Research Article

Chaejin Park, Sanmun Kim, Anthony W. Jung, Juho Park, Dongjin Seo, Yongha Kim, Chanhyung Park, Chan Y. Park* and Min Seok Jang*

Sample-efficient inverse design of freeform nanophotonic devices with physics-informed reinforcement learning

https://doi.org/10.1515/nanoph-2023-0852
Received November 28, 2023; accepted February 13, 2024; published online February 27, 2024

Abstract: Finding an optimal device structure in the vast combinatorial design space of freeform nanophotonic design has been an enormous challenge. In this study, we propose physics-informed reinforcement learning (PIRL) that combines the adjoint-based method with reinforcement learning to improve the sample efficiency by an order of magnitude compared to conventional reinforcement learning and overcome the issue of local minima. To illustrate these advantages of PIRL over other conventional optimization algorithms, we design a family of one-dimensional metasurface beam deflectors using PIRL, exceeding most reported records. We also explore the transfer learning capability of PIRL that further improves sample efficiency and demonstrate how the minimum feature size of the design can be enforced in PIRL through reward engineering. With its high sample efficiency, robustness, and ability to seamlessly incorporate practical device design constraints, our method offers a promising approach to highly combinatorial freeform device optimization in various physical domains.

Keywords: metasurface; adjoint-based method; reinforcement learning; physic-informed neural network; freeform design; inverse design

1 Introduction

Nanophotonic devices, having carefully designed arrangements of subwavelength elements that strongly interact with incident light waves, enable precise control of the amplitude, phase, and polarization of light at microscopic scales, allowing for highly efficient thin-film solar cells [1], optical information processing and computing [2], [3], ultrathin lenses beyond the conventional limits [4]–[7], and dynamic modulation of complex field amplitude [8]–[10]. The increasing demand for high-performance, multifunctional nanophotonic devices requires a design method that yields more performant devices than conventional fixed-shape design methods, such as a freeform design approach, which does not impose constraints on the shape or topology of the device to explore potential design candidates that were previously unattainable [11], [12]. However, due to the large number of design parameters involved, the computational load of electromagnetic simulation to generate the sample devices for structural optimization is significantly heavier when adopting a freeform approach. With the increasing demand for high-performance optical devices in recent years, the methodology of optimizing their structure has emerged as an important distinct discipline within the field of optical meta-devices, apart from the traditional theoretical progress in optics based on physical intuitions. The adjoint-based method provides a route to handle design problems involving a large design space thanks to its high sample efficiency [13]–[15], but it is essentially a local

Chaejin Park, Sanmun Kim, and Anthony W. Jung contributed equally to this work.

***Corresponding authors: Chan Y. Park**, KC Machine Learning Lab, Seoul 06181, Republic of Korea, E-mail: chan.y.park@kc-ml2.com; and **Min Seok Jang**, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea, E-mail: jang.minseok@kaist.ac.kr. https://orcid.org/0000-0002-5683-1925
Chaejin Park, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea; and KC Machine Learning Lab, Seoul 06181, Republic of Korea, E-mail: chjin777@kaist.ac.kr
Sanmun Kim, **Juho Park and Chanhyung Park**, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea, E-mail: sk902@kaist.ac.kr (S. Kim), tjrytlr12@kaist.ac.kr (J. Park)
Anthony W. Jung and Yongha Kim, KC Machine Learning Lab, Seoul 06181, Republic of Korea, E-mail: anthony@kc-ml2.com (A.W. Jung), yongha@kc-ml2.com (Y. Kim)
Dongjin Seo, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea; and AI Team, Glorang Inc., Seoul 06140, Republic of Korea, E-mail: dongjin.seo@kaist.ac.kr

Open Access. © 2024 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

optimization algorithm. Conventional population-based heuristics, which have been popularly used for global structural optimization of photonic devices [16]–[18], become inefficient when dealing with a large number of degrees of freedom (DOF) [11]. This calls for an alternative method for sample-efficient global optimization of nanophotonic devices, and machine learning can be a promising candidate.

