich, E.A. Pnevmatikakis, G.D. Stuber, R. Hen, Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data, elife 7 (2018), e28728.
[25] F. Diego, S. Reichinnek, M. Both, F.A. Hamprecht, Automated identification of neuronal activity from calcium imaging by sparse dictionary learning, in: 2013 IEEE 10th International Symposium on Biomedical Imaging, 2013, pp. 1058–1061, https://doi.org/10.1109/ISBI.2013.6556660.
[26] M. Pachitariu, A. Packer, N. Pettit, H. Dagleish, M. Hausser, M. Sahani, Extracting regions of interest from biological images with convolutional sparse block coding, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Curran Associates Inc, 2013, p. 1745.

[27] A. Petersen, N. Simon, D. Witten, SCALPEL: extracting neurons from calcium imaging data, Ann. Appl. Stat. 12 (4) (2018) 2430–2456, https://doi.org/10.1214/18-AOAS1159.
[28] G. Mishne, A.S. Charles, Learning spatially-correlated temporal dictionaries for calcium imaging, in: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2019, pp. 1065–1069, https://doi.org/10.1109/ICASSP.2019.8683375.
[29] F. Diego, F.A. Hamprecht, Sparse space-time deconvolution for calcium image analysis, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, MIT Press, 2014, pp. 64–72.
[30] Y. Wang, F. Su, S. Wang, C. Yang, Y. Tian, P. Yuan, X. Liu, W. Xiong, C. Zhang, Efficient implementation of convolutional neural networks in the data processing of two-photon in vivo imaging, Bioinformatics 35 (17) (2019) 3208–3210, https://doi.org/10.1093/bioinformatics/btz055.
[31] S.-B. Seong, H.-J. Park, Automated identification of neural cells in the multi-photon images using deep-neural networks, arXiv preprint arXiv:1909.11269 (2019).
[32] C. Stringer, T. Wang, M. Michaelos, M. Pachitariu, Cellpose: a generalist algorithm for cellular segmentation, Nat. Methods 18 (1) (2021) 100–106, https://doi.org/10.1038/s41592-020-01018-x.
[33] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, Springer International Publishing, 2015, pp. 234–241.
[34] A. Klibisz, D. Rose, M. Eicholtz, J. Blundon, S. Zakharenko, Fast, simple calcium imaging segmentation with fully convolutional networks, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer International Publishing, 2017, pp. 285–293.
[35] J. Lu, C. Li, J. Singh-Alvarado, Z.C. Zhou, F. Fröhlich, R. Mooney, F. Wang, MIN1PIPE: a miniscope 1-photon-based calcium imaging signal extraction pipeline, Cell Rep. 23 (12) (2018) 3673–3684, https://doi.org/10.1016/j.celrep.2018.05.062.
