Han Xiao (Orcid ID: 0000-0003-1728-3599)
Guo L. Jay (Orcid ID: 0000-0002-0347-6309)

Note: Reference count mismatch between author pdf and tex file. We followed tex file. Please take care.

Inverse Design of Metasurface Optical Filters using Deep Neural Network with High Degrees of Freedom

Xiao Han[1,2 †], Ziyang Fan[1,2 †], Zeyang Liu[1 †], Chao Li[1,2], L. Jay Guo[1]

[†]Equally contributing authors.

[1]Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109, USA

[2]Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

[*]Correspondence: L. Jay Guo, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109, USA. Email: guo@umich.edu

Abstract

In order to obtain a metasurface structure capable of filtering light of a specific wavelength range in the visible band, the traditional methods usually traverse the space consisting of possible designs, searching for a potentially satisfactory structure by performing iterative calculations to solve Maxwell's equations. In this paper, we propose a systematic method based on neural networks that can complete an inverse design process to solve the problem. Compared with the traditional methods, our method is much faster while competent to encompass a high degree of freedom to generate device structures, which can ensure that the spectra of generated structures resemble the desired ones.

Keywords: Metasurface Filter, High Degrees of Freedom, Neural Network, Visible Band

1 Introduction

Metasurfaces, which are constructed of 2D artificial material structures in the subwavelength scale, have received great attention due to its ability of unprecedented control over the intrinsic properties of light, including the amplitude[1, 2], phase[3, 4], polarization[5, 6] and the orbital angular momentum[7]. The most critical feature of metasurfaces is that the spatially varying patterns or material compositions provide high degrees of freedom in designing spatial inhomogeneity over an optically thin surface. The degree of freedom refers to the number of optimizable variables during the design process. A number of planar optics such as filters[1, 2], lenses[3, 4], polarizers[5, 6] and absorbers[8, 9] have been enabled by a variety of reflective or transmissive metasurfaces, featuring high optical performance as well as compact structures. Two central problems arise in the process of designing the metasurfaces. The first one is to obtain an accurate prediction of the optical spectrum for a given structure, named "Forward Simulation". This problem is generally addressed by solving Maxwell’s equations using different methods, including rigorous coupled wave analysis (RCWA), finite-difference time-domain method (FDTD), finite-element modeling (FEM) and so on. The second core problem is to find an optimal structure based on actual demands (e.g. desired optical responses), named "Inverse Design". Inverse design of photonic structures is conventionally demonstrated using adjoint sensitivity analysis[10–15]. However, these methods are still time-consuming and not universal in most cases, so that researchers have to find a near-optimal solution from a limited database, which contains a finite parameter space and corresponding simulated spectra.

In different fields, modern machine learning algorithms have shown advantages in a wide range of applications needing identifications and optimizations. Deep learning allows computational models composed of multiple processing layers to learn representations of data with different levels of abstraction. These methods have significantly improved the state-of-the-art in computer vision, natural language processing, speech signal processing and other applications[16]. Deep learning has also been successfully applied to conventional science and engineering fields outside of computer science, such as condensed matter[17], particle physics[18], chemical syntheses[19], microscopy[20] and proteomics[21, 22]. Furthermore, deep neural networks have also drawn interests from the optical community thanks to their robust fitting ability. Particularly, remarkable progresses based on deep learning techniques have been made in the inverse design of optical devices[23–33].

Feedforward neural network architecture is the typical and widely-used structure in most deep learning applications. To go from one layer to the next, a set of units compute a weighted sum of their inputs from the previous layer and pass the result through a non-linear function[16]. Fully connected layers, convolutional layers and transpose convolutional layers are the most basic components of the feedforward neural network. A fully connected network (FC), which is composed of several fully connected layers, is the simplest neural network structure and capable of handling one-dimensional vectors. A convolutional neural network (CNN) primarily consists of convolutional layers, which is often used in feature extraction tasks of multidimensional data. The transpose convolution can be considered as an upsampling process opposed to the convolution process, and the transpose convolutional layer is often embedded in networks associated with the tasks on generating patterns, such as generative adversarial networks (GAN)[34] and fully convolutional networks[35].

