ACS Photonics

pubs.acs.org/journal/apchd5

Article

Robust Freeform Metasurface Design Based on Progressively Growing Generative Networks

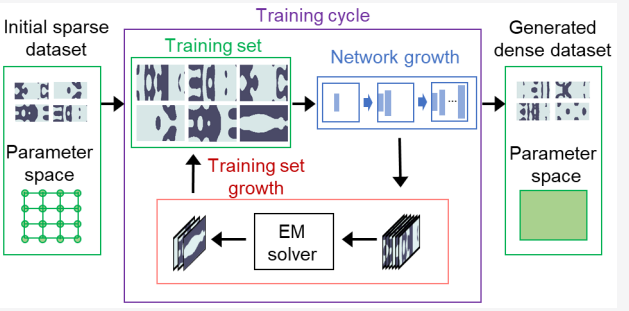
Fufang Wen,[§] Jiaqi Jiang,[§] and Jonathan A. Fan*

Cite This: https://dx.doi.org/10.1021/acsphotonics.0c00539

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: A longstanding objective of machine learning-enabled inverse design is the realization of inverse neural networks that can instantaneously output a device given a desired optical function. For complex freeform devices, generative adversarial networks (GANs) can learn from images of freeform devices, but basic GAN architectures are unable to fully capture the intricate features of topologically complex structures. We show that by coupling progressive growth of the network architecture and training set with the GAN framework, generative networks can be trained to output high-performance, robust freeform metasurface devices. A combination of convolutional and self-attention layers in the network enable the accurate capture of both short- and long-range spatial patterns within topologically complex layouts. In applying this training methodology to metagratings, the best generated devices have efficiency and robustness metrics that compare with or outperform the best devices produced by gradient-based topology optimization with comparable computational cost. This study showcases the capability of generative neural networks to capture highly intricate geometric trends in physical devices, such as robustness constraints in freeform metasurfaces, and demonstrates their potential as black box inverse design tools for complex photonic technologies.



KEYWORDS: *Generative adversarial networks, dielectric metasurfaces, metagrating, progressive growth, self-attention, inverse design*

INTRODUCTION

Metasurfaces are nanostructured electromagnetic media with responses tailored by structure geometry.[1−3] They are an emergent technology relevant to a wide scope of applications, including those in imaging,[4] sensing,[5] polarization control,[6] and holography.[7] The metasurface design process involves the creation of a library of meta-atoms which are subwavelength- or wavelength-scale structures that possess tailored optical response parameters such as amplitude, phase, polarization, and dispersion. A subset of meta-atoms is then stitched together to produce the desired macroscopic optical device. Meta-atoms can be specified as simple geometric shapes based on physical intuition, which work well for a limited subset of optical functions, or as complex freeform shapes based on numerical optimization.[8] For the latter, which is the focus here, design methods ranging from topology optimization[9,10] to heuristic methods[11] have enabled high-efficiency, multifunctional metagratings,[12−15] and metasurfaces.[16,17]

It is an open question how to effectively and efficiently fill out a meta-atom library of freeform elements, which can include many thousands of structures. One option is to optimize every meta-atom from scratch in a brute force fashion. This is computationally expensive, as each meta-atom optimization requires hundreds to thousands of simulations, and the computation time scales linearly with the size of the meta-atom library. An alternative method is to first optimize a sparse sampling of the meta-atom library and then use high-dimensional interpolation to fill out the rest of the library. This scheme has the potential to computationally scale more efficiently with the size of the meta-atom library because it utilizes geometric correlations between meta-atoms with related optical functions. In this vein, conditional generative networks are an attractive framework for performing high-dimensional interpolation due to their ability to identify and capture highly nonlinear distributions within data sets. In particular, a properly trained conditional generative network that can generate meta-atom layouts given a desired optical response parameter input can serve as the meta-atom library itself.

It remains an open question how to train a neural network to output high-performance freeform meta-atoms, particularly those possessing complex constraints such as robustness to geometric imperfections. Deep discriminative networks can

