

Real-time deep learning design tool for far-field radiation profile

JINRAN QIE,¹ ERFAN KHORAM,² DIANJING LIU,² MING ZHOU,²  AND LI GAO^{3,*}

¹Department of Electrical and Systems Engineering, Washington University, St Louis, Missouri 63130, USA

²Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin 53706, USA

³Key Laboratory for Organic Electronics & Information Displays (KLOEID), Institute of Advanced Materials (IAM), and School of Materials Science and Engineering, Nanjing University of Posts & Telecommunications, Nanjing 210046, China

*Corresponding author: iamlgao@njupt.edu.cn

Received 27 October 2020; revised 28 December 2020; accepted 2 February 2021; posted 2 February 2021 (Doc. ID 413567); published 23 March 2021

The connection between Maxwell's equations and artificial neural networks has revolutionized the capability and efficiency of nanophotonic design. Such a machine learning tool can help designers avoid iterative, time-consuming electromagnetic simulations and even allows long-desired inverse design. However, when we move from conventional design methods to machine-learning-based tools, there is a steep learning curve that is not as user-friendly as commercial simulation software. Here, we introduce a real-time, web-based design tool that uses a trained deep neural network (DNN) for accurate far-field radiation prediction, which shows great potential and convenience for antenna and metasurface designs. We believe our approach provides a user-friendly, readily accessible deep learning design tool, with significantly reduced difficulty and greatly enhanced efficiency. The web-based tool paves the way to present complicated machine learning results in an intuitive way. It also can be extended to other nanophotonic designs based on DNNs and replace conventional full-wave simulations with a much simpler interface. © 2021 Chinese Laser Press

<https://doi.org/10.1364/PRJ.413567>

1. INTRODUCTION

Nanophotonic devices offer new capabilities to control light with nanostructures designed for different functionalities. In these photonic devices, a large number of geometric parameters play critical roles in altering the light-matter interaction. For complex nanostructures, there could be millions or even billions of combinations of all possible structures. However, conventional design methods rely on time-consuming, full-wave simulations and an iterative optimization process. It is challenging to explore all the options; usually only a limited number of designs are explored, leaving an enormous parameter space underexplored.

Machine learning has led to revolutionary developments in numerous applications. Its complex models and algorithms can help exploit the enormous parameter space in nanophotonics, enabling both efficient forward prediction and on-demand inverse designs. Artificial neural networks (ANNs) [1–6] are an interconnected group of nodes that are similar to the complicated network of neurons in a brain with the capability of self-learning. It is a data-driven approach, which is in contrast to a computation-driven approach, such as optimization [7–10]. Recent representative examples include near- and far-field prediction [11], metasurface and metamaterials designs [12–14],

and structural color design [15]. The merging of deep learning and nanophotonics has reduced computation time by orders of magnitude and expands the design space that previously could not be realized.

In Ref. [11], the authors show good agreements of near-field and far-field scattering of three-dimensional (3D) nanostructures between the simulation results and the neural network prediction by using a convolutional neural network. Unfortunately, the transition from conventional simulation to deep-learning-based tools requires knowledge from both nanophotonics and computation. There is a steep learning curve that hinders researchers from accessing such a convenient and efficient tool. On the other hand, there is no demonstration of the application of deep neural networks (DNNs) for nanophotonic design in a web-based, real-time setting. If a trained DNN can be interfaced with a web-page tool, the real application process can be even simpler than using commercial simulation software, with accurate results displayed in real time. In this way, device designers can effortlessly benefit from deep-learning-enabled computation.

Here, we demonstrate the training of DNNs for accurate far-field pattern prediction of dielectric antennas. Then we interface the DNNs with a web-page tool for real-time design output. The far-field radiation profile [16] is used in the design

of many nanophotonic devices, such as optical antennas [17,18] and metasurfaces [19–21]. The conventional approach to obtain a far-field radiation pattern is through finite difference time domain (FDTD) simulation with near- to far-field transformation [22,23], or using a commercial/open source software package; either method can take from tens of minutes to hours to complete the computation. In practice, iterative optimization is necessary to get an optimal structure. For repetitive designs, it requires a considerable amount of computation resource and time. By investing one set of DNNs training data, we demonstrate an online tool to predict the far-field radiation pattern of any arbitrary scatterer in real time. The results suggest that a web-page tool can maximize the advantages of a DNN-based design method and significantly improve a designer's productivity.

