ACS Photonics

pubs.acs.org/journal/apchd5

Article

Structural Optimization of a One-Dimensional Freeform Metagrating Deflector via Deep Reinforcement Learning

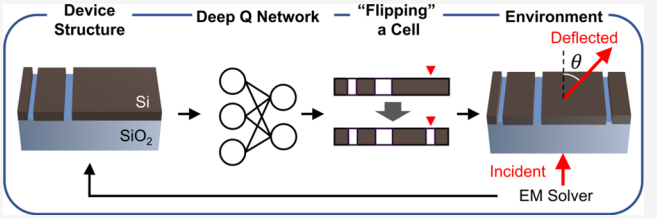
Dongjin Seo,[§] Daniel Wontae Nam,[§] Juho Park, Chan Y. Park,* and Min Seok Jang*

Cite This: *ACS Photonics* 2022, 9, 452−458

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The increasing demand on a versatile high-performance metasurface requires a freeform design method that can handle a huge design space, which is many orders of magnitude larger than that of conventional fixed-shape optical structures. In this work, we formulate the designing process of one-dimensional freeform Si metasurface beam deflectors as a reinforcement learning problem to find their optimal structures consistently without requiring any prior metasurface data. During training, a deep Q-network-based agent stochastically explores the device design space around the learned trajectory optimized for deflection efficiency. The devices discovered by the agents show overall improvements in maximum efficiency compared to the ones that state-of-the-art baseline methods find at various wavelengths and deflection angles. Furthermore, the efficiencies of the devices generated by agents trained from different neural network initializations have a small variance, demonstrating the robustness of the proposed design method.



KEYWORDS: *metasurface, freeform metagrating, structural optimization, inverse design, deep learning, reinforcement learning*

INTRODUCTION

Metasurface, an optical device with a subwavelength structure that controls the properties of light with an unprecedented spatial resolution, has attracted tremendous attention in the last decade. With their ability to manipulate the phase, amplitude, and polarization of light, metasurfaces have been utilized in various applications ranging from achromatic beam focusing,[1,2] vortex beam generation,[3,4] and holography[5,6] to optical computation[7,8] and quantum optical information processing.[9,10]

The exploding demand for high-performance metasurfaces has led researchers to seek for efficient and effective methods for an inverse design, which is a methodology of finding device structures that exhibit a desired optical response. Conventional approaches to inverse design include a genetic algorithm,[11,12] particle swarm method,[13,14] and adjoint-based algorithm.[15,16] Although these methods show a fair performance, their design space is constrained to a small number of structures. With the emergence of deep learning, which utilizes artificial neural networks to approximate arbitrary nonlinear functions, more research started to tackle problems with large dimensionality. Early approaches of the inverse design based on deep learning, however, limited the geometry of the composing device elements to fixed shapes such as circles or rectangles and altered only the sizes or the positions of the shapes to constrain the complexity of the design space within a computationally tractable amount.[17−21]

Freeform design, a domain of structural optimization in which the shapes are not explicitly defined, remained as a challenge due to its large dimensionality of search space until recent works[22−24] have suggested solutions to the freeform optimization based on a generative adversarial network (GAN)[25] and adjoint-based generative network.[26,27] However, these optimization methods either require a prior dataset with sufficiently high performance for network training[26] or suffer from a large variance of generated device performances.[26,27] Many solutions have been suggested including a recent study on deep Bayesian optimization,[28] but the enormous design space of a freeform device design has yet to be conquered.

Here, we propose a different solution by redefining the freeform optimization problem in the reinforcement learning (RL) framework, resolving both the prior data requirement and the large variance issue. RL proved itself as a tool that can solve profoundly complex problems such as the game of Go,[29] arcade video games,[30] optimization of chip floorplanning in TPU devices,[31] and the design of acoustic metasurfaces.[32] Unlike supervised learning, no training dataset is required for the algorithm since it explores the domain space and gradually exploits the experience to improve the objective function. It was shown in a previous research that deep RL can be utilized to design dielectric metasurfaces for color generation,[33] but the

Received: June 9, 2021
Published: December 30, 2021

https://doi.org/10.1021/acsphotonics.1c00839
ACS Photonics 2022, 9, 452−458

domain of the design problem had a search space of a relatively small size ($\sim 10^7$ possible structures) due to the pre-fixed device geometry. The small design space is also due to discretization of continuous variables in an attempt to make the problem more amenable to RL algorithms with discrete action spaces. On the other hand, if the optimization problem lies in a discrete space by nature, thus requiring a search algorithm that can handle discrete spaces, then the application with RL comes in naturally.

In this work, by utilizing a deep RL agent called the deep Q-network (DQN),[30] we perform the optimization of a freeform Si metasurface beam deflector whose design space is as large as $\sim 10^{17}$ possible structures. The optimal device structures found by the RL agent generally outperform those found by the previous state-of-the-art freeform optimization methods. The efficiencies of the devices found by agents trained from different neural network initializations have little variance, demonstrating the robustness of the proposed design method.