Developments in machine learning (ML) techniques have revolutionized the field of photonic device design. Recent studies have verified the capability of neural networks to approximate the relationship between a device's structure and its optical response [19]–[23]. Additionally, generative models have been proposed to address inverse design problems with high degrees of freedom (DOF) [24]–[26]. Reinforcement learning (RL) [27], another branch of ML, is recognized to be a competitive approach to solving combinatorial problems [28]–[30] that have large DOF. A combinatorial problem involves counting, arranging, or selecting objects or elements from a finite set according to specified rules or constraints, and RL has achieved numerous breakthroughs in various problems of combinatorial nature, including the game of Go [31] and the AI accelerator chip design [32], and has also been successfully employed in designing optical metasurfaces [33], [34]. However, the requirement for a large number of training samples in ML-based methods raises concerns about the effectiveness of utilizing neural networks in photonic device design, given the substantial computational cost of electromagnetic simulation associated with device sample acquisition. But the very fact that there is an underlying physics can ease the requirement of a large number of training samples by seamlessly integrating physics and machine learning.

The practice of incorporating the physics of a system into a neural network to enhance the sample efficiency of machine learning has been investigated in various domains of physical science. For neural networks aimed at predicting physical quantities, such as electromagnetic fields [35], fluid flow [36], and quantum mechanical wavefunctions [37], the governing physical equations of each system can be utilized during the training stage to ensure that the predictions agree with the laws of physics. By incorporating this physics-informed approach, neural networks have demonstrated the ability to provide accurate predictions even in the absence of numerically simulated samples [38]. In the field of photonics, physics-informed neural networks have also been employed for device design optimization and inverse design [38], [39].

Similarly, efforts have been made to tackle RL challenges by incorporating physics information into the training pipeline, aiming to simplify high-dimensional continuous states into more intuitive representations and achieve enhanced simulation accuracy. Notable examples include research in the system control field [40], [41], as well as in the computer science domain [42], [43]. However, there have been limited advancements in incorporating physical information into reinforcement learning (RL) within the field of optics.

In this work, we introduce physics-informed reinforcement learning (PIRL), which combines the physical information from the adjoint-based method with deep RL. With PIRL, we address the optimization problem of a one-dimensional freeform metasurface beam deflector with a combinatorial design space as large as $\sim 10^{74}$. By pretraining an RL agent using the physical information, PIRL demonstrates significantly higher sample efficiency compared to the previously developed RL approach [33]. Moreover, when compared to previous studies on the same design problem, the optimal devices discovered through PIRL generally exhibit superior performance with reduced variance in terms of the same figure of merit. We also demonstrate that the sample efficiency of PIRL can be further enhanced by employing the transfer learning method [44] to the RL agent network from one design problem to similar problems. Finally, we show that practical device design constraints, such as enforcing a minimum feature size for fabrication compatibility, can be seamlessly incorporated into our PIRL framework through simple reward engineering of RL [27].

2 Problem setup and methods

Our design objective is to create a one-dimensional silicon metagrating placed on a silica substrate. This metagrating functions as a beam deflector for a normally incident transverse magnetic (TM) polarized plane wave with a wavelength λ, redirecting the beam to a first-order diffraction angle θ, as illustrated in Figure 1(a). The refractive index of silica is set to 1.45, and we use the same dispersion relation for the refractive index of silicon as in previous publications on the same system [33], [39]. The height of the silicon pillar is $h = 325$ nm, and the grating period is $P = \lambda/\sin\theta$, determined by the condition for first-order diffraction. The period is divided into $N = 256$ uniform cells. Each cell of the metagrating can be filled with either air or silicon, and the metagrating structure is represented as a 1×256 array, s_i, where the ith element specifies the material of the ith cell (+1 for Si and −1 for air), as shown in the right panel of Figure 1(a). Our goal is to find an optimal structure that achieves the highest possible absolute deflection