Optimization problems in the field of optics can be modeled in a simpler way by using several one-dimensional parameters, regardless of the number of dimensions. D. Liu *et al.* used a FC to learn electromagnetic scattering of alternating dielectric thin films with a combination of variable thicknesses and materials[23]. They proposed a tandem architecture combining forward simulation and inverse design together in pursuit of overcoming the issue of data inconsistency and slow training process, which has become a feasible architecture for solving similar problems. J. Peurifoy *et al.* adopted a FC having 4 hidden layers with 100 neurons in each to approximate light scattering of core-shell nanoparticles made of S_iO_2 and T_iO_2[26]. I. Malkiel *et al.* expounded the relationship between the spectral complexity and design feasibility, then provided a FC with around a dozen layers and multiple input entries. Their method can be applied to direct on-demand engineering of plasmonic structures and metasurfaces[25]. Tahersima *et al.* built a robust deeper network based on FC to inversely design integrated photonic devices, whose design space is considerably larger[28]. They also utilized intensity shortcut from deep residual networks (ResNet)[36] to allow smooth backward propagation of the gradients. Recently, the revised neural tensor network[37] adopted by S. An *et al.* overcame three key challenges that perplexed the previous neural-network-based (NN-based) design schemes: input/output vector dimensional mismatch, inaccurate EM-wave phase prediction, as well as the inability to adapt to 3-D dielectric structures[30]. Later on, S. An *et al.* proposed a NN-based simulator to model meta-atoms with free-form 2D patterns and different lattice sizes, material refractive indices and thicknesses, achieving competitive results for wide spectrum prediction in the microwave band[33].

Additionally, some methods with a higher degree of freedom that focus on multi-dimensional representations have been proposed to obtain the desired spectra. T. Asano *et al.* provided a four-layer neural network including a convolutional layer for the prediction of the quality factor in two dimensional photonic crystals[27]. Z. Liu *et al.* proposed an approach combining a GAN and a simulation neural network that can efficiently discover and optimize unit cell patterns of metasurfaces in response to user-defined, on-demand spectra at the input[29]. In order to automatically design and optimize three-dimensional chiral metamaterials, W. Ma *et al.* reported a multi-task model dividing the task into a primary task and an auxiliary one, which comprises two bidirectional neural networks assembled by a partial stacking strategy[24]. J. Jiang *et al.* showed that GAN can be trained from image-based, periodic and topology-optimized metagratings to produce high-efficiency, topologically complex devices, which can perform over a broad range of deflection angles and wavelengths[32]. J. Jiang *et al.* also transformed a GAN into a global optimizer by replacing the traditional discriminator by an adjoint-based optimization algorithm followed by gradient estimation method for backpropagation, then incorporated them as a physics-driven, data-free neural network[31].

In this paper, we primarily focus on the design of filters in the visible band based on 2D periodic structures, as shown in Figure 1a. The parameters of the device structure involving the choice of materials, shapes, unit cell size and layered permutations[23] offer various design freedom. Under the same output requirements as well as the same input constraints, a higher degree of freedom in terms of the unit cell structure enhances the probability of generating a desired spectrum. Enlightened by previous research[25, 29, 30], we improve the degree of freedom significantly by describing the shape of the structure in a unit cell as an arbitrary binary pattern, rather than a regular one (circle, rectangle and so on). However, both the requirements on network performance and the computational cost grow at the same time, as illustrated in Figure 1b. Our target is to implement the inverse design by maintaining a higher degree of freedom and less computational cost. For this purpose, we exploit two dedicated networks working in series, which are referred as the simulator and the generator. The previous works typically deal with a wide range of wavelength to obtain devices having satisfactory spectra[23, 25, 29, 30]. In contrast, we focus on the visible band which has a remarkable narrower range of wavelength. In this case, previous methods are prone to miss the optimal solution due to the limited searching space provided to the network. Moreover, in many applications of inverse design, how to correctly describe human demands without knowing the exact value of each point of the desired spectrum is an inevitable problem. For instance, during the inverse design of the metasurface filter, the position, depth and width of the peaks/valleys in the spectra are the primary considerations, while the value of each point of the desired spectrum is of less interest and even may not be known precisely. Input spectra represented by mathematical functions have tendency to lead to unsatisfactory results, because most of these spectra are idealized so that cannot be generated in the limited given space. Thus, the input should better reflect our subjective requirements so as to narrow the gap between practical conditions and human demands. To solve this problem, we propose a novel encoding method using a contrast vector extracted from the desired spectrum to reflect the key features of filter. The qualitative comparison of the traditional approach and our proposed method is shown in Figure 1c. To address the issue of very limited data for training the network, we use RCWA to generate training data and make it as the benchmark for comparing the performance of our NN-based simulator. RCWA is an exact solution to Maxwell's equations for the diffraction of electromagnetic waves by the grating structures, which is a relatively straightforward, noniterative, deterministic technique[38, 39]. Once sufficient training data are obtained, our simulator can replace RCWA to an acceptable level in the training process, yielding higher efficiency in tradeoff of accuracy.