2. TRAINING OF DNN PREDICTORS

Here we use 2D scatterers to demonstrate our method. It can be easily generalized to high-dimensional spaces. We consider a silicon dioxide ($\epsilon_r = 2.1$) scatterer in air ($\epsilon_r = 1$) with a finite dimension. For example, here we limit the size of the scatterer to 1.5λ on each side, where λ is the incident wavelength. Our goal is to build a tool that can instantaneously provide a far-field radiation profile when the SiO_2 scatterer is illuminated by a plane wave. The structure of the scatterer, which is a binary image, is used as the input of the neural network. The output of the network is a vector, which corresponds to the directivity of the radiation pattern in the angular range from 0 to 360 deg. The relationship of the output and the input can be seen in Fig. 1(a).

We first need to create training samples to train the network so that it can perform the function as described above. Each training sample consists of a pair of data: the structure and the far-field radiation pattern. The training set contains 87,000 training samples. In addition to the training set, we also create a test set that contains 11,000 pairs of structures and their far-field patterns. It will be used to test the performance of the trained neural network.

One crucial factor in evaluating the quality of the training data set is the diversity of the training samples. To increase the diversity, we use a large number of geometries with random features across different length scales. Specifically, we use three fundamental shapes in our design: a rectangle, a circle, and a triangle. We randomly vary the geometrical parameters including positions, the side length for rectangles, the radius of circles, and the side length and angle for the triangles. We also randomly combine the number of each shape. One of the structure examples is shown in the left panel in Fig. 1(a), with the blue part representing SiO_2 . The corresponding matrix size of the structure is 30×30 , which corresponds to a spatial resolution of $\lambda/20$.

Next, we discuss the calculation of a far-field pattern in the training samples, which are done by full-wave simulations of Maxwell's equations. We use the 2D finite difference frequency domain (FDFD) method [24] to obtain the scattered field and total field of the input structure. In our case, the incident plane wave comes from the left side and propagates along the x axis with TE polarization. After getting the near-field radiation

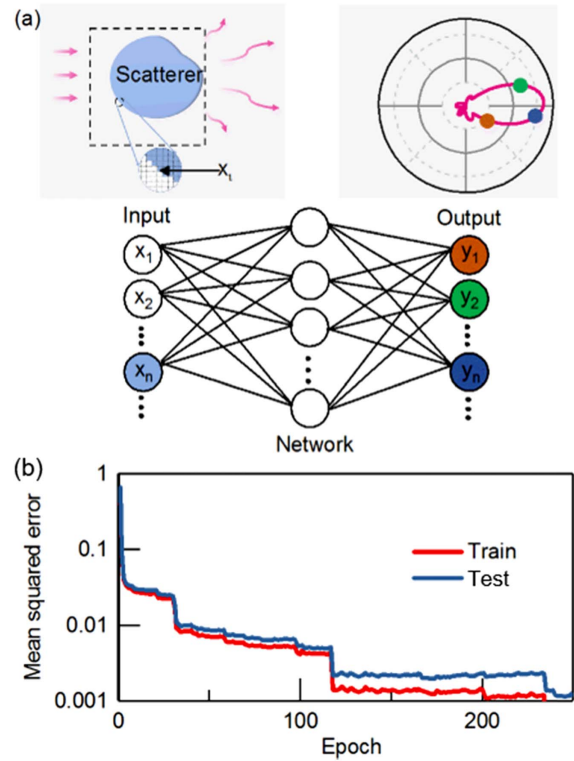


Fig. 1. Illustration of our approach and the loss curve of the neural network. (a) Sketch of a scatterer and its far-field pattern. (b) Loss curve of our neural network.