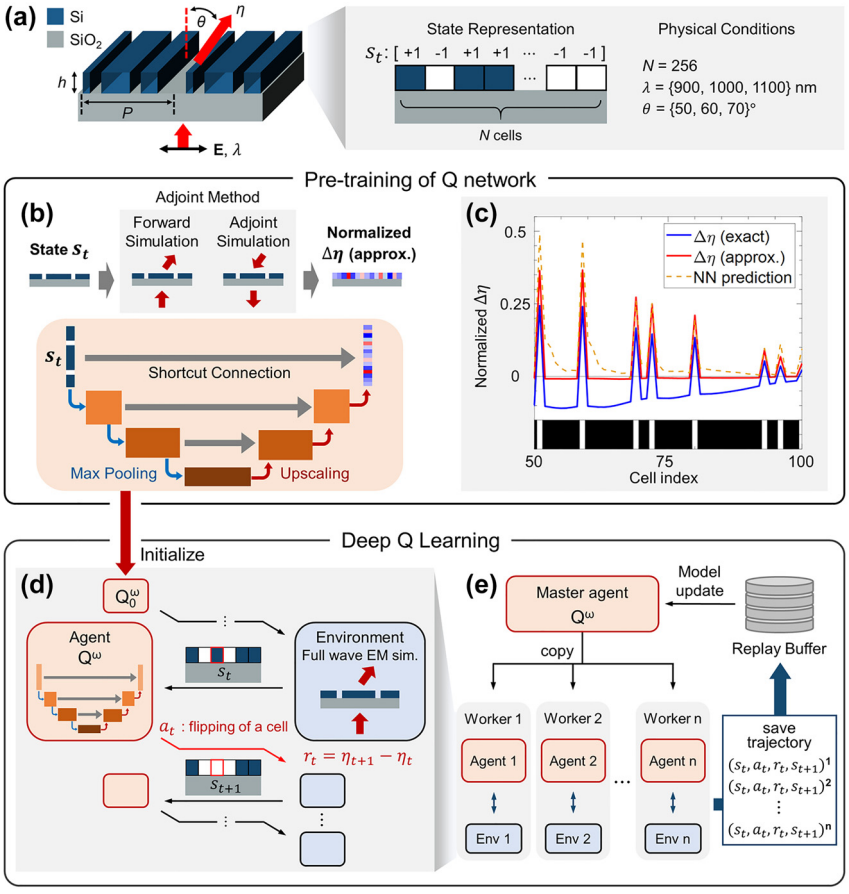

Figure 1: Summary of PIRL consisting of a pre-training stage and an RL optimization stage. (a) Schematic diagram illustrating the one-dimensional metagrating and its state representation. The metagrating is composed of silicon pillars on a silicon dioxide substrate. The goal is to maximize the first-order deflection efficiency, η, for normally incident light with transverse magnetic (TM) polarization. (b) The physics-informing pre-training stage of PIRL. The U-net shaped agent network is pre-trained to predict the normalized $\Delta\eta_{approx}$ of a given structure. The samples for network training are generated by the adjoint-based method illustrated in the top panel. (c) Comparison of $\Delta\eta_{exact}$ (blue), $\Delta\eta_{approx}$ (red), and the prediction result from the pre-trained neural network (yellow dashed line). The structure on the horizontal axis is chosen as an example that might emerge during the RL stage. (d) Illustration depicting how the agent interacts with the environment in RL. Definitions of state, action, and reward are provided. The pre-trained agent network serves as the initial state of the agent's network. (e) The parallelized RL stage comprises a master agent Q^{ω} and sixteen workers. Each worker has a copy of the agent network obtained from the master agent and independently generates trajectories by interacting with the environment.