PROBLEM SETUP AND METHODS

The problem we address here is designing a large-angle beam deflector, a one-dimensional (1D) metagrating composed of Si on SiO_2 illustrated in Figure 1a. Transverse magnetic (TM) polarized light of free-space wavelength λ_0 normally impinges on the metagrating and gets diffracted at specific angles $\sin\theta_m = m\lambda_0/P$ by crystal momentum conservation, where P is the grating period and m is the diffraction order. The goal is to design an efficient beam deflector by maximizing the efficiency of $m = +1$ diffraction while suppressing the other diffraction orders. This design problem has been previously investigated using various optimization strategies.[26,27,34] In accordance with a previous work, we set the thickness of the Si layer to be 325 nm and the refractive indices of Si as the experimental data[35] and SiO_2 to be 1.45. Here, we divide one period of the metagrating layer into $N = 64$ uniform subsections, each of which we call a "cell" and represent the grating structure using a 1D array of N elements whose values are either +1 or −1 depending on whether the cell is filled with Si or air, respectively. The design space, when taking permutation degeneracies into account, is thus as large as $2^{64}/64 \sim 10^{17}$ possible structures.

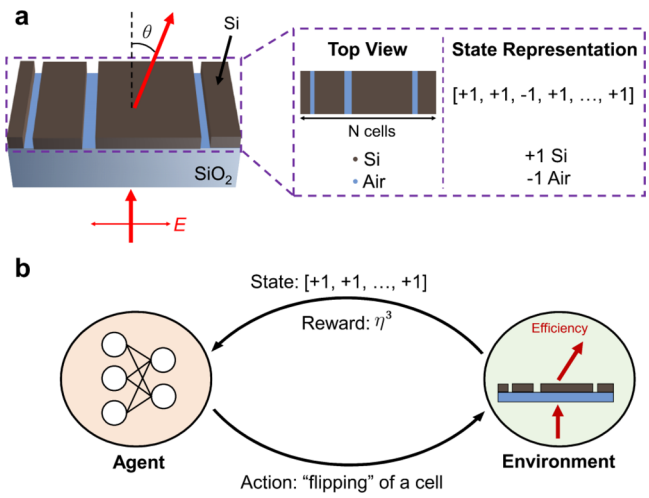


Figure 1. (a) Structure of 1D Si metagrating that deflects a normally incident TM polarized plane wave by angle θ through first-order diffraction. The Si metagrating sits on a SiO_2 substrate and has a thickness of 325 nm. One grating period is split into $N = 64$ cells and is expressed as a $1 \times N$ array whose element is either +1 or −1 depending on whether the specific cell is filled with Si or air, respectively. (b) Schematic diagram of the proposed algorithm in the RL framework. The interaction between the two main components of RL, an agent and an environment, is depicted.

Figure 1b illustrates two fundamental concepts that need to be defined to reformulate freeform optimization of the metasurface into an RL framework: the environment and agent. A state s_t, which describes the environment at given time t, is a $1 \times N$ array. Each of its elements contains +1 or −1 representing Si- and air-filled cells, respectively. The initial state is fixed as an array of only +1 values to stabilize the initial learning process. An action a_t corresponds to choosing a specific cell among N cells and switching its material between Si and air, which we call "flipping" the cell. The action of the agent changes the state of the environment to the subsequent state s_{t+1}. In this design problem, the environment performs an electromagnetic simulation based on rigorous coupled-wave analysis (RCWA)[36] that calculates the deflection efficiency η of a given metagrating structure, which is then translated into a reward r_{t+1}. An agent interacts with an environment through a series of steps or learning steps, at which the agent takes an action given a state in the environment and receives a new state and reward, producing a sequence $(s_0, a_0, r_1, \ldots, s_{M-1}, a_{M-1}, r_M, s_M)$ of states $\{s_t\}$, actions $\{a_t\}$, and rewards $\{r_{t+1}\}$. The entire sequence from the initial state s_0 to the terminal state s_M is called a trajectory or an episode. This process is formally described as a finite Markov decision process (MDP), where the sets of states S, actions A, and rewards R all have finite number of elements. Then, we can have a well-defined scalar reward function[37]

$$R: S \times A \times S \to \mathbb{R},\ R(s, a, s') \equiv \mathbb{E}[r_{t+1} | s_t = s,\ a_t = a,\ s_{t+1} = s'] \tag{1}$$

An RL agent interacts with the environment through multiple episodes while training itself with the data collected from the interactions. As the training progresses, the actions selected by the agent gradually change toward the optimal sets. However, the state-action-reward sequence trajectories that the agent has encountered are autocorrelated and thus may not cover the sufficient amount of the design space to reach near the optimality. Therefore, it is required to generate new experiences from actions that are not predicted by the agent, which is called exploration. On the other hand, generating actions that the agent deems to be optimal, called exploitation, is also necessary to improve its performance via learning. In this work, we view the whole process of exploration and exploitation as a part of the search algorithm inside the device design space.