pattern, we use near-field/far-field transformation method (i.e., the Stratton–Chu formula), to obtain the far-field patterns. The Stratton–Chu formula [25] in 2D is expressed as

$$\mathbf{E}_P = \sqrt{\lambda} \frac{jk}{4\pi} \mathbf{r}_0 \times \int [\mathbf{n} \times \mathbf{E} - \eta \mathbf{r}_0 \times (\mathbf{n} \times \mathbf{E})] e^{jkr \cdot \mathbf{r}_0} dS, \quad (1)$$

where \mathbf{E} and \mathbf{H} are fields on the surface S enclosing the scatterer, \mathbf{r}_0 is the unit vector pointing from the origin to the field point P , \mathbf{r} is the radius vector of the surface S , \mathbf{n} is the unit normal to the surface S , η is the impedance, approximately equaling 377Ω in air, k is the wavenumber, and \mathbf{E}_P is the calculated far field in the direction from the origin toward point P . A circular boundary is used inside the scattered field when realizing the integral of transformation, and the far-field radiation pattern of the example is shown in Fig. 1(a), on the right panel. To get accurate near-to-far-field transformation, here we used a very high spatial resolution $\lambda/100$ to perform the simulation. The simulations are performed at the Center for High Throughput Computing (CHTC) [26] at the University of Wisconsin–Madison. The far-field patterns are used as the ground-truth patterns that are consistent with the commercial software results.

The network is fully connected. The loss function is the L2 loss defined as

$$J = \frac{1}{2} \sum_i (r_i - o_i)^2, \quad (2)$$

where r_i is the magnitude of the electric field, and o_i is the network output value corresponding to the electric field at

different angles. Since the original values of the electric far field are about 10^{-4} , the values multiplied by 10^3 are used to facilitate the training process. Figure 1(b) shows the loss curve with a rapid decline, reaching a very small value of test loss, which means the neural network works very well for our problems. We also noted that there are a few steep drops in the curve, and we assume that, as the training process proceeds, the loss is approaching some local minimums or saddle points in the parameter space, so the loss value gradually converges first. For the learning rate that we used, however, it could help the loss escape from those local minimums or saddle points and decrease again quickly after a few epochs. The absolute value that the loss drops at large epochs is smaller, which means the loss value is finally close to the global minimum.

Hyperparameters are chosen by a grid search, including the learning rate, batch size, and the number of layers and units. Our network architecture has six hidden layers with 8192 units in every layer. We use the activation function leaky rectified linear unit (leaky ReLU) [27] whose leaky rate is 0.2 for every hidden layer, and ReLU for the output layer. The reason for using ReLU for the output layer is that our far-field values are positive numbers without a specific range and nonlinear functions like tanh or sigmoid work only within a small range of input values, which limits the scope of the output values. An AdamOptimizer with the learning rate 2×10^{-5} is employed, and the batch size is 128. The input structures are flattened to a 900-by-1 vector, and the output layer has 1000 units to depict the far-field radiation pattern at different angles. The optimization process of DNNs is briefly introduced as follows. First, the parameters (i.e., weight and bias of each neural

node) of the neural network are randomly initialized. With the input structures, the output of the whole neural network is then calculated. The L2 loss is used to evaluate the performance of these parameters, and they are optimized subsequently by error back propagation [28] to minimize the loss value. By modifying the hyperparameters and repeating the training process, we could finally achieve a very low L2 loss and, at the same time, the far-field pattern prediction is good enough.

Figure 2 shows a series of results from the test set. Here we compare the far-fields patterns produced by the neural network and the ground truth, which is produced using full-wave simulations. As shown in Fig. 2(a), they agree with each other very well for a variety of different geometries. In Fig. 2(b), we show some typical scatterer shapes with different physical sizes to see if the network can handle structures with different sizes. In fact, no matter how large or small the structures are, the far-field patterns generated by the neural network are consistent with the ground truth.

On a laptop with an Intel core-i7 4720HQ, the neural network takes about 500—or even fewer—milliseconds to compute the far-field patterns. This speed makes it possible to design a far-field radiation profile in real time. One can modify the structure and instantaneously obtain real-time feedback about the far fields. Here, we further develop an online tool to demonstrate this capability.