efficiency, η, at a given wavelength and an angle. The deflection efficiency, η, is defined as the power of the deflected beam to the first order when a beam of power unity is incident from the silica substrate. We consider a target range of deflection angles and wavelengths, $\lambda = \{900, 1000, 1100\}$ nm and $\theta = \{50, 60, 70\}$ degrees, as adopted from previous studies, for direct comparison [33], [39]. We focus on the $N = 256$ case, which entails identifying an optimal structure from approximately $(2^{256}/256) \approx 10^{74}$ possible configurations, excluding degeneracy resulting from cyclic permutation. It is worth noting that the size of the design space in this problem is comparable to the number of atoms in the universe ($\sim 10^{80}$) [45]. This problem setup has been widely used as a testbed for comparing the performance of optimization algorithms in photonics, hence is suitable for testing the performance of the optimization framework we introduce in this work [33], [39].

The overall procedure of PIRL comprises two stages: a pre-training stage using supervised learning and a fine-tuning stage using RL, as depicted in Figure 1(b–d), respectively. During the pre-training stage, a neural network is trained with the current state of the one-dimensional metagrating as an input, and a proxy of the efficiency gain ($\Delta\eta$) associated with flipping each cell from Si to Air or vice versa in the metagrating as a prediction. By leveraging the Lorentz reciprocity [46], the gradient of η with respect to the refractive indices of the cells, n_i, can be estimated with only two electromagnetic simulations regardless of the number of

cells involved, as shown in the top panel of Figure 1(b) [47]. Using the efficiency gradient, $\partial\eta/\partial n_i$, the efficiency change resulting from flipping the ith cell can be approximated as $\Delta\eta_{approx} = (\partial\eta/\partial n_i)\Delta n_i$. $\Delta\eta_{approx}$ is then normalized by its L2 norm for the stability of the training of the neural network and used as an output of the supervised learning. The input and output vectors of the adjoint gradient prediction network have the same size N, and the ith entry of the output vector is correlated to the $\Delta\eta$ of the device for flipping the ith cell in the input structure. For the architecture of the neural network, we employ a U-Net [48], which is commonly used as a function approximator in the photonics domain [21], [35]. In the U-Net, features are extracted from the input through the encoding network and mapped to the output through the decoding network. Skip connections are utilized between the encoding and decoding layers to preserve spatial information. Additionally, to account for the periodic nature of the deflector, our neural network employs cyclic padding for the convolutional layers. Further details of the network architecture are provided in Figure S1. The network is trained to minimize the mean squared error loss between the predictions and the normalized $\Delta\eta_{approx}$ calculated from the adjoint-based method. The training dataset consists of 20,000 pairs of structures and adjoint gradients, with the number of training samples chosen to strike a balance between maximizing sample efficiency and achieving higher training accuracy. The prediction error as a function of the training sample size is plotted in Figure S2(a), and additional information regarding the configuration of the training dataset is presented in Figure S2(b).

The predictions of the pre-trained neural network correlate well with $\Delta\eta_{approx}$ obtained from adjoint gradients, as demonstrated in Figure 1(c). It is worth noting that the actual efficiency difference resulting from a flip action, $\Delta\eta_{exact} = \eta_{after\ flip} - \eta_{before\ flip}$, may slightly differ from the gradient-based $\Delta\eta_{approx}$. This discrepancy arises because the refractive index change associated with flipping a cell, $|\Delta n| = |n_{Si} - n_{air}| \approx 2.5$, is substantial (Figure S3). However, because calculating $\Delta\eta_{exact}$ for a device using a finite difference approach would require $N + 1 = 257$ simulations, instead, we have opted to utilize $\Delta\eta_{approx}$, which involves only two simulations and is thus more than two orders of magnitude computationally efficient. Despite the significantly reduced computational cost, $\Delta\eta_{approx}$ and the neural network predictions exhibit a similar trend to $\Delta\eta_{exact}$ as illustrated in Figure 1(c).