Received: April 3, 2020
Published: June 19, 2020

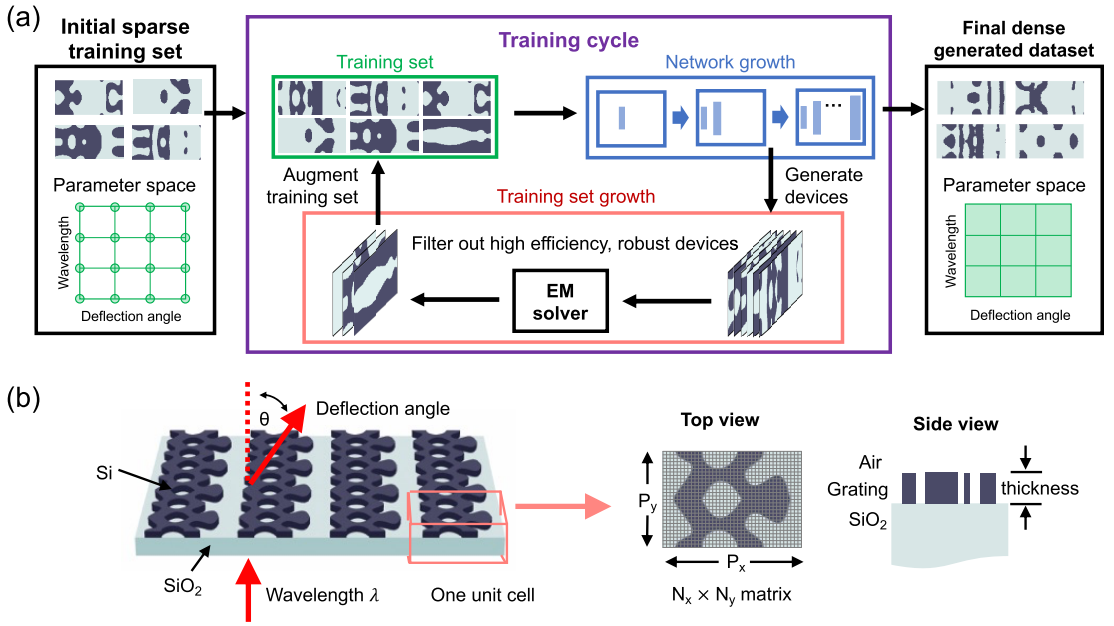


Figure 1. Overview of GANs based on progressively growing network architectures and training sets. (a) Schematic of the training protocol. The initial training set comprises high-performing devices that sparsely sample the device parameter space. The training process consists of multiple training cycles, each of which involves training the PGGAN from scratch, evaluating the generated devices upon training completion, and augmenting the training set with the best generated devices. The final trained PGGAN can output device layouts that continuously span the full device parameter space. (b) Freeform silicon metagratings deflect normally incident light to the +1 diffraction order. Each metagrating period is described by an $\mathbf{N}_x$ by $\mathbf{N}_y$ matrix, where each pixel represents silicon or air. The width of the period along the x-direction is $\lambda/\sin(\theta)$ and the width of the period along the y-direction is set to be $\lambda/2$.

serve as inverse models for devices described by a few geometric parameters[18,19] but cannot scale to complex freeform structures due to the enormity of the design space. Generative adversarial networks (GANs)[20−23] are a more promising network architecture that can learn salient features from images of freeform geometries. In a recent study, we trained a GAN using images of robust topology-optimized metagratings,[21] which are periodic metasurfaces that selectively diffract incident light to a desired diffraction order. Robustness here refers to the ability for devices to exhibit high efficiencies when the layouts are spatially eroded and dilated.[24] Our GAN, which possessed a conventional deep network architecture, could generate topologically complex metagratings operating across a range of wavelengths and deflection angles. However, the efficiencies of the best GAN-generated devices were consistently lower than those in the training set, and none of the generated devices were robust. The shortcomings of the conventional GAN approach are due to a few factors: the design space of the training set is vast, making high-dimensional interpolation difficult;[25] the limited representation capacity of convolutional layers in conventional GANs makes the learning of global structural patterns difficult;[26] and the use of a small training set makes it easy for the discriminator to overfit.

In this paper, we describe a new training procedure and network architecture for GANs that enable the neural network to output robust, freeform metasurfaces with performance metrics similar to or even exceeding high-performing topology-optimized devices. Our new training procedure, schematically presented in Figure 1a, features a combination of innovations. First, we utilize the progressive growth of GANs (PGGANs), which is a new GAN training scheme developed in the computer vision community that supports improved training stability and the ability to capture spatially fine features from a high-resolution training set.[27] Second, we progressively augment the training set by identifying high-performance GAN-generated metasurfaces using an electromagnetic simulator and adding them to the training set, while removing relatively low-performance devices. Third, we incorporate self-attention layers into the network, which learn long-range spatial trends within images.[26,28] These layers are complementary to convolutional layers, which are better at learning local regional trends within images. This study demonstrates the ability of neural networks to learn highly complex and nuanced spatial features from an ensemble of devices.

PROBLEM SETUP AND OVERVIEW OF NETWORK TRAINING

We focus on designing topologically complex metagratings that deflect normally incident TE-polarized light to the +1 diffraction order for a range of outgoing angles (35−85°) and operating wavelengths (500−1300 nm). A schematic of the device is summarized in Figure 1b. The metagratings comprise 325 nm thick polycrystalline silicon on an SiO_2 substrate, and devices are represented as images of single grating periods with dimensions of 64 × 128 pixels. Each pixel in the image has a value of either 0 or 1, which represents the refractive index of air or polycrystalline silicon, respectively. Mirror symmetry along the y-direction is enforced to simplify the design space. The deflection efficiency is defined as the intensity of light deflected to the desired angle, θ, and is normalized to the incident light intensity in glass.