3. ONLINE DNN TOOL

After obtaining the trained neural network model, traditionally one needs to work with the code script to input the test structure, call the trained model, and then show the results, which is not convenient and intuitive for a practical design process. It also is not straightforward for the designers in optics without knowledge of machine learning methods to improve productivity. In this case, a method that can translate the machine learning results to an intuitive manifestation will be very useful for the optical community. Here, we design and implement a very efficient web tool to realize the calculation procedure, which combines such features as user-friendly and highly efficient to generate results, with extensibility. It also can be hosted on a website to provide easy access to a community. The main advantage of the web-page tool is that designers can intuitively use the online tool for their designs without knowing the underlying programming methods.

The application programming interface functions in TensorFlow for deep learning are transferred into the Javascript provided by Ref. [29]. Basically, we must develop two interface windows to draw the scatterer structures and exhibit the output far-field profile, and the well-trained neural network model is imported by downloading it from a known source path. For the input window, we specify two materials for the scatterer (i.e., air and silica), and one can easily extend them to multiple materials. Two functional buttons, including the undo and clear operation, are used to help the design. Then we extract the structure data from the input window as a matrix and then flatten it into a vector. The same fully connected neural network architecture as the trained model is employed with the hyperparameters and activation function to calculate the far field. The raw trained model is transferred to the binary file

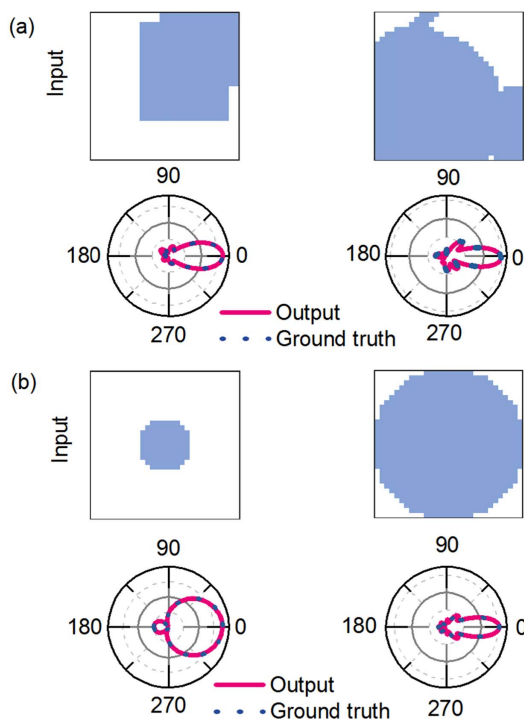


Fig. 2. Examples of results from our method on: (a) random structures from the test set; and (b) typical shapes with different sizes, compared to the ground truth.