[36] K. Xu, H. Su, J. Zhu, J. Guan, B. Zhang, Neuron segmentation based on CNN with semi-supervised regularization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 1324–1332, https://doi.org/10.1109/CVPRW.2016.167.
[37] S.P. Shen, H. Tseng, K.R. Hansen, R. Wu, H.J. Gritton, J. Si, X. Han, Automatic cell segmentation by adaptive thresholding (ACSAT) for large-scale calcium imaging datasets, Eneuro 5 (5) (2018), https://doi.org/10.1523/eneuro.0056-18.2018.
[38] T. Nöbauer, O. Skocek, A.J. Pernía-Andrade, L. Weilguny, F.M. Traub, M.I. Molodtsov, A. Vaziri, Video rate volumetric Ca2+ imaging across cortex using seeded iterative demixing (SID) microscopy, Nat. Methods 14 (8) (2017) 811–818, https://doi.org/10.1038/nmeth.4341.
[39] F.D.W. Radstake, E.A.L. Raaijmakers, R. Luttge, S. Zinger, J.P. Frimat, CALIMA: the semi-automated open-source calcium imaging analyzer, Comput. Methods Prog. Biomed. 179 (2019) 104991, https://doi.org/10.1016/j.cmpb.2019.104991.
[40] J. Tomek, O. Novak, J. Syka, Two-photon processor and SeNeCA: a freely available software package to process data from two-photon calcium imaging at speeds down to several milliseconds per frame, J. Neurophysiol. 110 (1) (2013) 243–256, https://doi.org/10.1152/jn.00087.2013.
[41] J. Guan, J. Li, S. Liang, R. Li, X. Li, X. Shi, C. Huang, J. Zhang, J. Pan, H. Jia, L. Zhang, X. Chen, X. Liao, NeuroSeg: automated cell detection and segmentation for in vivo two-photon Ca2+ imaging data, Brain Struct. Funct. 223 (1) (2018) 519–533, https://doi.org/10.1007/s00429-017-1545-5.
[42] N.J. Apthorpe, A.J. Riordan, R.E. Aguilar, J. Homann, Y. Gu, D.W. Tank, H.S. Seung, Automatic neuron detection in calcium imaging data using convolutional networks, in: Advances in Neural Information Processing Systems, Neural Information Processing Systems (Nips), vol. 29, 2016. https://papers.nips.cc/paper/2016/file/0771fc6f0f4b1d7d1bb73bbbe14e0e31-Paper.pdf.
[43] Y. Levin-Schwartz, D.R. Sparta, J.F. Cheer, T. Adalı, Parameter-free automated extraction of neuronal signals from calcium imaging data, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 1033–1037, https://doi.org/10.1109/ICASSP.2017.7952313.
[44] T. Takekawa, H. Asai, N. Ohkawa, M. Nomoto, R. Okubo-Suzuki, K. Ghandour, M. Sato, Y. Hayashi, K. Inokuchi, T. Fukai, Automatic sorting system for large calcium imaging data, bioRxiv (2017) 215145.