2 Methods

Our purpose is to acquire an optimized structure having the specified response in the visible band, which impels us to expand the search space by increasing the degrees of freedom, such as the complexity of the pattern within the unit cell. In our approach, a generative model named generator is realized elaborately to distill the input spectrum as guidance information to generate a structure satisfying the expected response. Additionally, we define contrast vectors as well as another neural network named simulator to educate the generator. The simulator extracts information from the input structure and then produces regression estimation of the spectrum. These two neural networks can respectively solve forward simulation and inverse design problems efficiently with minor error in the design space for arbitrary input.

The physical structure of the metasurface that can realize the spectrum filtering function consists of 2D periodic patterns of polycrystalline silicon (Poly-Si) (with a fixed thickness of 500 nm) on a silica substrate. Poly-Si is chosen to take the advantage of its high refractive index, and hence strong scattering efficiency. Considering prior knowledge and actual fabrication requirements, the period of the 2D patterns ranges from 200nm to 400nm, and the shape of the pattern in a unit cell is described by a 64 × 64 pixelated binary image. Besides, 29 points are used to quantify a transmittance spectrum where the wavelength ranges from 400nm to 680nm. The direction of incident light is perpendicular to the metasurface. These limitations do not affect the universality of our method, which will be discussed later. For the sake of convenience, each pair of TE and TM responses at 400-680nm, represented by $\boldsymbol{T_{TE}}$ and $\boldsymbol{T_{TM}}$ with 29 points respectively, can de spliced into a spectrum $\boldsymbol{T} = \{t_1, t_2, \cdots, t_{58}\}$. Therefore, the problem is formulized as following: given a specified $\boldsymbol{T}$, how to use algorithms to generate a structure described by a binary image $\boldsymbol{S}$ and a period P, whose response $\hat{\boldsymbol{T}}$ is $\arg\min_{\hat{\boldsymbol{T}}} Distance(\boldsymbol{T}, \hat{\boldsymbol{T}})$. In other words, the generated structure is required to have the response $\hat{\boldsymbol{T}}$ that makes the similarity between $\boldsymbol{T}$ and $\hat{\boldsymbol{T}}$ maximum.

To avoid missing the optimal solution due to the limited wavelength range, we use a novel encoding method to extract the spectrum information which benefits network training as well. We define the contrast of a certain range in a spectrum as the ratio of the maximum transmittance within this range to the maximum value outside this range. After that, a contrast vector of the particular spectrum can be obtained by sequentially concatenating contrast values of different ranges together. As shown in Figure 2a, the contrast vectors defined in this way highlight the peaks and valleys in the spectra, while ignore minor fluctuations in the spectra, thus helping the network emphasize the filter property. In essence, this approach weakens the strong correlation between the expected spectrum and the network input in an intelligent way, such that a great quantity of spectra can correspond to one definite network input. By this approach, for a given spectrum $\boldsymbol{T_{TE}}$ for TE polarized light, let c_i be the value of the ith contrast, contrast vector $\boldsymbol{C_{TE}}$ is obtained according to the following algorithm 1. Then we calculate $\boldsymbol{C_{TM}}$ similarly and splice $\boldsymbol{C_{TE}}$, $\boldsymbol{C_{TM}}$ in the same way that splices $\boldsymbol{T_{TE}}$, $\boldsymbol{T_{TM}}$ to get $\boldsymbol{C}$.

Algorithm 1 Calculating contrast vector according to the transmittance spectrum

```
Data $\boldsymbol{T_{TE}} = \{t_1, t_2, \cdots, t_{29}\}$
Result $\boldsymbol{C_{TE}}$
$\Gamma = \{1, 2, \cdots, 7\}$
$\Theta = \{1, 2, \cdots, 29\}$
for $i \in \Gamma$ do
    $\Omega = \{4i - 3, 4i - 2, \cdots, 4i + 1\}$
    $max_{in} = \max(t_k), k \in \Omega$
    $max_{out} = \max(t_l), l \notin \Omega \wedge l \in \Theta$
    $c_i = \frac{max_{in}}{max_{out}}$
end
$\boldsymbol{C_{TE}} = \{c_1, c_2, \cdots, c_7\}$
```