Prior to PGGAN training, we create a training set containing 600 high-efficiency metasurfaces, each produced by performing 350 iterations of gradient-based topology optimization on an initially random dielectric distribution. These devices sparsely

https://dx.doi.org/10.1021/acsphotonics.0c00539
ACS Photonics XXXX, XXX, XXX−XXX

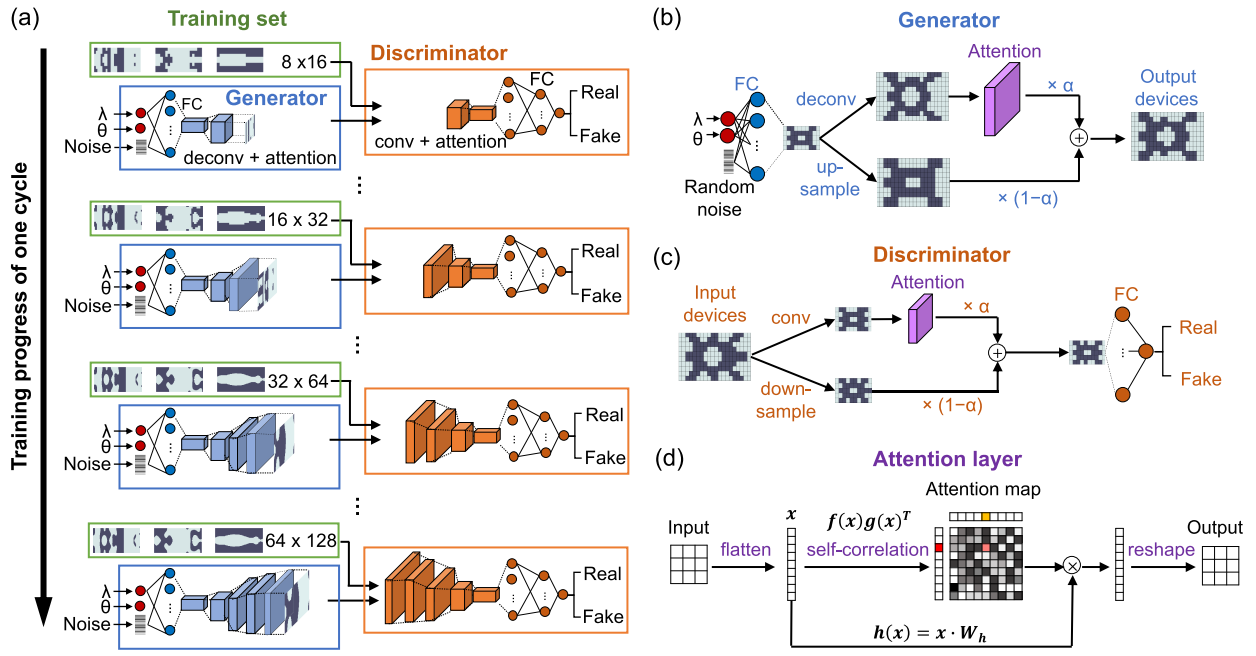


Figure 2. PGGAN network architecture and training process. (a) Schematic of a single cycle of the network training process in which the generative model, discriminative model, and network resolution progressively grow. Training starts by processing training set devices with a resolution of 8 × 16 pixels, downsampled from full resolution devices of 64 × 128 pixels. The device and network resolution gradually increase to the full resolution of the original training set. (b,c) Transition process for the generator and discriminator as they increase in spatial resolution. In (b,c), the impact of newly added deconvolutional and convolutional layers, respectively, is progressively increased in the network architecture by linearly tuning the weight α from 0 to 1. (d) Schematic of an attention layer. The attention layer first learns an attention map that measures the correlation between every two pixels within the input feature map and then aggregates the input feature map by multiplying this attention map.

sample the design parameter space: the training set devices sample the wavelength space in increments of 200 nm and the deflection angle space in increments of 10°. To evaluate the performance of the training set devices and those generated by the GAN, we use the Rigorous Coupled-Wave Analysis (RCWA)[29] electromagnetic simulator to compute the efficiency of geometrically eroded, ideal, and dilated devices. Our performance metric is defined to be the weighted efficiency of these three device variants with weights of 0.25, 0.5, and 0.25, respectively, so that both efficiency and robustness to fabrication imperfections are evaluated.[24]

Our PGGAN comprises two neural networks, a generator and discriminator. The inputs to the generator include the operating wavelength λ, deflection angle θ, and an eight-dimensional uniformly distributed random noise vector $\mathbf{z}$, and its output is images of the device layout. Given a distribution of noise values as inputs, the outputs are a diverse distribution of device layouts. The discriminator is a classifier that attempts to distinguish whether a presented input image is from the generator or the training set. Architecturally, the generator contains fully connected layers followed by deconvolution layers, while the discriminator contains convolution layers followed by fully connected layers. In addition, there are attention layers that follow each deconvolution and convolution layer.