[45] E.K. Buchanan, I. Kinsella, D. Zhou, R. Zhu, P. Zhou, F. Gerhard, J. Ferrante, Y. Ma, S. Kim, M. Shaik, Penalized matrix decomposition for denoising, compression, and improved demixing of functional imaging data, ar Xiv Preprint ar Xiv (2018), 1807.06203.
[46] H. Inan, M.A. Erdogdu, M.J. Schnitzer, Robust estimation of neural signals in calcium imaging, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc, 2017, pp. 2905–2914.
[47] B.D. Haeffele, R. Vidal, Structured low-rank matrix factorization: global optimality, algorithms, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 42 (6) (2020) 1468–1482, https://doi.org/10.1109/TPAMI.2019.2900306.
[48] S. Reynolds, T. Abrahamsson, R. Schuck, P.J. Sjöström, S.R. Schultz, P.L. Dragotti, ABLE: an activity-based level set segmentation algorithm for two-photon calcium imaging data, Eneuro 4 (5) (2017), https://doi.org/10.1523/eneuro.0012-17.2017.
[49] A. Giovannucci, J. Friedrich, M. Kaufman, A. Churchland, D. Chklovskii, L. Paninski, E.A. Pnevmatikakis, OnACID: online analysis of calcium imaging data in real time, bioRxiv (2017) 193383, https://doi.org/10.1101/193383.
[50] J. Friedrich, A. Giovannucci, E.A. Pnevmatikakis, Online analysis of microendoscopic 1-photon calcium imaging data streams, PLoS Comput. Biol. 17 (1) (2021), https://doi.org/10.1371/journal.pcbi.1008565, e1008565.
[51] B.F. Grewe, D. Langer, H. Kasper, B.M. Kampa, F. Helmchen, High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision, Nat. Methods 7 (5) (2010) 399–405, https://doi.org/10.1038/nmeth.1453.
[52] J. Oñativia, S.R. Schultz, P.L. Dragotti, A finite rate of innovation algorithm for fast and accurate spike detection from two-photon calcium imaging, J. Neural Eng. 10 (4) (2013), https://doi.org/10.1088/1741-2560/10/4/046017, 046017.
[53] J. Sebastian, M.G. Kumar, V.S. Viraraghavan, M. Sur, H.A. Murthy, Spike estimation from fluorescence signals using high-resolution property of group delay, IEEE Trans. Signal Process. 67 (11) (2019) 2923–2936, https://doi.org/10.1109/TSP.2019.2908913.
[54] J.T. Vogelstein, B.O. Watson, A.M. Packer, R. Yuste, B. Jedynak, L. Paninski, Spike inference from calcium imaging using sequential Monte Carlo methods, Biophys. J. 97 (2) (2009) 636–655, https://doi.org/10.1016/j.bpj.2008.08.005.
[55] E.A. Pnevmatikakis, J. Merel, A. Pakman, L. Paninski, Bayesian spike inference from calcium imaging data, in: 2013 Asilomar Conference on Signals, Systems and Computers, 2013, pp. 349–353, https://doi.org/10.1109/ACSSC.2013.6810293.
[56] J.T. Vogelstein, A.M. Packer, T.A. Machado, T. Sippy, B. Babadi, R. Yuste, L. Paninski, Fast nonnegative deconvolution for spike train inference from population calcium imaging, J. Neurophysiol. 104 (6) (2010) 3691–3704, https://doi.org/10.1152/jn.01073.2009.
[57] T. Deneux, A. Kaszas, G. Szalay, G. Katona, T. Lakner, A. Grinvald, B. Rózsa, I. Vanzetta, Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo, Nat. Commun. 7 (1) (2016) 12190, https://doi.org/10.1038/ncomms12190.
[58] J. Tubiana, S. Wolf, T. Panier, G. Debregeas, Blind deconvolution for spike inference from fluorescence recordings, J. Neurosci. Methods 342 (2020) 108763, https://doi.org/10.1016/j.jneumeth.2020.108763.
[59] J. Friedrich, P. Zhou, L. Paninski, Fast online deconvolution of calcium imaging data, PLoS Comput. Biol. 13 (3) (2017), https://doi.org/10.1371/journal.pcbi.1005423, e1005423.
[60] M. Pachitariu, C. Stringer, K.D. Harris, Robustness of spike deconvolution for neuronal calcium imaging, J. Neurosci. 38 (37) (2018) 7976–7985, https://doi.org/10.1523/jneurosci.3339-17.2018.
[61] A. Kazemipour, J. Liu, K. Solarana, D.A. Nagode, P.O. Kanold, M. Wu, B. Babadi, Fast and stable signal deconvolution via compressible state-space models, IEEE Trans. Biomed. Eng. 65 (1) (2018) 74–86, https://doi.org/10.1109/TBME.2017.2694339.
[62] S.W. Jewell, T.D. Hocking, P. Fearnhead, D.M. Witten, Fast nonconvex deconvolution of calcium imaging data, Biostatistics 21 (4) (2019) 709–726, https://doi.org/10.1093/biostatistics/kxy083.