The pre-trained network is then utilized as the initial weights of the agent's network in the RL stage, as illustrated in Figure 1(d). Unlike the pre-trained network that focuses on the immediate return of action, the RL agent learns to pursue long-term returns, even if it entails short-term losses, through deep Q-learning [59]. This aspect is crucial for a global optimization method, since relying solely on immediate rewards may lead to convergence to local optima. During the RL stage, the agent explores the design space by iteratively interacting with the environment. The environment in this context is modeled using a rigorous coupled-wave analysis (RCWA) solver [49]. The interaction involves exchanging information such as state, action, and reward. The agent selects an action based on the current state, and the environment provides a reward as a consequence of the action.

In our approach, the state s_t is represented by a vector of length N, which corresponds to the metagrating structure as defined in the right panel of Figure 1(a). At each step t, the transition from state s_t to s_{t+1} occurs through the action a_t. The action a_t is defined as flipping the material (silicon and air) in one of the cells. Therefore, the action space is the cell number (1, 2, …, N) that will be flipped. The reward r_t is defined as the change in optical efficiency $\Delta\eta = \eta_{t+1} - \eta_t$ resulting from the action a_t. This reward setting allows the RL objective function, which is the sum of sequential rewards, to be equivalent to the final change in optical efficiency after a series of consecutive actions along the trajectory. Introducing the discount factor γ, the discounted return G_t is defined as Eq. (1), where R is the reward function.

$$G_t = \sum_{t=0}^{T-t-1} \gamma^i R\left(s_{t+i+1}, a_{t+i+1}\right) \tag{1}$$

We set $1 > \gamma \geq 0.99$ to ensure that the discounted return provides a sufficiently accurate approximation of the net change in deflection efficiency over the trajectory. By choosing a value of γ close to 1, we emphasize the long-term impact of actions on the overall optimization process. This allows the RL agent to prioritize actions that lead to substantial improvements in deflection efficiency, even if they result in temporary reductions along the trajectory.

The trial-and-error process in RL is formally represented as a Markov decision process (MDP), described as a time series tuple (s_t, a_t, r_t, s_{t+1}). The policy π, also known as the decision-making function, determines how the agent selects actions a_t given a state s_t. The Q-function, $Q_\pi = \mathbb{E}\left[G_t | s_t = s, a_t = a\right]$, estimates the expected return of acting a_t at state s_t, under the policy π. This can be also rewritten as $Q_\pi = \mathbb{E}\left[r_{t+1} + \gamma Q_\pi\left(s_{t+1}, a_{t+1}\right) | s_t = s, a_t = a\right]$, to derive optimal Q-function Q^*. Bellman optimality equation (Eq. (2)) [27] describes the recursive relationship of Q^* with itself, that optimal Q value equals the expected return for the best action among possible actions a' from that state. The best action is the action that maximizes Q value. Practically,

since the exact Q^* cannot be explicitly evaluated due to the huge state space, a neural network is used as a function approximator.

$$Q^* = \mathbb{E}\left[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a\right] \quad (2)$$

In this study, we utilize a physics-informed neural network denoted as $Q^{\omega}(s, a)$ to model the Q-function. This neural network takes a state vector as an input and predicts the Q-value for each action as the output. The $Q^{\omega}(s, a)$ is physics-informed as it is initialized with the adjoint gradient predicting network at the beginning of the RL process.

During RL, the agent follows the epsilon-greedy algorithm [27]. In this algorithm, the agent chooses either a random action with a probability ϵ (exploration) or the action with the highest Q-value according to $Q^{\omega}(s, a)$ (exploitation). The exploration probability, ϵ, linearly decreases from 0.99 (exploration-dominant) to 0.01 (exploitation-dominant) during the first half of the RL stage and remains constant at 0.01 during the second half.

Throughout the RL stage, the agent accumulates a trajectory consisting of states, actions, and rewards in the experience replay buffer, which serves as the agent's memory. The agent is trained using randomly selected data from the replay buffer. The weights of the agent's network are updated using the Huber loss [50] and Adam optimizer [51]. The summary of the PIRL algorithm is provided in Tables S1, 2.