The complete network training process involves training the PGGAN multiple times from scratch in a cyclic fashion. At the end of each cycle, the training set is augmented and becomes the initial training set for the new PGGAN in the following cycle. We require the PGGAN to be trained from scratch every time the training set is updated because it is the only way for the network to accurately learn the nuanced geometric details of the new training set.

The PGGAN training process in a single training cycle can be described as a competition between the generator and discriminator. The discriminator aims to successfully distinguish between generated and training set devices, while the generator aims to fool the discriminator by generating devices mimicking the training set. The generator and discriminator train through an iterative process in alternating steps, and upon training completion the final generator will have learned the underlying topological features from high-efficiency devices in the training set and be able to generate layouts with high-efficiency features. The details pertaining to the training procedure can be found in the Supporting Information.

Progressive Growth of the Network Architecture. To improve the stability of the training process and enhance the capabilities of both the generator and discriminator, we progressively grow the resolution of the PGGAN during each training cycle.[27] In this scheme, the PGGAN architecture operates with a low spatial resolution during initial training iterations and focuses on learning spatially coarse features from the training set. Additional neural layers are then progressively added, at which point the PGGAN focuses on learning finer-scale details from the training set. Given the success of these techniques to improve the generation of high-resolution images, such as the faces of people,[27] these networks serve as plausible candidate solutions to improving the generation of physical devices such as metasurfaces.

Schematics of the PGGAN architecture at different stages of network training are presented in Figure 2a. The initial network architecture processes devices with a resolution of only 8 × 16 pixels and trains from downsampled images of the training set. Our use of low-resolution devices and networks has three main implications. First, it allows the network to focus on learning the large-scale topological features from the

https://dx.doi.org/10.1021/acsphotonics.0c00539
ACS Photonics XXXX, XXX, XXX−XXX

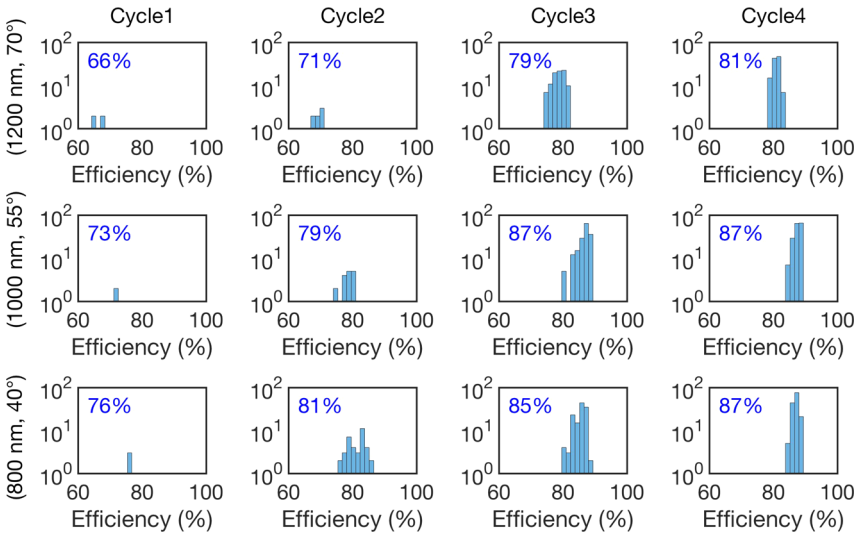


Figure 3. Evolution of weighted efficiency distributions from the training set over the course of the full cyclic network training process for three representative wavelength-deflection angle pairs: (800 nm, 40°), (1000 nm, 55°) and (1200 nm, 70°). These efficiency distributions are sampled over the course of four network training cycles. The average weighted efficiency of the training set for each wavelength-deflection angle pair is denoted in each plot.

training set. Second, it restricts the design space to a much smaller dimension, which improves the overall accuracy of the network training process itself. Third, training at lower spatial resolutions leads to dramatic reductions in computational cost.

Once the network has undergone sufficient training with these low-resolution devices, the network and device resolution increase to 16 × 32 pixels, and then later to 32 × 64 pixels and finally to 64 × 128 pixels. During each increase in resolution, an additional deconvolution and convolution layer is added to the generator and discriminator, respectively, each with dimensions matching the new device resolution. With these added layers, the PGGAN accurately learns and captures higher spatial resolution features in the training set.