[63] A. Speiser, J. Yan, E. Archer, L. Buesing, S.C. Turaga, J.H. Macke, Fast amortized inference of neural activity from calcium imaging data with variational autoencoders, arXiv preprint arXiv:1711.01846 (2017).
[64] P. Berens, J. Freeman, T. Deneux, N. Chenkov, T. McColgan, A. Speiser, J.H. Macke, S.C. Turaga, P. Mineault, P. Rupprecht, S. Gerhard, R.W. Friedrich, J. Friedrich, L. Paninski, M. Pachitariu, K.D. Harris, B. Bolte, T.A. Machado, D. Ringach, M. Bethge, Community-based benchmarking improves spike rate inference from two-photon calcium imaging data, bio Rxiv (2018) 177956, https://doi.org/10.1101/177956.
[65] L. Theis, P. Berens, E. Froudarakis, J. Reimer, M. Román Rosón, T. Baden, T. Euler, A.S. Tolias, M. Bethge, Benchmarking spike rate inference in population calcium imaging, Neuron 90 (3) (2016) 471–482, https://doi.org/10.1016/j.neuron.2016.04.014.
[66] J. Sebastian, M. Sur, H.A. Murthy, M. Magimai-Doss, Signal-to-signal neural networks for improved spike estimation from calcium imaging data, PLoS Comput. Biol. 17 (3) (2021), https://doi.org/10.1371/journal.pcbi.1007921, e1007921.
[67] P. Rupprecht, S. Carta, A. Hoffmann, M. Echizen, A. Blot, A.C. Kwan, Y. Dan, S.B. Hofer, K. Kitamura, F. Helmchen, R.W. Friedrich, Database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging, bioRxiv (2021), https://doi.org/10.1101/2020.08.31.272450, 2020.08.31.272450.
[68] J. Denis, R.F. Dard, E. Quiroli, R. Cossart, M.A. Picardo, DeepCINAC: a deep-learning-based python toolbox for inferring calcium imaging neuronal activity based on movie visualization, Eneuro 7 (4) (2020), https://doi.org/10.1523/eneuro.0038-20.2020.
[69] T. Sasaki, N. Takahashi, N. Matsuki, Y. Ikegaya, Fast and accurate detection of action potentials from somatic calcium fluctuations, J. Neurophysiol. 100 (3) (2008) 1668–1676, https://doi.org/10.1152/jn.00084.2008.
[70] H. Hoang, M. Sato, S. Shinomoto, S. Tsutsumi, M. Hashizume, T. Ishikawa, M. Kano, Y. Ikegaya, K. Kitamura, M. Kawato, K. Toyama, Improved hyperacuity estimation of spike timing from calcium imaging, Sci. Rep. 10 (1) (2020) 17844, https://doi.org/10.1038/s41598-020-74672-y.
[71] A. Kazemipour, O. Novak, D. Flickinger, J.S. Marvin, A.S. Abdelfattah, J. King, P.M. Borden, J.J. Kim, S.H. Al-Abdullatif, P.E. Deal, E.W. Miller, E.R. Schreiter, S. Druckmann, K. Svoboda, L.L. Looger, K. Podgorski, Kilohertz frame-rate two-photon tomography, Nat. Methods 16 (8) (2019) 778–786, https://doi.org/10.1038/s41592-019-0493-9.
[72] J. Prada, M. Sasi, C. Martin, S. Jablonka, T. Dandekar, R. Blum, An open source tool for automatic spatiotemporal assessment of calcium transients and local 'signal-close-to-noise' activity in calcium imaging data, PLoS Comput. Biol. 14 (3) (2018), https://doi.org/10.1371/journal.pcbi.1006054, e1006054.
[73] A. Song, A.S. Charles, S.A. Koay, J.L. Gauthier, S.Y. Thiberge, J.W. Pillow, D.W. Tank, Volumetric two-photon imaging of neurons using stereoscopy (vTwINS), Nat. Methods 14 (4) (2017) 420–426, https://doi.org/10.1038/nmeth.4226.
[74] R. Shibue, F. Komaki, Deconvolution of calcium imaging data using marked point processes, PLoS Comput. Biol. 16 (3) (2020), https://doi.org/10.1371/journal.pcbi.1007650, e1007650.
[75] A. Nejatbakhsh, E. Varol, E. Yemini, V. Venkatachalam, A. Lin, A.D.T. Samuel, L. Paninski, Extracting neural signals from semi-immobilized animals with deformable non-negative matrix factorization, bioRxiv (2020), https://doi.org/10.1101/2020.07.07.192120, 2020.07.07.192120.
[76] C. Cai, J. Friedrich, E.A. Pnevmatikakis, K. Podgorski, A. Giovannucci, Vol Py: automated and scalable analysis pipelines for voltage imaging datasets, bioRxiv (2020), https://doi.org/10.1101/2020.01.02.892323, 2020.01.02.892323.