The learning objective of RL is to maximize the discounted sum of the future rewards in an episode, which is called the return G_t defined as

$$G_t = \sum_{i=t}^{T} \gamma^{i-t} R(s_i, a_i, s_{i+1}) \tag{2}$$

for any given time t, where $\gamma \in [0,1]$ is the discount factor that assigns less weight to the reward acquired from a more distant future. If we define a policy $\pi(\cdot \mid s)$, which represents a probability distribution over actions for a given state s, then the expected return from an initial state of $s_t = s$ under policy π for

https://doi.org/10.1021/acsphotonics.1c00839
ACS Photonics 2022, 9, 452−458

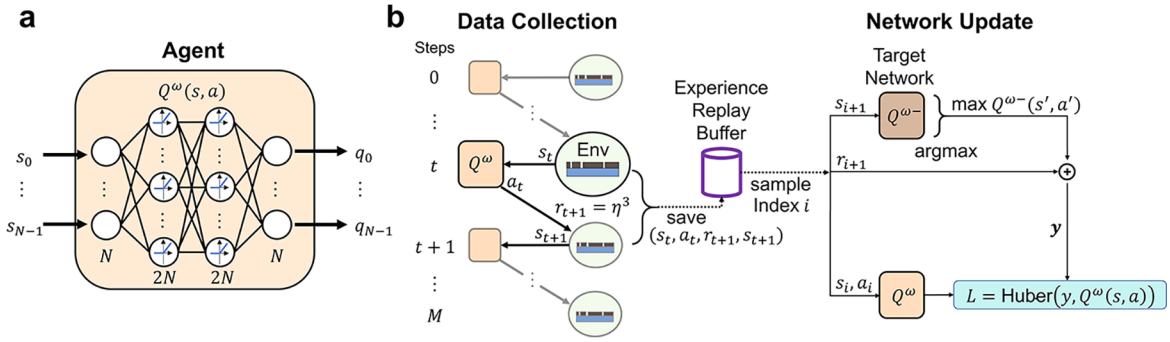


Figure 2. (a) Neural network architecture of the agent composed of two fully connected hidden layers. The input is a 1D array of the device state representation. The dimension of each hidden layer of the neural network is twice the input dimension. The output q_i is the predicted state-action value of each action a_i. (b) Information flow represented by arrows in the learning process for an episode with the length of M steps (left) and the network update process (right). The parameterized network Q^ω and the environment interact with each other by exchanging states and actions. At every state transition, a tuple of $(s_t, a_t, r_{t+1}, s_{t+1})$ is saved in the experience replay buffer. Then, during the network update stage, a minibatch of transitions is sampled from the buffer to update the network.

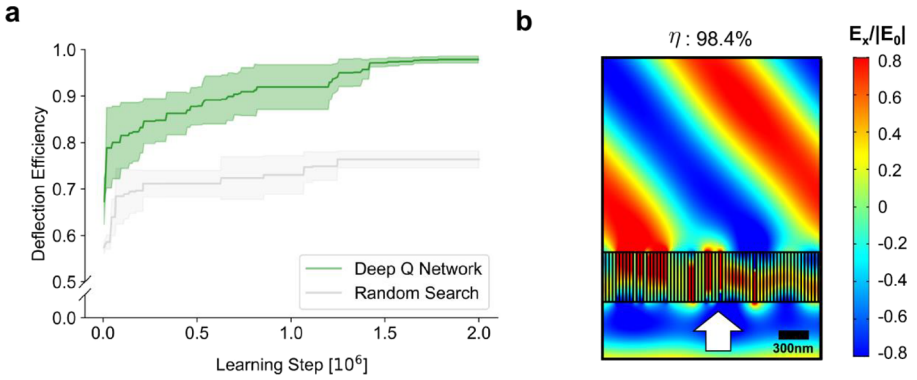


Figure 3. Optimization result of the Si metagrating beam deflector for $\lambda_0 = 1100$ nm and $\theta = 50°$. (a) Discovered maximum efficiencies during overall training across three different random neural network initializations (green). The result from three different random searches is plotted for comparison (gray). The darker line indicates the mean value, and the shaded area corresponds to one standard deviation from the mean. (b) Electric field distribution of the optimized device simulated by using the finite element method (FEM).

choosing an action $a_t = a$ at step t is called the state-action value function, or Q_π function, defined as[38]

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a] \tag{3}$$

For any MDP, the Bellman optimality equation[39] shows that taking an action according to the optimal policy for every state leads to the optimal Q^* as

$$\begin{aligned} Q^*(s, a) &= \max_\pi Q_\pi(s, a) \\ &= \mathbb{E}[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a] \end{aligned} \tag{4}$$

Here, the optimal policy is implicitly defined as the argmax operator $\max_{a'}$ for the given state-action values (s', a'), and the goal is to find the optimal state-action value function Q^*.