Referring to the network structure of deep convolutional GAN (DCGAN)[40], the conditional inputs of conditional GAN (cGAN)[41] and the shortcut of residual network (ResNet)[36], the generator can turn a noise input (random seed for generation) into a qualified structure (i.e. a 2D binary image with an integer) according to different conditions (e.g. desired spectrum or desired contrast vector). Pytorch[42] is chosen as our deep learning framework. In order to simplify the training process, we use the structural similarity index (SSIM)[43] to evaluate the similarity between two images instead of a discriminator. As illustrated in Figure 2b, we transform the routine GAN training into a supervised and non-alternating one. Detailed network configurations and training methods are presented in the first section of the Supporting Information.

The simulator is trained by the data produced by RCWA[44]. Aiming at better performance and higher generalizability, we implement a data augmentation for more available spectra. This data augmentation is completed by rotating the shapes by 90, 180 and 270 degrees, respectively, following by exchanging the TE and TM responses. The training data set contains the shapes of the patterns in each unit cell, the periods, as well as the corresponding spectra. In other words, the inputs of the simulator are the period *P*, the shape ***S***, and the output is the predicted spectrum $\hat{\boldsymbol{T}}$. The loss function is the mean square error (MSE) between the predicted spectrum $\hat{\boldsymbol{T}}$ and real spectrum ***T*** as below, where *N* equals to 58.

$$SimulatorLoss = MSE(\hat{\boldsymbol{T}}, \boldsymbol{T}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{t}_i - t_i)^2 \qquad (1)$$

The generator produces the patterns based on the desired spectrum. The inputs of the generator are a contrast vector ***C*** (calculated from the desired spectrum) and a random noise ***Z***. The outputs are the generated shape $\hat{\boldsymbol{S}}$ and the generated period $\hat{P}$. It is worth noting that $\hat{\boldsymbol{S}}$ must be refined into a binary image, where 0 and 1 represent air and Poly-Si respectively in the RCWA code. Following the generator, $\hat{\boldsymbol{S}}$ and $\hat{P}$ are the inputs of the simulator, which can predict the spectrum of the generated device and guide the generator accordingly. Data augmentation is not applied in the training process of the generator, because it can bring difficulty to the convergence of the generator[23]. Spectrum loss is the MSE between the real spectrum ***T*** from training data and the simulated spectrum $\hat{\boldsymbol{T}}$ from simulator; the shape loss is taken to be the SSIM between the real shape ***S*** and the generated shape $\hat{\boldsymbol{S}}$; the period loss is the MSE between the real period *P* and the generated period $\hat{P}$. The generator loss comprises these three parts, as shown below. *α* and *β* are two hyperparameters describing the relative importance of the three.

$$\begin{aligned} GeneratorLoss &= SpectrumLoss + \alpha \cdot ShapeLoss + \beta \cdot PeriodLoss \\ &= \frac{1}{N}\sum_{i=1}^{N}(\hat{t}_i - t_i)^2 + \alpha \cdot SSIM(\hat{\mathbf{S}}, \mathbf{S}) + \beta \cdot (\hat{P} - P)^2 \end{aligned} \qquad (2)$$

If we feed the shape as a 2D pixel array and the period as an integer, the simulator architecture becomes complex to handle two data structures with different dimensions. Thus, the period is duplicated and expanded to a 2D array in the same size as the input binary image, then concatenated with the counterpart of the image after one respective convolution operation to constitute multiple channels, which provide convenience to the simultaneous convolution. In terms of the generator, to help it extract higher-dimensional features, the noise vector, as well as the contrast vector, are also expanded and concatenated. Inside the generator, the shape is produced by the transpose convolutional layers first, after which the period is generated by the fully connected layers. Our generator is not designed to obtain the shape and period simultaneously with a single network module, because obtaining these two properties of the metasurface involves two different tasks, i.e. generation and regression.

Thanks to the initial network that can map the random noise ***Z*** into the full design space, the probability of finding the optimal solution is enhanced. For our metasurface filter, the 1D parameter considered is the period of the unit cell, but it can generally involve any combination of design parameters in the design problems including the structure thickness, refractive index, or the polarization of light, etc.