The computation involved in the RL process is parallelized using Ray [52] as depicted in Figure 1(e). This parallelization allows for data collection by multiple workers and asynchronous network updates [53]. In this setup, sixteen workers each have their own copy of the central Q-network and interact with their own copy of the environment in parallel, collecting trajectories and storing them in the central experience replay buffer. All the hyperparameters used in the RL stage are provided in Table S6. The total number of electromagnetic simulations performed during the RL stage is set to 200,000, which is comparable to previous work on the same design problem [39]. Each RL stage takes approximately 1.3 h to complete on a server computer equipped with four Nvidia RTX 3080 GPUs and two Intel Xeon Gold 5220 processors.

3 Results/discussions

3.1 Performance of PIRL

The PIRL algorithm generally outperforms other device optimization methods. Figure 2(a) illustrates the optimization curves of various methods for the representative case of $\lambda = 1100$ nm and $\theta = 60°$. The optimization statistics were collected from ten different executions. The RL-based approaches exhibit a gradual increase in efficiency as the number of simulations increases. In contrast, the greedy

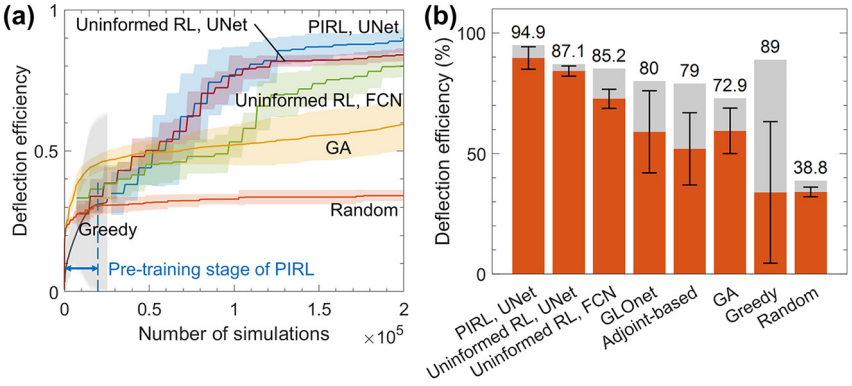

Figure 2: Performance of PIRL compared with other algorithms. (a) Optimization curves showing the maximum values of the deflection efficiency obtained using U-Net based PIRL (blue), U-Net based uninformed RL (red), fully connected network (FCN) based uninformed RL (green), genetic algorithm (GA) (yellow), random search algorithm (orange), and greedy algorithm (gray) under the target condition $\lambda = 1100$ nm, $\theta = 60°$. Each algorithm was executed ten times. The solid line represents the average maximum efficiency, and the shaded area represents the standard deviation over the ten runs. The result of the greedy algorithm has been truncated before 30,000 simulations since every run of the greedy algorithm converged to the maximum before the stop. (b) The maximum deflection efficiency of each algorithm over the ten runs. The gray bar represents the maximum of the best deflection efficiencies from the ten optimization runs, the orange bar represents the average value of the ten best deflection efficiencies, and their standard deviation is displayed as an I-shaped error bar in the graph. The numerical values of the average, standard deviation, and maximum can be found in Table 1. A summarized algorithm table comparing the algorithms can be found in Tables S1–5. The time consumption of each optimization process is summarized in Table S8. The data for GLOnet was extracted from Jiaqi Jiang et al. [58].

algorithm, which selects the cell with the highest efficiency gain ($\Delta\eta$) at each step, quickly converges to a local optimum with high dependence on initial conditions. If the PIRL agent is not trained during the RL stage and instead follows the adjoint gradient learned during the physics-informed pre-training stage, the optimization curve would resemble that of the greedy algorithm since the adjoint gradient predicts immediate rewards. However, by training the agent to approximate the discounted return in Eq. (1), the agent effectively mimics an infinite-depth greedy algorithm. This leads to slower convergence but with higher terminal efficiency. On the other hand, the optimization curve of the genetic algorithm (GA) shows slower convergence compared to the RL-based methods and does not reach an optimum value within 200,000 simulations. While there is a possibility for the GA to eventually find a better device, its low sample efficiency limits its effectiveness in optimizing devices with high degrees of freedom (DOF).