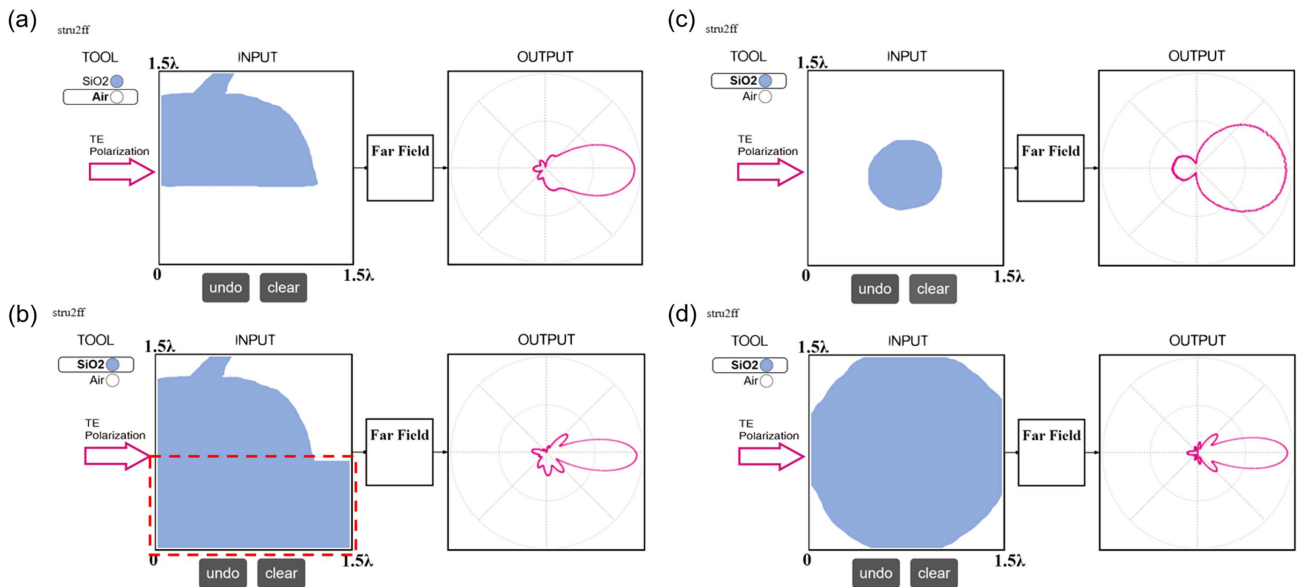


Fig. 3. Two examples of calculating a far-field pattern using our web tool. (a), (b) Two steps of drawing the final structure, which is the one on the right side in Fig. 2(a) and the related far-field pattern after each drawing stroke. (c), (d) Corresponding to the structures in Fig. 2(b). We also provide a video to show the design process online.

format for the web script to recognize the trained network parameters. Finally, we draw the far-field profile based on the polar coordinate. A screenshot of the web tool is shown in Fig. 3, and the framework is based on the one developed by Ref. [29]. In our web tool, one can hand-draw a shape on the input box, and the far-field radiation pattern will be available right away after each drawing stroke. Currently, the web page supports Firefox 67.0 or earlier. We also host the source codes on GitHub (see Code 1, Ref. [30]).

The operational details of the web are depicted as follows. There are 256×256 pixels in the input box. During the drawing process, a handwritten figure step by step, we extract a 30×30 matrix by downsampling from the original input matrix. At each step, we use our trained network model imported from TensorFlow to calculate and exhibit the far-field radiation pattern on the output window. Figures 3(a) and 3(b) show the operation of the tool. The far field is calculated in real time as we draw a structure. The structures we draw in Fig. 3 are similar to those in Fig. 2(a), and we can see that the far-field profile is almost the same as the ground truth. A real application process of the web tool can be found in Visualization 1. It proves that our tool will be very effective to improve the far-field design efficiency. Our tool demonstrates a new perspective to use the current emerging machine learning technologies to facilitate the complicated design process in the optics and improve productivity greatly. The neural network can also be trained to realize the inverse design [12,13,15,31, 32], and thus, by using the same method, one can also integrate multiple features into the tool.

4. CONCLUSION

In summary, we propose a deep learning integrated online tool to facilitate the design of far-field radiation. Unlike solving a

full-wave Maxwell's equation, our tool produces the results through DNNs in real time. Our demonstration shows that the DNN online design tool not only decreases the computation cost and time by orders of magnitude, but also provides a user-friendly platform compared to conventional software. More importantly, it shows that complicated DNNs methods could be translated to a very simple interface so that others could use it easily without any prior knowledge about a neural network. There are also other aspects of the tool that can be further improved to extend its utility, including different materials, wideband applications, and even inverse design of nanophotonics. In the future, researchers could share their trained model by integrating it with the web-based tool. We believe this will definitely enhance the usefulness of deep learning as a method to improve optical design efficiency.

Funding. National Natural Science Foundation of China (61974069, 62022043); Natural Science Foundation of Jiangsu Province (BK20191379); NUPTSF (NY219008); NJUPT 1311 Talent Program.