During the transition moments in the training process when the spatial resolution of the network increases, care must be taken to ensure that this increase in spatial resolution does not destabilize the network. These instabilities can arise because the convolution and deconvolution layers that are added to the PGGAN during these moments are untrained and possess random weights. To address this issue, the impact of the added convolution and deconvolution layers are gradually incorporated into the PGGAN in a manner illustrated in Figure 2b,c. Consider the PGGAN generator. When a new deconvolution layer is added to the network, the device at the output of this layer X_{new} is the weighted superposition of two images. The first image is X_{deconv}, which is the device outputted from the prior deconvolution layer and processed by the new deconvolution layer. The second image is the device $X_{upsampled}$, which is the device outputted from the prior deconvolution layer and upsampled to match the spatial resolution of the new deconvolution layer. X_{new} relates to $X_{upsampled}$ and X_{deconv} with the following expression:

$$X_{new} = \alpha^* X_{deconv} + (1 - \alpha)^* X_{upsampled} \quad (1)$$

Over the course of five thousand network training iterations, α linearly increases from zero to one. Initially, α is zero and the untrained deconvolution layer does not contribute to the generated device pattern. As training progresses and α increases, this layer begins to properly learn and capture spatially fine device features, and it contributes more to X_{new}. After five thousand iterations, the generated device patterns are exclusively generated from the new deconvolution layer. Five thousand additional training iterations are performed after α is set to one to further stabilize the network, after which the network resolution is increased again and the process is repeated. The PGGAN discriminator evolves in a similar way, except that convolutional layers are progressively added to the network instead of deconvolutional layers. Image downsampling is performed using average pooling.

Self-Attention Mechanism. In both the generator and discriminator, we add a self-attention layer (Figure 2d) after each deconvolution and convolution layer. Self-attention layers enable the network to better learn global geometric trends[26,28,30] and are complementary concepts to convolutional layers, which are better at learning local geometric information. In the case of dog images, as an example, convolutional layers are well suited to learn the fur texture of dogs while self-attention layers are better at learning the global shape of dogs.[26] This multiscalar learning approach is important for metasurfaces, which strongly depend on both fine local features and global structural geometry.

We summarize the concept as follows. For each convolutional and deconvolutional layer output, we first flatten the outputted image into a vector $\mathbf{x}$. We then transform $\mathbf{x}$ into three distinct feature maps: $\mathbf{f}(\mathbf{x}) = \mathbf{x}\cdot\mathbf{W}_f$, $\mathbf{g}(\mathbf{x}) = \mathbf{x}\cdot\mathbf{W}_g$, and $\mathbf{h}(\mathbf{x}) = \mathbf{x}\cdot\mathbf{W}_h$, where $\mathbf{W}_f$, $\mathbf{W}_g$, and $\mathbf{W}_h$ are learnable weight matrices. By multiplying $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})^T$, we obtain an attention map that measures the correlation between every feature of $\mathbf{f}(\mathbf{x})$ to every feature of $\mathbf{g}(\mathbf{x})$. This map represents how much attention the model should pay to the ith feature when synthesizing the jth feature. The output of the attention layer is obtained by multiplying the attention map with the feature map $\mathbf{h}(\mathbf{x})$. As the network trains, the model will learn the dependencies and correlations between different device regions. The performance gain by adding attention layers is shown in Figure S2.

Progressive Growth of the Training Set. It is essential to improve the training set in order for the generator to consistently generate devices with greater geometric diversity