Among the RL variants, PIRL achieves the highest deflection efficiency with the fastest rate of improvement. Uninformed RL, where the Q-networks are randomly initialized without pre-training, is tested with two different network architectures: U-Net and a fully connected network (FCN). Between the two versions, U-Net outperforms FCN in terms of convergence speed and final η value. Despite having a similar number of trainable weights as FCN, the inherent network architecture of U-Net, which specializes in mapping geometric features from inputs to outputs, likely contributes to its superior performance.

Figure 2(b) summarizes the performance of various optimization methods, demonstrating that PIRL also outperforms the adjoint-based method and a physics-assisted generative model GLOnet [39] in terms of average and maximum optimized η for this specific problem. This can be qualitatively explained as follows: While the adjoint-based method is highly likely to fall into local optima, PIRL mitigates this issue by training a deep network during the RL stage (Supplementary S6). Moreover, while GLOnet relies on its stochastic nature, our method, which continually improves as the RL agent learns over time, resulted in better design than GLOnet, except in a few cases where GLOnet has a chance to discover exceptional structures. This trend is consistent across problems with different target conditions, as summarized in Table 1. It is important to note, however, that the results presented in Figure 2(b) and Table 1 should be taken with a grain of salt, as fine-tuning each algorithm could lead to improved results. A summary of each algorithm can be found in Tables S1–5. Furthermore, the optimization results from physics informed, FCN network is summarized in Figure S4 and Table S7.

3.2 Transfer learning with different target deflection angle conditions

Transfer learning [44] can enhance the sample efficiency of PIRL. In transfer learning, a neural network trained for a specific wavelength and deflection angle can be utilized for optimizing devices with different wavelength or angle conditions. There are two types of transfer learning applicable to PIRL, which are color-coded in blue and green in Figure 3(a). The first type involves transferring a pre-trained network to a different condition, while the second type transfers the fully optimized agent network from one condition to another. Figure 3(a) also depicts the regular PIRL

Table 1: Maximum, average, and standard deviation of final devices from PIRL, GLOnet, and adjoint-based method for target conditions of $\lambda = \{900\text{ nm}, 1000\text{ nm}, 1100\text{ nm}\}$ and $\theta = \{50°, 60°, 70°\}$. The reported number in PIRL column is based on the results obtained at epoch 180,000 of RL stage, as 20,000 device samples have been used in the pretraining stage.

Target condition		PIRL		GLOnet [55]		Adjoint-based method [55]	
λ	θ	Max	Mean $\pm$ Std dev	Max	Mean $\pm$ Std dev	Max	Mean $\pm$ Std dev
900 nm	50°	97.4	**96.0** $\pm$ 1.3	**98**	90 $\pm$ 10	93	64 $\pm$ 16
	60°	**99.7**	**99.5** $\pm$ 0.2	97	73 $\pm$ 18	93	59 $\pm$ 18
	70°	**99.1**	**98.6** $\pm$ 0.4	98	83 $\pm$ 14	92	59 $\pm$ 13
1000 nm	50°	**97.9**	**91.1** $\pm$ 6.2	96	85 $\pm$ 12	95	55 $\pm$ 16
	60°	**98.7**	**97.4** $\pm$ 1.3	98	85 $\pm$ 17	92	56 $\pm$ 14
	70°	**92.6**	**87.8** $\pm$ 4.3	**93**	76 $\pm$ 18	84	62 $\pm$ 12
1100 nm	50°	**95.4**	**93.3** $\pm$ 3.5	91	77 $\pm$ 11	91	49 $\pm$ 10
	60°	**94.9**	**89.6** $\pm$ 4.7	80	59 $\pm$ 17	79	52 $\pm$ 15
	70°	**84.1**	**77.8** $\pm$ 3.6	**84**	65 $\pm$ 14	**84**	59 $\pm$ 14

The bold values represent highest maximum and average value of device performance for each physical condition.