Index

Note: Page numbers followed by *f* indicate figures and *t* indicate tables.

A

Active information storage (AIS), 236
Active particles, feedback control, 119–124
  experimental realization, 120–124, 121–124*f*
  light-controllable active particles, 119–120
Adversarial autoencoder (AAE), 25
AE. *See* Autoencoder (AE)
ANNs. *See* Artificial neural networks (ANNs)
Anti-counterfeiting, 361–362
  applications, 363–367
    luminescent materials, 365–366
    photonic crystals, 364–365
    plasmonic materials, 366–367
  encryption mechanism, 362–363, 362*f*
Artificial intelligence (AI), 14, 219
  authentication, 388–390, 389*f*
Artificial neural networks (ANNs), 15, 48, 130–133
Artificial synapses, 191–205, 226
  electronic migration, 199–202
    external field, 199–200
    photogenerated carrier migration, 200–202, 201*f*
  ferroelectric, 202–205, 203–204*f*
  ion migration, 191–199
    anion migration, 194–196, 195*f*
    cation migration, 191–194, 193*f*
    electrolyte-gated transistor, 196–199, 197–198*f*
  magnetic, 205
  phase transition, 202
Atomic force microscopy (AFM), 351
Autoencoder (AE), 49

B

Bandgap, 47
Bi-directional evolutionary structural optimization (BESO), 72
Biomedical informatics, 307
  applications, 309–312
    computational biology, 309–310
    gene expression, 310–311
    genomics, 311–312
    splicing analysis, 310
    transcriptomics, 310
    transfer learning, 309
  deep learning network, 308–312
    AE, 309
    CNN(s), 308
    LSTM, 308
    RNN(s), 308
  drug discovery, 312–315
    computer-aided, 312–315
  drug molecule identification, 313–314
  gene expression data analysis, 314–315
  medical images, 315–318
    computer-aided diagnosis, 317
    image registration, 316–317
    image segmentation, 316
    physical simulation, 317–318
  pharmacodynamics modeling, 315
  protein engineering, 314

C

cDCGAN. *See* Conditional deep convolutional generative adversarial network (cDCGAN)
Charles Darwin's theory, 10
Chromosome, 10
CNNs. *See* Convolutional neural networks (CNNs)
Computational inverse design, 3–13
  evolutionary algorithms, 10–13
    genetic algorithm (GA), 10–11, 12*f*
    particle swarm optimization (PSO), 11–13, 13*f*
  gradient-based, 3–10, 4*f*
    adjoint optimization, 5–7, 6*f*
    topology optimization (TO), 7–10, 9*f*
Conditional AAE (c-AAE), 25
Conditional deep convolutional generative adversarial network (cDCGAN), 21
Conditional generative adversarial network (cGAN), 21
Convolutional neural networks (CNNs), 36, 48, 145–146

Convolutional neural networks (CNNs) *(Continued)*
photonics, 146–154
mathematical operations, 148–150, 149–151*f*
parameterization strategies, 146–148, 147*f*
sampling, training, 151–154, 152–153*f*
Crossover, 10

D

Decentralized partially observable Markov decision processes (DEC-POMDP), 139–141
Deep learning approach, antibiotic discovery, 41
Deep learning-based inverse design, 14–21
deterministic neural network-based, 15–19
convolutional neural networks (CNNs), 16*f*, 18–19
multilayer perceptron, 15–18, 16*f*
generative neural network-based (GNN), 19–21
autoencoders (AE), 19–21, 20*f*
generative adversarial networks (GANs), 20*f*, 21
Deep learning merged, computational optimization, 22–27
generative neural networks
metaheuristic optimization, 15–19, 25–27
topology optimization (TO), 14–24
Deep Q-learning (DQL), 130–133
Density functional theory (DFT), 34
Differential evolutionary (DE) algorithm, 25
Dimensionality reduction models, 173–183
applications, 177–183, 178–179*f*, 182*f*
unsupervised learning, 174–177, 175*f*
Direct tabu search (DTS), 97

E

Echo state networks (ESNs), 240–242
Electrical resistance tomography (ERT), 236
Electric double layer (EDL), 196–197
Electrochemical metallization (ECM), 192
Electron beam lithography (EBL), 220–221
Electronic health record (EHR), 318–323
healthcare, 322–323
medical informatics, 318–320
public health, 320–322
Electronic synapse, 190–191
Energy transmission coefficient, 52
Evolutionary structural optimization (ESO), 79

F

Feedforward neural networks (FNNs), 240–242
Ferroelectric RAM (FeRAM), 190–191
Fiber Bragg grating (FBG), 97
Figure-of-merit (FOM), 88
Finite difference-time domain (FDTD), 151
Finite element analysis (FEA), 151
Finite element method (FEM), 3, 99
Fluorescent microscopy, 416
Frequency-resolved optical gating (FROG), 161
Fully connected neural networks (FCNNs), 145–146

G

GA. *See* Genetic algorithm (GA)
Generative adversarial networks (GANs), 114–115
Generative models, spectra design, 164–172
applications, GCNN, 168–172, 169*f*, 171–173*f*
generative CNNs, 164–168, 165–166*f*
Genes, 10
Genetic algorithm, 72
Goodfellow, Ian, 21
Gradient-based algorithms, 72–85
objective-first algorithm, 80–83, 82*f*
other gradient-based algorithms, 83–85
topology optimization, 72–80
bi-directional evolutionary structural optimization, 79–80, 79*f*
level set method, 76–79, 77–78*f*
variable density method, 73–76, 74*f*, 76*f*
Graph neural networks (GNNs), 145–146