Finding the exact Q^* function involves sweeping over the entire search space multiple times, which is infeasible for real world problems. We simplify the problem by approximating the optimal Q^* using a neural network $Q^\omega(s, a)$ with parameters ω, as shown in Figure 2a, following the exemplary work of the DQN.[30] Finding the optimal set of parameters ω can be achieved through an optimization process of temporal-difference learning,[40] which can be expressed as the following updated equation

$$\begin{aligned} Q^\omega(s, a) \leftarrow Q^\omega(s, a) + \alpha(r_{t+1} + \gamma \max_{a'} Q^{\omega-}(s_{t+1}, a') \\ - Q^\omega(s, a)) \end{aligned} \tag{5}$$

where α is the learning rate. An additional network named target network $Q^{\omega-}$, which periodically copies the parameters from Q^ω, is introduced to stabilize possible oscillation or divergence,[30] whose value is used to calculate $\max_{a'} Q^{\omega-}(s_{t+1}, a')$ in eq 5. We parameterize our model Q^ω as a fully connected neural network with two hidden layers of $2N = 128$ nodes and the rectified linear unit (ReLU)[41] activation function.

The training procedure of Q^ω can be roughly categorized into two independent phases of data collection and network update, as depicted in Figure 2b. During the data collection phase, an agent undergoes many episodes. An episode starts with an initial state s_0. For a given state s_t, with some probability $\epsilon \in [0,1]$, an action a_t is randomly selected for exploration, called the ϵ-greedy algorithm;[37] otherwise, the agent chooses the action that leads to the maximum predicted state-action value. The exploration parameter value ϵ starts from 0.9 in the beginning and linearly decays to 0.01 and stays constant for enough number of learning steps so that the agent can exploit its experience for better decisions after enough exploration. An electromagnetic simulation is performed at every state transition to evaluate the reward $r_{t+1} = \eta(s_{t+1})^3$.