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323**, 533–536 (1986).
2. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.* **2**, 359–366 (1989).
3. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA* **79**, 2554–2558 (1982).
4. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield model," *Appl. Opt.* **24**, 1469–1475 (1985).
5. Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and

- M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**, 441–446 (2017).
6. M. Hermans, M. Burm, T. Van Vaerenbergh, J. Dambre, and P. Bienstman, "Trainable hardware for dynamical computing using error backpropagation through physical media," *Nat. Commun.* **6**, 6729 (2015).
 7. P. Seliger, M. Mahvash, C. Wang, and A. F. J. Levi, "Optimization of aperiodic dielectric structures," *J. Appl. Phys.* **100**, 034310 (2006).
 8. A. Oskooi, A. Mutapcic, S. Noda, J. D. Joannopoulos, S. P. Boyd, and S. G. Johnson, "Robust optimization of adiabatic tapers for coupling to slow-light photonic-crystal waveguides," *Opt. Express* **20**, 21558–21575 (2012).
 9. A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, and J. Vučković, "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," *Nat. Photonics* **9**, 374–377 (2015).
 10. B. Shen, P. Wang, R. Polson, and R. Menon, "An integrated-nanophotonics polarization beamsplitter with $2.4 \times 2.4 \mu\text{m}^2$ footprint," *Nat. Photonics* **9**, 378–382 (2015).
 11. P. R. Wiecha and O. L. Muskens, "Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3D nanostructures," *Nano Lett.* **20**, 329–338 (2019).
 12. Z. Liu, D. Zhu, S. P. Rodrigues, K. T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Lett.* **18**, 6570–6576 (2018).
 13. W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," *ACS Nano* **12**, 6326–6334 (2018).
 14. I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, "Plasmonic nanostructure design and characterization via deep learning," *Light Sci. Appl.* **7**, 60 (2018).
 15. L. Gao, X. Li, D. Liu, L. Wang, and Z. Yu, "A bidirectional deep neural network for accurate silicon color design," *Adv. Mater.* **31**, 1905467 (2019).
 16. C. A. Balanis, *Antenna Theory Analysis and Design* (Wiley, 2005).
 17. P. Bharadwaj, B. Deutsch, and L. Novotny, "Optical antennas," *Adv. Opt. Photon.* **1**, 438–483 (2009).
 18. L. Novotny and N. Van Hulst, "Antennas for light," *Nat. Photonics* **5**, 83–90 (2011).
 19. N. Yu and F. Capasso, "Flat optics with designer metasurfaces," *Nat. Mater.* **13**, 139–150 (2014).
 20. A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, "Planar photonics with metasurfaces," *Science* **339**, 1232009 (2013).
 21. D. Lin, P. Fan, E. Hasman, and M. L. Brongersma, "Dielectric gradient metasurface optical elements," *Science* **345**, 298–302 (2014).
 22. K. Umashankar and A. Taflove, "A novel method to analyze electromagnetic scattering of complex objects," *IEEE Trans. Electromagn. Compat.* **EMC-24**, 397–405 (1982).
 23. X. Li, A. Taflove, and V. Backman, "Modified FDTD near-to-far-field transformation for improved backscattering calculation of strongly forward-scattering objects," *IEEE Antennas Wireless Propag. Lett.* **4**, 35–38 (2005).
 24. W. Shin and S. Fan, "Choice of the perfectly matched layer boundary condition for frequency-domain Maxwell's equations solvers," *J. Comput. Phys.* **231**, 3406–3431 (2012).
 25. J. A. Stratton and L. J. Chu, "Diffraction theory of electromagnetic waves," *Phys. Rev.* **56**, 99 (1939).
 26. Center for High Throughput Computing, <http://chtc.cs.wisc.edu/>.
 27. B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv:1505.00853 (2015).
 28. R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception* (Academic, 1992), pp. 65–93.
 29. P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1125–1134.
 30. <https://github.com/JRQie/Web-demo-for-far-field-pattern>.
 31. D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training deep neural networks for the inverse design of nanophotonic structures," *ACS Photon.* **5**, 1365–1369 (2018).
 32. J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.* **4**, eaar4206 (2018).