https://dx.doi.org/10.1021/acsphotonics.0c00539
ACS Photonics XXXX, XXX, XXX−XXX

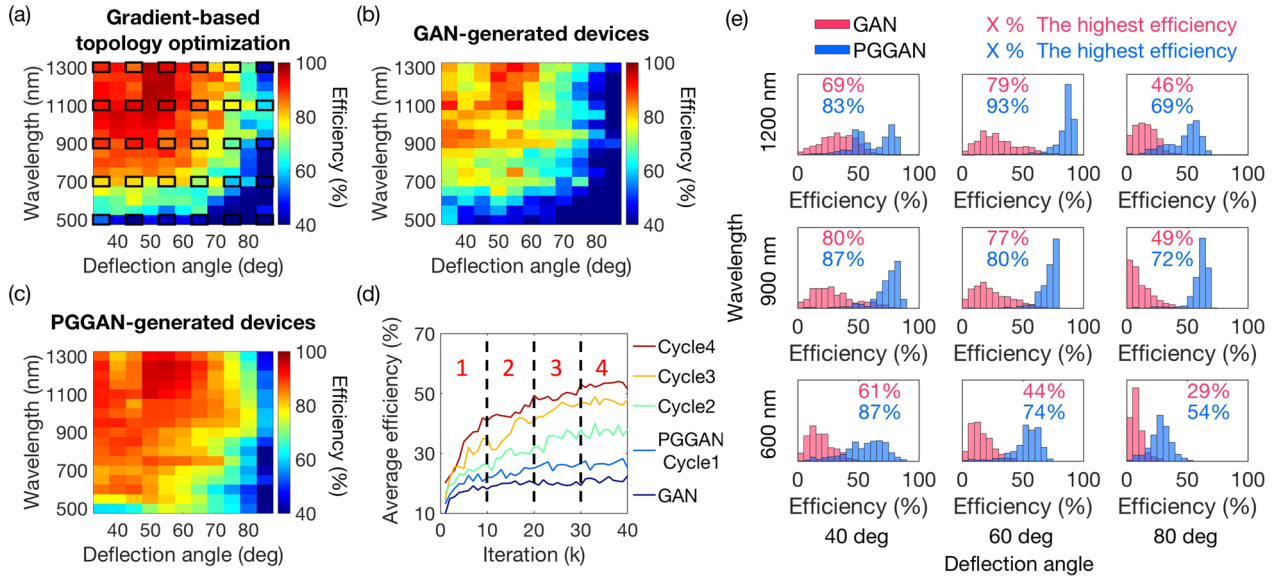


Figure 4. Summary of PGGAN performance. (a–c) Plot of the best devices, quantified by weighted efficiency, at each wavelength-deflection angle pair for devices designed using (a) gradient-based topology optimization, (b) basic GAN, and (c) PGGAN. The initial training set parameters used for the basic GAN and PGGAN are denoted by the solid black boxes in (a). (d) Average weighted efficiency of generated devices as a function of training iteration for the basic GAN (purple) and four PGGAN cycles (blue, green, yellow, and red for the first, second, third, and fourth cycles, respectively). The impact of network growth on network performance can be visualized in comparing the PGGAN cycle1 and basic GAN efficiency curves. The impact of training set augmentation on network performance can be visualized in comparing the PGGAN efficiency curves for different cycles. (e) Representative efficiency histograms of basic GAN-generated devices (red) and PGGAN-generated devices (blue). Five hundred devices are generated from both the basic GAN and PGGAN for each wavelength-deflection angle pair. The highest weighted efficiency value in each histogram are denoted.

and enhanced performance compared to the initial training set. We augment the training set at the end of each PGGAN training cycle by first generating an ensemble of devices and calculating the performance metric of each device. Generated devices with performance metrics higher than the average metric value in the training set are added to the training set, while devices in the training set with lower than threshold metric values are removed (see Supporting Information for more details). Our ability to improve the PGGAN using PGGAN-generated devices leads to a positive feedback loop between network and training set augmentation: improvements to the training set leads to a better GAN model, while improvements to the GAN model lead to enhanced generation of high-performance metasurfaces that are subsequently added to the training set. In this manner, the network is able to more effectively learn the desired features from the training set and also better explore and interpolate topological features in the design space.

Histograms of training set device efficiencies for three representative wavelength–deflection angle pairs, over the course of four cycles of network training, are summarized in Figure 3. As training progresses, more high-performance devices are added to the training set and both the quantity and average efficiency of the devices in the training set increases. The average weighted efficiency of the training set almost converges to an asymptotic value after the fourth training cycle.

RESULTS AND DISCUSSION

We produce final device designs from our fully trained PGGAN by generating two thousand devices for a given wavelength-deflection angle pair, evaluating the weight efficiencies of these devices using RCWA, and taking the highest efficiency device. We choose a discrete operating device parameter space that includes wavelengths ranging from 500 to 1300 nm in increments of 50 nm and deflection angles ranging from 35° to 85° in increments of 5°, though we note that the generative network can sample the input parameters in a continuous manner. As benchmarks, we also design devices in the following two ways: we design a total of 9350 devices using gradient-based topology optimization[31] (50 devices for each wavelength-angle pair) and select the best device for a given wavelength-deflection angle pair, and we train a basic GAN without progressive growth and filter for high-performance generated devices in the same manner above. Plots that summarize the efficiencies of devices designed using these three methods are shown in Figure 4a–c.

A comparison between Figure 4a,c shows that the best devices generated by the PGGAN have efficiencies similar to or even better than those produced from gradient-based topology optimization. Statistically, the best PGGAN-generated devices outperform the best device designed using gradient-based topology optimization for 53% of the wavelength-deflection angle pairs. Furthermore, when averaging all of the efficiency values from the plots in Figure 4a,c, the average efficiency value from the PGGAN is 3% higher than that from gradient-based topology optimization. The high efficiencies for devices generated at all wavelength–deflection angle pairs, including those not covered in the original training set, indicate that our PGGAN design strategy can properly generalize across the full wavelength and deflection angle parameter space without significant overfitting. An examination of the best generated devices as a function of wavelength and angle is presented in the Supporting Information Figure S4 and displays a diversity of layouts, showing that the network does not collapse to a particular geometric output but is able to

https://dx.doi.org/10.1021/acsphotonics.0c00539
ACS Photonics XXXX, XXX, XXX–XXX

conditionally capture the nuanced geometric variations spanning the input parameter space.

A comparison between Figure 4b,c shows that the PGGAN outperforms the basic GAN by a wide margin, further demonstrating the effectiveness of the progressive growth strategy. For 96% of the wavelength-deflection angle pairs, the best PGGAN-generated devices outperform those generated from the basic GAN. This performance disparity can be further visualized by tracking device efficiencies over the course of network training for the PGGAN and basic GAN, which is summarized in Figure 4d. For the basic GAN, the average efficiency of the generated devices initially increases but plateaus after approximately 20 000 iterations. For the PGGAN, the average efficiency of the generated devices increases over the course of the full network training process, and upon training completion PGGAN-generated devices have an average efficiency that is 14% higher than those generated by the basic GAN. The superior performance of the PGGAN compared to the basic GAN is further enforced in Figure 4e, which shows efficiency histograms generated from the two methods. We find that the PGGAN consistently generates devices with efficiency distributions that have peak and average values that are higher than those from the basic GAN.