https://doi.org/10.1021/acsphotonics.1c00839
ACS Photonics 2022, 9, 452−458

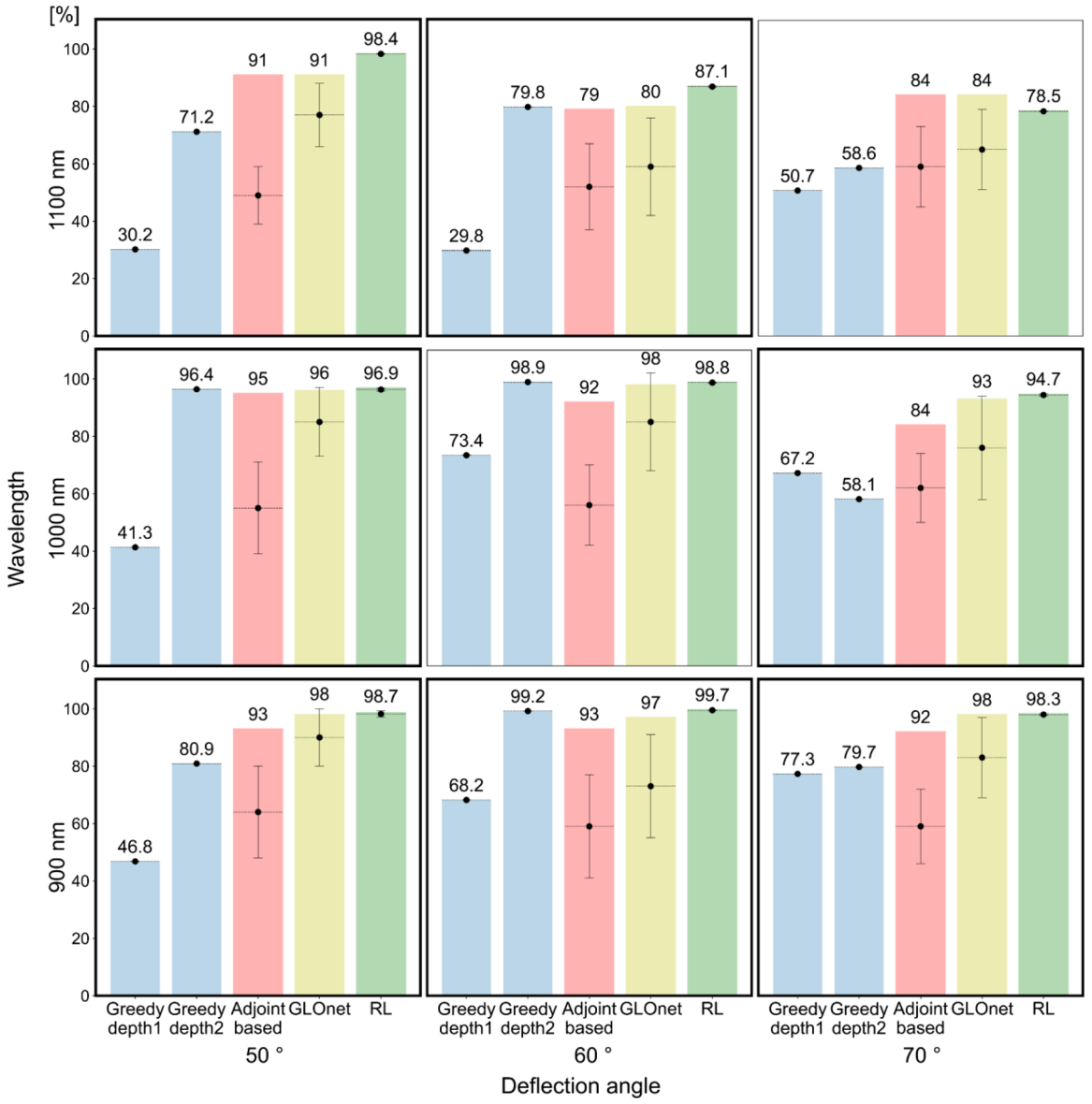


Figure 4. Performance comparison of the greedy algorithm of depths 1 and 2 (blue), the adjoint-based method (red),[27] GLOnet (yellow),[27] and the proposed RL method (green). The maximum deflection efficiency found by each algorithm is plotted as a colored bar with the value written on top of it. The mean and the standard deviation of each algorithm are depicted as a black dot and a black error bar, respectively.

The value of the reward is positively correlated with the generated device efficiency and is determined through a reward engineering process. Each transition data consisting of $(s_t, a_t, r_{t+1}, s_{t+1})$ is stored in a memory structure called experience replay buffer[42] of a fixed size of 10^6. We end an episode when the total number of steps reaches twice the number of cells in the structure $(2N = 128)$ to ensure that the agent finishes the optimization within finite and acceptable number of steps. In the network update phase, minibatches of transition data are sampled from the experience replay buffer and are used to calculate Huber loss[43]

$$\text{Huber}(x, x') = \begin{cases} \frac{1}{2}(x - x')^2 & \text{for } |x - x'| \le \delta, \\ \delta\left(|x - x'| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \tag{6}$$

where we fixed $\delta = 1$. The network update is done by minimizing the Huber loss, $L = \text{Huber}(y, Q^{\omega}(s_t, a_t))$, where $y = r_{t+1} + \gamma \max_{a'} Q^{\omega^-}(s_{t+1}, a')$ is the Bellman target calculated using a target network Q^{ω^-}. Then, the parameterized network is updated through stochastic gradient descent with the Adam optimizer.[44] We provide the summarized algorithm table of our methodology in Table S1.

RESULTS AND DISCUSSION

The result of the RL-based freeform metagrating design for $\theta = 50°$ and $\lambda_0 = 1100$ nm is summarized in Figure 3. We show the overall maximum efficiency discovered up to each training step in Figure 3a. The optimal structure found at the end of the whole process has a deflection efficiency of 98.4%, which is 7.4%p higher than the previous state-of-the-art optimization result.[27] We note that the number of structures considered in the optimization process is 2×10^6, which is a tiny fraction $(\sim 10^{-11})$ of the search space $(\sim 10^{17}$ possible structures).

https://doi.org/10.1021/acsphotonics.1c00839
ACS Photonics 2022, 9, 452−458

Compared to uniform random generation of the device, the overall maximum efficiency found by our RL method is significantly higher across different network randomizations. This implies that the agent is gradually moving toward the optimum, which cannot be obtained simply by increasing the amount of sampling data. The electric field distribution of the optimized device, which is obtained by a full-field electromagnetic simulation based on the finite element method (FEM), shows a clean wavefront of the deflected beam and a complicated local field pattern in the metagrating regime, as plotted in Figure 3b. The deflection efficiency calculated by the FEM simulation quantitatively agrees with the RCWA result within three significant figures.

To demonstrate the general performance of the proposed algorithm, we train our method for nine target conditions: wavelengths of 900, 1000, and 1100 nm and deflection angles of 50, 60, and 70°. We include greedy algorithms with $N = 64$ as baselines and the adjoint-based method and GLOnet with $N = 256$ as benchmarks.[27] The data usage for each method are 200,000 for both the adjoint-based method and GLOnet and 2,000,000 for RL. The performances of each algorithm at the nine target conditions are shown in Figure 4, where a bar graph, a dot, and an error bar represent the maximum value, the mean, and the standard deviation of the efficiencies, respectively. The numerical values of the graphs are indicated in Table S2. The results show that the RL agents are able to find the most efficient device structure in seven of the nine settings tested; the performance of the RL method remains competitive in the other two settings as well.

As its name suggests, a greedy algorithm chooses an action of the maximum figure of merit (FoM) at each step. When there are multiple actions with the same maximum FoM within three significant figures, the algorithm performs a tie-break by randomly choosing one of them. The greedy algorithm can be generalized to arbitrary depths by selecting a tuple of actions of length d that results in the highest FoM. Here, we test greedy algorithms with depth $d = \{1, 2\}$ and match the initial state to that of the RL algorithm for consistency. We note that the greedy algorithm is chosen as a baseline because our RL algorithm is an approximation to a discounted infinite depth greedy algorithm, as shown in eq 5. Both greedy algorithms were run 10 times, but the resulting efficiencies had zero variance, meaning that the random tie-break did not have an effect for the target conditions. The greedy algorithms often converge to local optima, and simply increasing the depth does not guarantee escaping the local optima as it sometimes makes the result worse, as indicated in the case of $\lambda_0 = 1000$ nm and $\theta = 70°$ in Figure 4.