H

Hardware architecture, 189–190, 190*f*
Heuristic algorithms, 85–108
ant colony algorithm, 106–108, 107*f*
differential evolution algorithm, 101–103, 102*f*
direct binary search, 87–90, 88–89*f*
genetic algorithm, 97–101, 99–101*f*

hill-climbing algorithm, 86–87, 86*f*
particle swarm optimization algorithm, 103–106, 104*f*
simulated annealing algorithm, 90–95, 92*f*, 94*f*
tabu search, 95–97, 96*f*
High resistance state (HRS), 192, 221–222

I

In-situ optical backpropagation training methods, 268–272
gradient calculation, electronic, 270–272
gradient calculation, optical, 268–270, 269*f*

L

Latent space, 19
Liquid state machines (LSMs), 240–242
Low resistance state (LRS), 192

M

Machine learning (ML), 14, 34, 114–115, 114*f*
nonimage data, 296–299, 297*f*
2D microscopy image analysis, 283–292
classification, regression, 283–285, 284*f*
object detection, 289–292, 291*f*
segmentation, 285–289, 287*f*
3D tomography reconstruction, 292–296, 293*f*
Magnetic RAM (MRAM), 190–191
Magnetic tunnel junction (MTJ), 205
Markov decision process (MDP), 125
MD. *See* Molecular dynamics (MD)
Metaheuristic approaches, 10
Metal-insulator-metal (MIM), 221
Metamaterial/metastructure, 47, 84
Microelectromechanism system (MEMS), 209
ML. *See* Machine learning (ML)
Mode analysis, 183–184
Molecular dynamics (MD), 34
Multi-agent reinforcement learning (MARL), 136–141, 140*f*
Multilayer perceptron(s) (MLP(s)), 14
Mutation, 10

N

Nanomaterial microscopy data, 279–283
high-throughput analysis, 279–280
machine learning models, 280–282
training datasets, 282–283
Nanophotonics, 5
Nanoscale systems, 348–349
Nanotechnology, AE, 331–332
case studies, 341–348
autonomy, automation, 348
carbon nanotube synthesis (CNT), 342–345, 344*f*
optimization, nanoparticles, 347–348
optoelectronic, 345–347
development, 332–341, 333*f*, 341*f*
automation, robotics, 333–336
domain knowledge, discovery systems, 336–338
optimization, data analysis, 338–340
platform technologies, 348–356
miniaturized liquid handling, 349–351, 350*f*
patterning, interrogating nanoscale materials, 351–356, 353*f*
Nanowire random networks, artificial neural networks, 228–243, 241–242*f*
criticality, avalanches effects, 239
emergent dynamics, 233–236, 235*f*
memristive behavior, single network elements, 231–233
reweighting effect, 231–233, 232*f*
rewiring effects, 233
nanowire network topology, 228–231, 229–230*f*
synaptic plasticity, 236–239
homosynaptic, heterosynaptic plasticity, 238–239
short-term, long-term synaptic plasticity, 236–238, 237*f*
structural plasticity, 238
Nanowires (NWs), 219
memristive devices, 220–225
ECM mechanism, 221–224, 222–223*f*
fabrication strategies, 221
NW synthesis, 220–221, 220*f*
VCM mechanism, 224–225, 225*f*
Neural networks, 14, 34
Neuromorphic engineering, 205–213
integration, cognitive functions, 209–213
artificial neuromorphic devices, 211–213, 212*f*
computing, synapses, 209–211, 210*f*
synaptic plasticity, 205–208

Neuromorphic engineering *(Continued)*
long-term potentiation and depression (LTP/LTD), 205–206
other plasticity, 207–208, 208*f*
short-term potentiation and depression (STP/STD), 206, 206*t*
spiking-time dependent plasticity (STDP), 207
Neuroscience, data processing, 399–401, 400*f*
genetically encoded indicators, 399
identifying active neurons, 404–412, 405*f*
2D segmentation, 406–408, 407*f*
3D block segmentation, 408–411
3D frame-by-frame segmentation, 411–412
ML, 401–404, 403*f*
spike inference, 412–415, 413*f*
Nonlinearity, 252