An estimation of the computational cost (in the Supporting Information) for the differing optimization methods featured in Figure 4a,c indicates that the PGGAN concept requires the less computational overhead than gradient-based topology optimization. For the gradient-based topology-optimized devices in Figure 4a, 19.6 M RCWA EM simulations are required. The PGGAN featured in Figure 4c requires the production of the initial training set, evaluation of generated devices during network training to grow the training set, and the evaluation of final PGGAN-generated devices. In total, 7.0 M RCWA EM simulations are required. Furthermore, most of the computational cost with the PGGAN is a one-time cost incurred during network training, such that a fully trained network only requires the evaluation of generated devices to produce high-performance structures and does not require any refinement steps.

CONCLUSION

In summary, we show that a generative neural network based on the PGGAN architecture can learn to output high-efficiency, robust metasurfaces from a sparse training set. Compared to the basic GAN, our modified PGGAN utilizes three new principle features: progressive network growth, which enables more robust learning of local topological features from the training set; the self-attention mechanism, which allows the network to better learn global spatial correlations within device images; and progressive growth of the training set, from which high-efficiency topological features in the design space can be better captured. Our training results indicate that the PGGAN can generate devices that compare well with those generated from gradient-based topology optimization while incurring less computational cost.

Directions for future work include identifying methods to reduce the computational cost of the training set and further optimizing the PGGAN architecture and a resolution transition scheme. While we focus on metagratings as the model system for this proof-of-concept demonstration, it is clear that PGGAN's superior modeling capacity over GAN allows it to handle more complex parameter spaces for practical utility. We anticipate that our PGGAN can apply to aperiodic meta-atom structures with tailored dispersion and scattering properties. We also anticipate that the PGGAN approach can be widely utilized in other fields that require the design of high-resolution, high-dimensional layouts, particularly in domains where simulators can quantitatively evaluate the performance of generated layouts.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsphotonics.0c00539.

Protocol of PGGAN training set growth; impact of the self-attention mechanism; computational cost calculation; images of generated devices; comparison with the stretched training set; benchmark of RCWA simulation with FDTD simulation; Figures S1 to S5 and Tables S1 to S3 (PDF)

AUTHOR INFORMATION

Corresponding Author

Jonathan A. Fan − *Department of Electrical Engineering, Stanford University, Stanford, California 94305, United States;* orcid.org/0000-0001-9816-9979; Email: jonfan@stanford.edu

Authors

Fufang Wen − *Department of Electrical Engineering, Columbia University, New York, New York 10027, United States*

Jiaqi Jiang − *Department of Electrical Engineering, Stanford University, Stanford, California 94305, United States;* orcid.org/0000-0001-7502-0872

Complete contact information is available at: https://pubs.acs.org/10.1021/acsphotonics.0c00539

Author Contributions

[§]F.W. and J.J. contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The simulations were performed in the Sherlock computing cluster at Stanford University. This work was supported by the U.S. Air Force under Award Number FA9550-18-1-0070, the Sony Research Award, and the David and Lucile Packard Foundation Fellowship.

REFERENCES

(1) Yu, N.; Capasso, F. Flat Optics with Designer Metasurfaces. *Nat. Mater.* **2014**, *13*, 139.

(2) Kildishev, A. V.; Boltasseva, A.; Shalaev, V. M. Planar Photonics with Metasurfaces. *Science* **2013**, *339*, 1232009.

(3) Kuznetsov, A. I.; Miroshnichenko, A. E.; Brongersma, M. L.; Kivshar, Y. S.; Luk'yanchuk, B. Optically Resonant Dielectric Nanostructures. *Science* **2016**, *354*, aag2472.

(4) Colburn, S.; Zhan, A.; Majumdar, A. Metasurface Optics for Full-Color Computational Imaging. *Sci. Adv.* **2018**, *4*, No. eaar2114.

(5) Tittl, A.; Leitis, A.; Liu, M.; Yesilkoy, F.; Choi, D.-Y.; Neshev, D. N.; Kivshar, Y. S.; Altug, H. Imaging-Based Molecular Barcoding with Pixelated Dielectric Metasurfaces. *Science* **2018**, *360*, 1105−1109.

(6) Arbabi, A.; Horie, Y.; Bagheri, M.; Faraon, A. Dielectric Metasurfaces for Complete Control of Phase and Polarization with Subwavelength Spatial Resolution and High Transmission. *Nat. Nanotechnol.* **2015**, *10*, 937−943.