Additionally, we show that variances of our RL method are much smaller than those of the adjoint-based method and GLOnet.[27] Both the adjoint-based method and GLOnet are based on a generative model from a distribution of devices, which inherently results in a relatively large variance (10−20% p standard deviation). On the other hand, our RL-based method finds devices whose efficiencies have a small variance (standard deviation <2%p) over multiple random network initializations, which is the source of the variance of our RL method. This shows the robustness of our algorithm in finding out the structure near optimal efficiency. We emphasize that the sources of variance of the methods are different but are depicted in Figure 4 in the same way to compare the possible variance of each method upon the optimal device generation.

EXISTENCE OF HIGH-IMPACT CELLS

In the process of optimizing the structure of 1D deflector metagrating devices, we have found a domain property that we call "high-impact" cells where a significant change in efficiency is induced by flipping a specific cell. These phenomena occur at various wavelengths and deflection angles and thus generally happen, without exceptions. An example is presented in Figure 5, showing that two nearly identical metagrating structures that

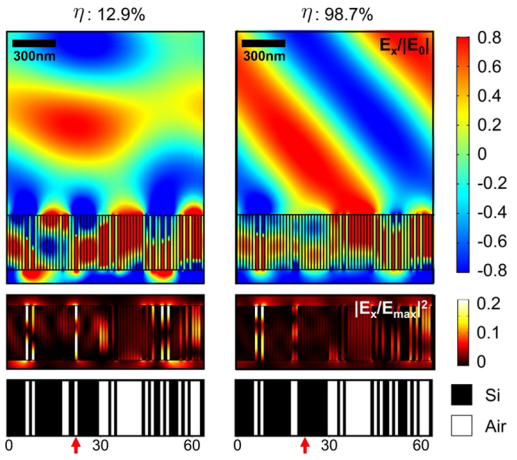


Figure 5. Analysis of a "high-impact" cell for $\lambda_0 = 900$ nm and $\theta = 50°$. The E_x field normalized by the electric field of the incident wave (top), the electric field intensity normalized by the maximum intensity (middle), and the device structure (bottom). The deflection efficiency changes from 12.9% (left) to 98.7% (right) when the cell marked by the red arrow is flipped.

differ only by one cell exhibit very different deflection efficiencies of 12.9 and 98.8%, respectively. Full-wave simulations reveal the origin of this striking phenomenon by identifying a highly concentrated electric field that is formed at specific cell positions when the structure involves narrow air gaps; if the gap becomes blocked by filling up the cell with Si, then the confined field will vanish, making a great change of the overall field profile. The existence of high-impact cells causes a large variance of FoM in a device distribution, which makes it inherently difficult to solve the optimization problem. In addition, this phenomenon implies that a new design consideration for fabrication should be introduced as we can predict the fabrication error tolerance of each cell based on the simulated electric field distribution of the device.

CONCLUSIONS

In summary, this work shows that the DQN-based RL design strategy successfully handles a high complexity freeform 1D metasurface optimization problem. The proposed design scheme can be effortlessly expanded to the problems with more than two component material types by setting the number of actions to the number of material choices. Future works also include applying the methodology to a higher-dimensional design space with higher structural and compositional complexity such as 2D or 3D designs. The high complexity of the corresponding states can be overcome by using a convolutional neural network (CNN),[45] which embeds the input features to a latent space with less dimensionality than the original design space. Our work proves a feasibility of using RL to solve an optimization problem that is combinatorial in nature, suggesting that this method can be

https://doi.org/10.1021/acsphotonics.1c00839
ACS Photonics 2022, 9, 452−458

applied to various device design problems that have been previously considered intractable due to their high complexity.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsphotonics.1c00839.

Algorithm summary, results of optimization, implementation details, validation of optimization, high-impact unit cell, the effect of random structural initialization, and additional supporting tables and figures (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Chan Y. Park − *KC Machine Learning Lab, Seoul 06181, Republic of Korea*; Email: chan.y.park@kc-ml2.com

Min Seok Jang − *School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea*; orcid.org/0000-0002-5683-1925; Email: jang.minseok@kaist.ac.kr

Authors

Dongjin Seo − *School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea; KC Machine Learning Lab, Seoul 06181, Republic of Korea*

Daniel Wontae Nam − *KC Machine Learning Lab, Seoul 06181, Republic of Korea*

Juho Park − *School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsphotonics.1c00839

Author Contributions

§D.S. and D.W.N. contributed equally to this work. D.S., C.Y.P., and M.S.J. devised the ideas. D.W.N. analyzed the idea in the RL point of view. D.S. mainly developed the algorithm code. D.W.N. and J.P. contributed to the algorithm code. D.S., D.W.N., and C.Y.P. conducted detailed analysis on the optimization process and results. J.P. performed FEM simulations. M.S.J. and C.Y.P. supervised the project. The manuscript was mainly written by D.S., D.W.N., C.Y.P., and M.S.J. with the contributions of all authors.

Funding

This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (grant nos. 2019K1A3A1A14064929, 2017R1E1A1A01074323, and 2016M3D1A1900038).

Notes

The authors declare no competing financial interest.