O

Optical anti-counterfeiting labels, 367–388
conventional, 367–384
luminescent anti-counterfeiting labels, 373–377, 375*f*
plasmonic anti-counterfeiting labels, 377–384, 379*f*, 382*f*
structural color-based, 367–373, 369*f*, 373*f*
physical unclonable function (PUF), 384–388, 385*f*
Optical encryption, 363
Optical metamaterials, 145
Optoelectronic devices, AI systems, 263–268
on-chip light source, 263–264
optoelectronic AI systems, 267–268
optoelectronic modulator, 266–267
photonic memory, 264–266, 265*f*

P

Paired pulse facilitation (PPF), 226–228
Partially observable Markov decision process (POMDP), 139–141
Particle swarm optimization (PSO), 103
Passbands, 47
Patterning techniques, 367
Perceptron, 14
Phononic crystals (PnCs), 47
design, neural networks, 63–68, 64*f*
datasets, inverse design, 65, 65*t*
one-parameter design, 65–66, 66*f*
three-parameter design, 67–68, 68*f*
two-parameter design, 66–67, 67*f*
forward prediction, neural networks, 1D, 53–63, 53*f*
dispersion curves, 54–61, 55*f*, 58–60*f*
transmission spectrums, 61–63, 62*f*
one-dimensional, transfer matrix method, 50–53, 50*f*, 53*f*
Photonic intelligent computing, 247–248
architectures, 257–263
convolutional, 260–261
optical fully connected, 258–259, 259*f*
recurrent, 261–262
spiking, 262–263
neuron activation function, 252–257
optical Kerr effect, 255
optical resonance, 256
photoelectronic effect, 256–257
photorefractive effect, 252–253
saturable absorption, 253–254, 254*f*
structural phase transition, 255
optical weighted interconnections, 248–252
free-space, 248–251, 249*f*
integrated, 251–252
Photonic memory, 263
PnCs. *See* Phononic crystals (PnCs)
Population selection, 10
Predictive models, spectra calculation, 154–163
fundamental spectra analysis, 154–161, 156*f*, 158*f*, 160*f*
ultrafast spectroscopy signal analysis, 161–163, 162*f*
Principal component analysis (PCA), 174
Principal component general projections algorithm (PCGPA), 163
PSO. *See* Particle swarm optimization (PSO)

Q

Quasi-Newton method (QNM), 97

R

Radial basis function (RBF) neural networks, 49
RCWA. *See* Rigorous coupled-wave analysis (RCWA)

Rectified linear unit (ReLU), 150
Recurrent neural networks (RNNs), 15, 145–146, 240–242
Reinforcement learning (RL), 124–139, 125*f*
  deep reinforcement learning, 130–136, 131–135*f*
  multi-agent systems, 136–139
    multi-agent reinforcement learning (MARL), 136–139, 137*f*, 139*f*
  real microswimmer, 128–130, 128–129*f*
Representative learning, 19
Reservoir computing, 240
Resistive RAM (ReRAM), 190–191
Resistive switching, 223–224
Rigorous coupled-wave analysis (RCWA), 8, 85
RNNs. *See* Recurrent neural networks (RNNs)

S

Second harmonic generation (SHG), 161
Signal-to-noise ratio (SNR), 163
Single nanowire memristor, artificial synapse, 226–228, 227*f*
Software-defined technology, 17
Solid isotropic material with penalization (SIMP), 38, 73
Solid mechanics, 33
  graph neural networks, 41–42, 42*f*
  inverse material design, 38–39, 38*f*
  physics-informed neural network, 39–41, 40*f*
  surrogate modeling, 35–37, 35*f*
    examples, 37
    predicting material properties, binary composites, 36–37
Spatial light modulator (SLM), 121
Spectra analysis, 146
Staged continuous tabu search (SCTS), 97
Static random access memory (SRAM), 190–191
Stress–strain curve, 36
Support vector machine (SVM), 25
Synthetic active particles, 116–119, 117–118*f*

T

Topological photonics, 75–76
Transfer entropy (TE), 236

U

Unconventional computing, 219–220
User-defined objective, 10

V

Variational autoencoder (VAE), 19–21, 176
Variational Bayes (VB), 176
Video processing, 399–401
Virtual electrodes, 194

W

Watt-Strogatz (WS) networks, 230–231