3 Results and Discussion

To evaluate the performance of the simulator, we feed it with randomly generated polygons. The real spectra from validation set and the predicted spectra are plotted in Figure 3a. Statistically, the average error rate for each point between the real spectrum and the generated spectrum is 4% to 5% when we feed 5,200 paired shapes and periods (quadrupled by data augmentation) selected from the validation set. We also experiment the time efficiency to demonstrate the computational advantage of our simulator and summarize in Table 1. On average, for simulation of the single structure, the time consumed by a simulator with the same CPU is about 5×10^{-4} of that of RCWA, and this trend will be reinforced significantly with the increase in batch size and the involvement of GPU. As for the generator, we feed it with contrast vectors calculated from the real spectra. The real and generated shapes, as well as the periods, are shown in Figure 3b. Likewise, the average error rate for each point between the desired spectrum and the simulated spectrum of the generated structure is approximately 5% when we test the validation set containing 1,300 real spectra (without data augmentation). All the simulated results of the generator are conducted by RCWA rather than NN-based simulator in order to measure the performance of the generator more precisely.

To illustrate the universality and superiority of our method when the precise value of each point of desired spectrum is unknown, we also carry out several comparative experiments. We take whether the contrast is high/low enough in the range of interest (575-625nm in this case) as the evaluation indicator, because a high/low contrast can represent a good filter in a specific range. Firstly, we use an inverse gaussian function (mean=600, variance=40, amplitude=0.9) as the desired spectrum ***T*** (representing a reflective filter in the yellow band) to express our idealized demand, which cannot be realized in the design space due to the loss of Poly-Si for blue light. The spectrum and the corresponding device pattern found by traversing the whole data set with the smallest MSE are shown in Figure 4a, which are on behalf of the result of traditional method. To test the effectiveness of using contrast vectors as the input, we firstly acquire another benchmark generator trained by spectra while fixing all other training hyperparameters. Then we feed in the same gaussian-like desired spectrum ***T*** for this benchmark generator. The generated device and the corresponding simulated spectrum are shown in Figure 4b, which shows that idealized input for NN can lead to a disappointing result worse than that of traversing method. In the following step, we calculate a contrast vector $\boldsymbol{C}$ converted from $\boldsymbol{T}$ and feed it into another generator trained by contrast vectors. The generated device and the spectrum are plotted in Figure 4c. Thanks to the ability of the contrast vectors to describe subjective demands, we can see that they alleviate the strict limitation brought by the idealized input to some extent, thus helping the model to generate better results. However, in order to meet practical conditions based on the property of the material, we adjust $\boldsymbol{C}$ slightly by applying minor jitter to every single value while maintaining the overall tendency of the previous vector. This contrast vector $\boldsymbol{C_a}$ is not converted from mathematical functions and is designed to facilitate finding a spectrum that better meets subjective demands. Slight jitters will not seriously affect the nature of the filter, because we only focus on the position of the peak/valley and the contrast. We name the contrast vectors obtained by this method as artificially designed contrast vectors. Figure 4d is the result obtained by feeding $\boldsymbol{C_a}$ to the generator. This experiment illustrates that the valley of simulated spectra has a low contrast in the centre and matches the desired valley as well, thereby demonstrates the proposed method has the ability to adapt to a more subjective input. Consequently, the contrast vectors can circumvent the problem brought by the unknown of the value of each spectrum point.

We need to point out again that such an idealized spectrum $\boldsymbol{T}$ does not exist in our possible design space, according to Figure 6, due to the high loss of Poly-Si at the blue wavelength range. That is why results in Figure 4b-4d are not as satisfactory as those in Figure 3b. An interesting phenomenon is the generated shape may have completely different style from the training data. For example, in Figure 4c, the generated shape not only has one part in the middle but also another isolated part in the upper left corner, although all shapes in the training set are single polygons. This example shows a fact: if the desired spectrum cannot be realized by a physical structure within the design space, the network understands the mapping between structures and spectra during the training process and will not give false positive (FP) results. In contrast, if the input spectrum can be physically realized under the given restrictions, our model can provide a realizable device whose simulated spectrum matches fairly well with the desired spectrum. Figure 5 presents more examples about the generated devices and their simulated spectra with artificially designed contrast vectors as inputs. They are meant to show examples of producing the band pass (reflective) or band notch (transmissive) filters in different color ranges.