(7) Zheng, G.; Mühlenbernd, H.; Kenney, M.; Li, G.; Zentgraf, T.; Zhang, S. Metasurface Holograms Reaching 80% Efficiency. *Nat. Nanotechnol.* **2015**, *10*, 308−312.
(8) Campbell, S. D.; Sell, D.; Jenkins, R. P.; Whiting, E. B.; Fan, J. A.; Werner, D. H. Review of Numerical Optimization Techniques for Meta-Device Design. *Opt. Mater. Express* **2019**, *9*, 1842−1863.
(9) Molesky, S.; Lin, Z.; Piggott, A. Y.; Jin, W.; Vucković, J.; Rodriguez, A. W. Inverse Design in Nanophotonics. *Nat. Photonics* **2018**, *12*, 659−670.
(10) Jensen, J. S.; Sigmund, O. Topology Optimization for Nano-Photonics. *Laser Photonics Rev.* **2011**, *5*, 308−321.
(11) Jafar-Zanjani, S.; Inampudi, S.; Mosallaei, H. Adaptive Genetic Algorithm for Optical Metasurfaces Design. *Sci. Rep.* **2018**, *8*, 11040.
(12) Sell, D.; Yang, J.; Wang, E. W.; Phan, T.; Doshay, S.; Fan, J. A. Ultra-High-Efficiency Anomalous Refraction with Dielectric Metasurfaces. *ACS Photonics* **2018**, *5*, 2402−2407.
(13) Yang, J.; Sell, D.; Fan, J. A. Freeform Metagratings Based on Complex Light Scattering Dynamics for Extreme, High Efficiency Beam Steering. *Ann. Phys.* **2018**, *530*, 1700302.
(14) Jiang, J.; Fan, J. Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network. *Nano Lett.* **2019**, *19*, 5366−5372.
(15) Jiang, J.; Fan, J. Simulator-Based Training of Generative Models for the Inverse Design of Metasurfaces. *Nanophotonics* **2019**, *9*, 1059.
(16) Sell, D.; Yang, J.; Doshay, S.; Fan, J. A. Periodic Dielectric Metasurfaces with High-Efficiency, Multiwavelength Functionalities. *Adv. Opt. Mater.* **2017**, *5*, 1700645.
(17) Lin, Z.; Groever, B.; Capasso, F.; Rodriguez, A. W.; Lončar, M. Topology-Optimized Multilayered Metaoptics. *Phys. Rev. Appl.* **2018**, *9*, 044030.
(18) Liu, D.; Tan, Y.; Khoram, E.; Yu, Z. Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures. *ACS Photonics* **2018**, *5*, 1365−1369.
(19) Inampudi, S.; Mosallaei, H. Neural Network Based Design of Metagratings. *Appl. Phys. Lett.* **2018**, *112*, 241102.
(20) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *Advances in Neural Information Processing Systems (NIPS 2014)*; Curran Associates, Inc.: Red Hook, New York, 2014; pp 2672−2680.
(21) Jiang, J.; Sell, D.; Hoyer, S.; Hickey, J.; Yang, J.; Fan, J. Free-Form Diffractive Metagrating Design Based on Generative Adversarial Networks. *ACS Nano* **2019**, *13*, 8872−8878.
(22) Liu, Z.; Zhu, D.; Rodrigues, S. P.; Lee, K.-T.; Cai, W. Generative Model for the Inverse Design of Metasurfaces. *Nano Lett.* **2018**, *18*, 6570−6576.
(23) So, S.; Rho, J. Designing Nanophotonic Structures Using Conditional Deep Convolutional Generative Adversarial Networks. *Nanophotonics* **2019**, *8*, 1255−1261.
(24) Wang, E.; Sell, D.; Phan, T.; Fan, J. A. Robust Design of Topology-optimized Metasurfaces. *Opt. Mater. Express* **2019**, *9*, 469.
(25) Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. *International Conference on Machine Learning (ICML 2017)*; PMLR: Sydney, Australia, August 6−11, **2017**; pp 2642−2651.
(26) Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-Attention Generative Adversarial Networks. *International Conference on Machine Learning (ICML 2019)*; PMLR: Long Beach, California, June 10−15, 2019; pp 354.
(27) Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations (ICLR 2018)*; Aug 6−11, 2017.
(28) Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. *Computer Vision and Pattern Recognition (CVPR 2018)*; IEEE Computer Society: Washington, D.C., June 18−22, 2018.
(29) Hugonin, J.; Lalanne, P. *Reticolo Software for Grating Analysis*; Institut d'Optique: Palaiseau, France, 2005.
(30) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all You Need. *Advances in Neural Information Processing Systems (NIPS 2017)*; Curran Associates, Inc.: Red Hook, New York, December 4−9, 2017; pp 5998−6008.
(31) Sell, D.; Yang, J.; Doshay, S.; Yang, R.; Fan, J. A. Large-Angle, Multifunctional Metagratings Based on Freeform Multimode Geometries. *Nano Lett.* **2017**, *17*, 3752−3757.

https://dx.doi.org/10.1021/acsphotonics.0c00539
ACS Photonics XXXX, XXX, XXX−XXX