We open our source code and optimal structure conditions freely accessible at: https://github.com/dongjin-seo2020/1DFreeFormDQN.

■ ABBREVIATIONS

FoM,figure of merit; RL,reinforcement learning; deep RL,deep reinforcement learning; DNN,deep neural network; DQN,deep Q-learning; TM,transverse magnetic; RCWA,rigorous coupled-wave analysis; MDP,Markov decision process; ReLU,rectified linear unit; FEM,finite element method; % p,percent point

■ REFERENCES

(1) Wang, S.; Wu, P.-C.; Su, V.-C.; Lai, Y.-C.; Chu, C. H.; Chen, J.-W.; Lu, S.-H.; Chen, J.; Xu, B.; Kuan, C.-H.; Li, T.; Zhu, S.; Tsai, D. P. Broadband achromatic optical metasurface devices. *Nat. Commun.* **2017**, *8*, 187.

(2) Feng, T.; Potapov, A. P.; Liang, Z.; Xu, Y. Huygens Metasurfaces Based on Congener Dipole Excitations. *Phys. Rev. Appl.* **2020**, *13*, No. 021002.

(3) Heiden, J. T.; Ding, F.; Linnet, J.; Yang, Y.; Beermann, J.; Bozhevolnyi, S. I. Gap-Surface Plasmon Metasurfaces for Broadband Circular-to-Linear Polarization Conversion and Vector Vortex Beam Generation. *Adv. Opt. Mater.* **2019**, *7*, 1801414.

(4) Yu, N.; Genevet, P.; Kats, M. A.; Aieta, F.; Tetienne, J.; Capasso, F.; Gaburro, Z. Light Propagation with Phase Discontinuities: Generalized Laws of Reflection and Refraction. *Science* **2011**, *334*, 333−337.

(5) Han, S.; Kim, S.; Kim, S.; Low, T.; Brar, V. W.; Jang, M. S. Complete Complex Amplitude Modulation with Electronically Tunable Graphene Plasmonic Metamolecules. *ACS Nano* **2020**, *14*, 1166−1175.

(6) Ni, X.; Kildishev, A. V.; Shalaev, V. M. Metasurface holograms for visible light. *Nat. Commun.* **2013**, *4*, 2807.

(7) Estakhri, N. M.; Edwards, B.; Engheta, N. Inverse-designed metastructures that solve equations. *Science* **2019**, *363*, 1333−1338.

(8) Kwon, H.; Sounas, D.; Cordaro, A.; Polman, A.; Alù, A. Nonlocal Metasurfaces for Optical Signal Processing. *Phys. Rev. Lett.* **2018**, *121*, 173004.

(9) Jha, P. K.; Ni, X.; Wu, C.; Wang, Y.; Zhang, X. Metasurface-Enabled Remote Quantum Interference. *Phys. Rev. Lett.* **2015**, *115*, No. 025501.

(10) Bekenstein, R.; Pikovski, I.; Pichler, H.; Shahmoon, E.; Yelin, S. F.; Lukin, M. D. Quantum metasurfaces with atom arrays. *Nat. Phys.* **2020**, *16*, 676−681.

(11) Park, J.; Kim, S.; Lee, J.; Menabde, S. G.; Jang, M. S. Ultimate light trapping in a free-form plasmonic waveguide. *Phys. Rev. Appl.* **2019**, *12*, No. 024030.

(12) Fan, Y.; Xu, Y.; Qiu, M.; Jin, W.; Zhang, L.; Lam, E. Y.; Tsai, D. P.; Lei, D. Phase-controlled metasurface design via optimized genetic algorithm. *NANO* **2020**, *9*, 3931−3939.

(13) Haji-Ahmadi, M.; Nayyeri, V.; Soleimani, M.; Ramahi, O. M. Pixelated Checkerboard Metasurface for Ultra-Wideband Radar Cross Section Reduction. *Sci. Rep.* **2017**, *7*, 11437.

(14) Zhang, B.; Chen, W.; Wang, P.; Dai, S.; Li, H.; Lu, H.; Ding, J.; Li, J.; Fu, Q.; Dai, T.; Wang, Y.; Yang, J. Particle swarm optimized polarization beam splitter using metasurface-assisted silicon nitride Y-junction for mid-infrared wavelengths. *Opt. Commun.* **2019**, *451*, 186−191.

(15) Giles, M. B.; Peirce, N. A. An Introduction to the Adjoint Approach to Design. *Flow Turbul Combust* **2000**, *65*, 393−415.

(16) Molesky, S.; Lin, Z.; Piggott, A. Y.; Jin, W.; Vucković, J.; Rodriguez, A. W. Inverse design in nanophotonics. *Nat. Photonics* **2018**, *12*, 659−670.

(17) Peurifoy, J.; Shen, Y.; Jing, L.; Yang, Y.; Cano-Renteria, F.; Delacy, B. G.; Joannopoulous, J. D.; Tegmark, M.; Soljačić, M. Nanophotonic inverse design using artificial neural networks. *Sci. Adv.* **2017**, *4*, No. eaar4206.

(18) 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.

(19) 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.