As mentioned above, we transform the generated shape into a binary image in the process of testing but not in training, since the mandatory binarization essentially yields a piecewise constant function in the last layer. Binarization for the generated shape makes the gradients too steep, which is pointless in the training process. A previous work trained a simulator with a noise similar to the generated patterns to circumvent the binarization and smoothing during the predicting process[29]. It is less efficient but can ensure that the input image of the simulator is binary and not too complex to fabricate. We purposely conducted one more study to find out how the binarization post-process affects our results. We feed a generated non-binary image and its binary version to the simulator respectively. The spectra produced by these two images in Figure 7 are both very similar to the ground truth given by the RCWA simulation. Statistically, after training the generator, the mean Manhattan distance between ground truth shapes and generated shapes is approximately 0.01. In other words, the intensity of each pixel ranges from 0 to 0.01 or 0.99 to 1 in average. Thus, the inaccurate input brought by the training process without mandatory binarization can be ignored for the frozen simulator.

Since SSIM is utilized in the training process, an unsupervised learning task is changed into a supervised one labeled with shape $\boldsymbol{S}$. As shown in Figure 8, SSIM influences both the appearance and the image contrast of the shape generated by the generator, so including SSIM helps to expedite the network convergence towards a specific direction. Considering that one spectrum can correspond to multiple structures when the generator is in different epochs, it is likely that different shapes can be generated for the same input spectrum. Without SSIM, the same loss from distinct shapes will be given to the network in such case, so that the network will suffer from this ambiguity leading to difficult convergence.

4 Conclusions

In summary, taking the metasurface filter design as an example, we propose a systematic method based on NN and contrast vectors to achieve inverse design with high degrees of freedom. For the "Forward Simulation" problem, when the required precision (error tolerance) is greater than the average error of the NN model, our simulator can substitute the traditional numerical electromagnetic simulation method with a great improvement in efficiency. For the "Inverse Design" problem, in the given design space which includes the arbitrary shapes and the periods within a specific range, our generator can generate a roughly optimal structure for the desired spectrum. Generally, the advantage of the proposed NN framework is its competence to generate devices with high degrees of freedom by means of optimizing the 2D shape and the 1D parameter of the unit cell in a synchronous manner. Compared with the traditional traversing method, ours can generate a diversity of device structures while ensuring speed and accuracy. The methodology presented here can be used in other applications to obtain desired reflection and transmission, which is essential to the metasurface design. Our models can also be improved and simplified to adapt to problems where multiple 1D parameters work together.

In our future work, we intend to utilize semi-supervised methods with prior knowledge in photonics to decrease the demands on data. The architecture of neural networks can also be further perfected and simplified because we observe that the gradients disappear under certain circumstances. Other possible extensions are worth further explorations, such as applying our models in other frequency bands and using more descriptive ways to improve or replace contrast vectors. Improvement on network performance can also be implemented by the participation of other deep learning methods, such as adopting the recurrent neural networks used in natural language processing to extract sequential information from the spectrum, imitating the attention mechanism in computer vision to make the process of binary image generation more explicable.

References

[1] Cheng F, Gao J, Luk TS, Yang X. Structural color printing based on plasmonic metasurfaces of perfect light absorption. Scientific reports 2015;5:11045.

[2] Proust J, Bedu F, Gallas B, Ozerov I, Bonod N. All-dielectric colored metasurfaces with silicon Mie resonators. Acs Nano 2016;10(8):7761–7767.

[3] Khorasaninejad M, Chen WT, Devlin RC, Oh J, Zhu AY, Capasso F. Metalenses at visible wavelengths: Diffraction-limited focusing and subwavelength resolution imaging. Science 2016;352(6290):1190–1194.

[4] Khorasaninejad M, Capasso F. Metalenses: Versatile multifunctional photonic components. Science 2017;358(6367):eaam8100.

[5] Yu N, Aieta F, Genevet P, Kats MA, Gaburro Z, Capasso F. A broadband, background-free quarter-wave plate based on plasmonic metasurfaces. Nano letters 2012;12(12):6328–6333.

[6] Guo T, Argyropoulos C. Broadband polarizers based on graphene metasurfaces. Optics letters 2016;41(23):5592–5595.

[7] Devlin RC, Ambrosio A, Rubin NA, Mueller JB, Capasso F. Arbitrary spin-to–orbital angular momentum conversion of light. Science 2017;358(6365):896–901.

[8] Liu X, Fan K, Shadrivov IV, Padilla WJ. Experimental realization of a terahertz all-dielectric metasurface absorber. Optics express 2017;25(1):191–201.

[9] Azad AK, Kort-Kamp WJ, Sykora M, Weisse-Bernstein NR, Luk TS, Taylor AJ, et al. Metasurface broadband solar absorber. Scientific reports 2016;6:20347.