(20) Sajedian, I.; Kim, J.; Rho, J. Finding the optical properties of plasmonic structures by image processing using a combination of convolutional neural networks and recurrent neural networks. *Microsyst. Nanoeng.* **2019**, *5*, 27.

(21) An, S.; Zheng, B.; Shalaginov, M. Y.; Tang, H.; Li, H.; Zhou, L.; Ding, J.; Agarwal, A. M.; Rivero-Baleine, C.; Kang, M.; Richardson, K. A.; Gu, T.; Hu, J.; Fowler, C.; Zhang, H. Deep learning modeling

approach for metasurfaces with high degrees of freedom. *Opt. Express* **2020**, *28*, 31932−31942.
(22) So, S.; Rho, J. Designing nanophotonic structures using conditional deep convolutional generative adversarial networks. *NANO* **2019**, *8*, 1255−1261.
(23) Jiang, J.; Sell, D.; Hoyer, S.; Hickey, J.; Yang, J.; Fan, J. A. Free-Form Diffractive Metagrating Design Based on Generative Adversarial Networks. *ACS Nano* **2019**, *13*, 8872−8878.
(24) An, S.; Zheng, B.; Tang, H.; Shalaginov, M. Y.; Zhou, L.; Li, H.; Kang, M.; Richardson, K. A.; Gu, T.; Hu, J.; Fowler, C.; Zhang, H. Multifunctional Metasurface Design with a Generative Adversarial Network. *Adv. Opt. Mater.* **2021**, *9*, 2001433.
(25) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Adv. Neural Info. Process. Syst.* **2014**, *27*, 2672−2680.
(26) Jiang, J.; Fan, J. A. Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network. *Nano Lett.* **2019**, *19*, 5366−5372.
(27) Jiang, J.; Fan, J. A. Simulator-based training of generative neural networks for the inverse design of metasurfaces. *NANO* **2019**, *9*, 1059−1069.
(28) Kim, S.; Lu, P. Y.; Loh, C.; Smith, J.; Snoek, J.; Soljačić, M. Scalable and Flexible Deep Bayesian Optimization with Auxiliary Information for Scientific Problems. 2021-04-23, *arXiv:2104*. 11667 [cs.LG]. https://arxiv.org/abs/2104.11667
(29) Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Driessche, G. V. D.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *429*, 484−489.
(30) Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; Hassabis, D. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529−533.
(31) Mirhoseini, A.; Goldie, A.; Yazgan, M.; Jiang, J. W.; Songhori, E.; Wang, S.; Lee, Y.; Johnson, E.; Pathak, O.; Nazi, A.; Pak, J.; Tong, A.; Srinivasa, K.; Hang, W.; Tuncer, E.; Le, Q. V.; Laudon, J.; Ho, R.; Carpenter, R.; Dean, J. A graph placement methodology for fast chip design. *Nature* **2021**, *594*, 207−212.
(32) Shah, T.; Zhuo, L.; Lai, P.; Rosa-Moreno, A. D. L.; Amirkulova, F.; Gerstoft, P. Reinforcement learning applied to metamaterial design. *J. Acoust. Soc. Am.* **2021**, *150*, 321−338.
(33) Sajedian, I.; Badloe, T.; Rho, J. Optimisation of colour generation from dielectric nanostructures using reinforcement learning. *Opt. Express* **2019**, *27*, 5874−5883.
(34) Chen, M.; Jiang, J.; Fan, J. A. Design Space Reparameterization Enforces Hard Geometric Constraints in Inverse-Designed Nanophotonic Devices. *ACS Photonics* **2020**, *7*, 3141−3151.
(35) Jiang, J.; Lupoiu, R.; Wang, E. W.; Sell, D.; Hugonin, J. P.; Lalanne, P.; Fan, J. A. Metanet: a new paradigm for data sharing in photonics research. *Opt. Express* **2020**, *28*, 13670−13681.
(36) Hugonin, J.; Lalanne, P. RETICOLO software for grating analysis. 2021-01-04, arXiv:2101.00901 [physics.optics]. https://arxiv.org/abs/2101.00901
(37) Sutton, R. S.; Barto, A. G. *Reinforcement learning: An Introduction*, 2nd edition; MIT press: 2018.
(38) Meanwhile, the expected return from an initial state of s under policy π defines a value function $V_\pi(s) = \mathbb{E}_\pi[G_t|s_t = s]$, which is also an important quantity in RL but does not play a role in this paper.
(39) Bellman, R. A Markovian Decision Process. *J. Appl. Math. Mech.* **1957**, *6*, 679−684.
(40) Sutton, R. S. Learning to Predict by the Methods of Temporal Differences. *Machine learning* **1988**, *3*, 9−44.
(41) Hanhloser, R. H. R.; Sarpeshkar, R.; Mahowald, M. A.; Douglas, R. J.; Seung, H. S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **2000**, *405*, 947−951.
(42) Lin, L. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning* **1992**, *8*, 293−321.
(43) Huber, P. J. Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 73−101.
(44) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *International Conference for Learning Representations* (ICLR 2015); 2015.
(45) Lecun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541−551.