[10] Piggott AY, Lu J, Lagoudakis KG, Petykiewicz J, Babinec TM, Vučković J. Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. Nature Photonics 2015;9(6):374.

[11] Piggott AY, Petykiewicz J, Su L, Vučković J. Fabrication-constrained nanophotonic inverse design. Scientific reports 2017;7(1):1786.

[12] Frandsen LH, Sigmund O. Inverse design engineering of all-silicon polarization beam splitters. In: Photonic and Phononic Properties of Engineered Nanostructures VI, vol. 9756 International Society for Optics and Photonics; 2016. p. 97560Y.

[13] Cao Y, Li S, Petzold L, Serban R. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. SIAM journal on scientific computing 2003;24(3):1076–1089.

[14] Phan T, Sell D, Wang EW, Doshay S, Edee K, Yang J, et al. High-efficiency, large-area, topology-optimized metasurfaces. Light: Science & Applications 2019;8(1):48.

[15] Molesky S, Lin Z, Piggott AY, Jin W, Vucković J, Rodriguez AW. Inverse design in nanophotonics. Nature Photonics 2018;12(11):659–670.

[16] LeCun Y, Bengio Y, Hinton G. Deep learning. nature 2015;521(7553):436–444.

[17] Carrasquilla J, Melko RG. Machine learning phases of matter. Nature Physics 2017;13(5):431.

[18] Baldi P, Sadowski P, Whiteson D. Searching for exotic particles in high-energy physics with deep learning. Nature communications 2014;5:4308.

[19] Segler MH, Preuss M, Waller MP. Planning chemical syntheses with deep neural networks and symbolic AI. Nature 2018;555(7698):604.

[20] Chen CL, Mahjoubfar A, Tai LC, Blaby IK, Huang A, Niazi KR, et al. Deep learning in label-free cell classification. Scientific reports 2016;6:21471.

[21] Zhou XX, Zeng WF, Chi H, Luo C, Liu C, Zhan J, et al. pdeep: Predicting MS/MS spectra of peptides with deep learning. Analytical chemistry 2017;89(23):12690–12697.

[22] Gessulat S, Schmidt T, Zolg DP, Samaras P, Schnatbaum K, Zerweck J, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. Nature methods 2019;16(6):509.

[23] Liu D, Tan Y, Khoram E, Yu Z. Training deep neural networks for the inverse design of nanophotonic structures. ACS Photonics 2018;5(4):1365–1369.

[24] Ma W, Cheng F, Liu Y. Deep-learning-enabled on-demand design of chiral metamaterials. ACS nano 2018;12(6):6326–6334.

[25] Malkiel I, Mrejen M, Nagler A, Arieli U, Wolf L, Suchowski H. Plasmonic nanostructure design and characterization via deep learning. Light: Science & Applications 2018;7(1):60.

[26] Peurifoy J, Shen Y, Jing L, Yang Y, Cano-Renteria F, DeLacy BG, et al. Nanophotonic particle simulation and inverse design using artificial neural networks. Science advances 2018;4(6):eaar4206.

[27] Asano T, Noda S. Optimization of photonic crystal nanocavities based on deep learning. Optics express 2018;26(25):32704–32717.

[28] Tahersima MH, Kojima K, Koike-Akino T, Jha D, Wang B, Lin C, et al. Deep neural network inverse design of integrated nanophotonic devices. arXiv preprint arXiv:180903555 2018;.

[29] Liu Z, Zhu D, Rodrigues SP, Lee KT, Cai W. Generative model for the inverse design of metasurfaces. Nano letters 2018;18(10):6570–6576.

[30] An S, Fowler C, Zheng B, Shalaginov MY, Tang H, Li H, et al. A Deep Learning Approach for Objective-Driven All-Dielectric Metasurface Design. ACS Photonics 2019;6(12):3196–3207.

[31] Jiang J, Fan JA. Global optimization of dielectric metasurfaces using a physics-driven neural network. Nano letters 2019;19(8):5366–5372.

[32] Jiang J, Sell D, Hoyer S, Hickey J, Yang J, Fan JA. Free-form diffractive metagrating design based on generative adversarial networks. ACS nano 2019;13(8):8872–8878.

[33] An S, Zheng B, Shalaginov MY, Tang H, Li H, Zhou L, et al. A Freeform Dielectric Metasurface Modeling Approach Based on Deep Neural Networks. arXiv preprint arXiv:200100121 2020;.

[34] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems; 2014. p. 2672–2680.

[35] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–3440.

[36] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

[37] Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems; 2013. p. 926–934.

[38] Moharam M, Grann EB, Pommet DA, Gaylord T. Formulation for stable and efficient implementation of the rigorous coupled-wave analysis of binary gratings. JOSA a 1995;12(5):1068–1076.

[39] Moharam M, Pommet DA, Grann EB, Gaylord T. Stable implementation of the rigorous coupled-wave analysis for surface-relief gratings: enhanced transmittance matrix approach. JOSA A 1995;12(5):1077–1086.

[40] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:151106434 2015;.

[41] Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint arXiv:14111784 2014;.

[42] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in pytorch 2017;.

[43] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, et al. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 2004;13(4):600–612.

[44] Hugonin JP, Lalanne P, RETICOLO: Rigorous Coupled Wave Analysis for gratings with Matlab interface; 2005. https://www.lp2n.institutoptique.fr/Membres-Services/Responsables-d-equipe/LALANNE-Philippe.

Figure 1 (a) Schematic diagram of the structural color filters based on 2D grating. (b) The tradeoff between the degree of freedom and computational cost for different methods. (c) The comparisons between the three methods discussed in this paper.

Figure 2 (a) Schematic of constructing contrast vector. (b) General scheme of the proposed approach: 1). Obtain data using RCWA. 2). Train the simulator using these data. 3). Train the generator with the frozen simulator together by the same data. 4). Use the fully trained simulator and the generator to conduct forward simulation and inverse design, respectively. (c) The simulator consists of convolutional layers to extract information from images and fully connected layers to convert images into vectors. (d) The generator consists of the transpose convolutional layers to generate images from sequences and the fully connected layers to extract features from images. For the blocks shown in (c)(d), the mapping relationships between color and function are described in Table 2 of the Supporting Information.

Figure 3 Several test results of the simulator and the generator. The blue curves in (a) and (b) are the real spectra in the validation set. The predicted spectra from the simulator and the simulated spectra of the devices generated by the generator are shown by the red curves in (a) and (b), respectively. The simulated spectra in (b) are obtained by RCWA.

Figure 4 Several comparative experiments for different methods when the desired spectrum cannot be described by precise value of each point. The desired spectrum is described by an inverse gaussian function in (a) and (b). A contrast vector converted from the inverse gaussian function is used as the desired contrast vector in (c). An artificially designed contrast vector considering properties of materials is used as the desired contrast vector in (d). The the generator in (b) is a benchmark trained by spectra. The generators in (c) and (d) are the same one trained by contrast vectors. The simulated spectra in (b)(c)(d) are obtained by RCWA.

Figure 5 Examples about the generated devices and their simulated spectra with artificially designed contrast vectors as inputs. The red curves stand for the artificially designed contrast vectors, while blue ones stand for the spectra simulated by RCWA. (a)(b)(c) show three peaks at the green band (492nm - 577nm), the yellow band (577nm - 597nm) and the red band (622nm - 700nm), respectively. (d)(e)(f) show three valleys at the blue band (455nm - 492nm), the green band (492nm - 577nm) and the yellow band (577nm - 597nm), respectively.

Figure 6 Visualization of the whole spectra in the entire dataset. Each row represents a real spectrum (because the data set of optical response of TM and TE polarization are similar for randomly generated patterns, only the spectra of TE polarization are shown here).

Figure 7 Comparison between the results of RCWA and simulator's output with binary/non-binary image (only TE results are shown). The threshold of image binarization is 0.5.

Figure 8 A comparative test on whether SSIM is included as a part of loss function. For each row, the *n*th image is the predicted shape of the generator at the time of 2^{n-1} epoch for the

same validation input. (a) Predicted shape from the generator with SSIM. (b) Predicted shape from the generator without supervision on shape information.

Table 1 Time efficiency comparative experiment. The CPU test is conducted on an Intel i5-6300HQ, while the GPU one on a NVIDIA GeForce GTX 960M. We take an average time based on 100 results for each method. Fourier harmonics retained for the RCWA computation is 11.

Simulation Tool	Time (ms)	Batch Size (item)	Efficiency (ms/item)
RCWA	78992.319	1	78992.319
NN-based Simulator (CPU)	4.079	1	4.079
	208.851	128	1.632
	771.966	512	1.508
NN-based Simulator (GPU)	4.945	1	4.945
	4.668	128	0.036
	4.281	512	0.008