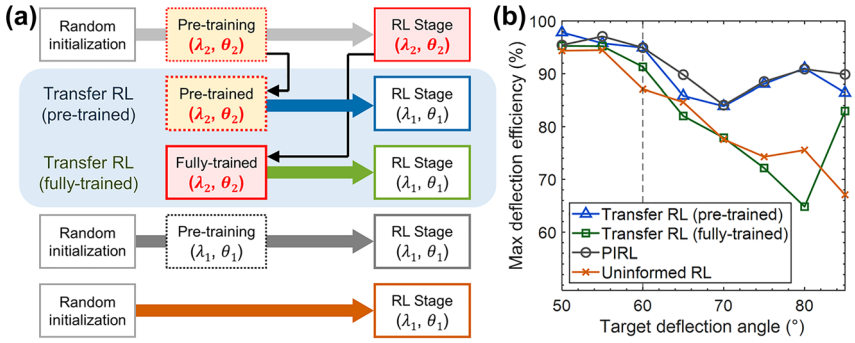

Figure 3: Transfer learning of PIRL with different target deflection angle conditions. (a) Schematic diagram illustrating two different transfer learning processes. In the first case (blue arrow), a neural network trained with the adjoint gradients of condition (θ_1, λ_1) is used as the initial network for the RL agent optimizing the problem (θ_2, λ_2). In the second case (green arrow), the neural network that underwent the full PIRL process for condition (θ_1, λ_1) is used as the initial network for the RL agent optimizing the problem (θ_2, λ_2). (b) Maximum deflection efficiency of the device obtained using each transfer learning method. Both PIRL and transfer RL with a pre-trained network outperform transfer RL with a fully-trained network and uninformed RL. The averages and standard deviations can be found in Figure S6.

and uninformed RL without any transfer learning, shown in gray and orange, respectively.

To assess the effectiveness of transfer learning, we compare the deflection efficiencies of the final devices obtained from both transfer learning cases with the outcomes of PIRL and uninformed RL, as presented in Figure 3(b). Surprisingly, even when using a pre-trained model with a mismatched pre-training dataset for $\lambda = 1100$ nm and $\theta = 60°$, transferring it to other angle conditions yields optimization performance similar to that of proper PIRL, which significantly outperforms uninformed RL. These results are remarkable because the state vector, i.e. the configuration of the deflection grating, optimized for one condition often leads to much lower deflection efficiency for a different target condition, as demonstrated in Figure S5.

In contrast, transferring a fully trained RL agent from one condition to another proves to be ineffective and yields results comparable to those of uninformed RL with a randomly initialized Q network. Although the fully trained agent typically starts with better optimization performance, it eventually converges into a low-performance device. In other words, it appears that a pre-trained network exhibits enough flexibility to adapt to a new target condition, while a fully trained network is too rigid to effectively learn new strategies to escape the local optimum from which it starts. Similar behaviors of pre-trained deep neural networks for fine-tuning have been observed in previous studies of meta-learning [54].

3.3 Enforcing the minimum feature size

In general, optimal devices found using the PIRL which does not constrain the minimum feature size (MFS) lack fabricability. For example, the device optimized for $\lambda = 1100$ nm and $\theta = 60°$ in Figure 4(a) can hardly be fabricated even with cutting-edge facilities. This is because the MFS of the device is approximately 5 nm, which corresponds to the width of a single cell in the design grid. Simply removing these small features through a Gaussian filter with a half-MFS standard deviation is not a viable solution, as it leads to a catastrophic failure with a drastic drop of 90 %p in deflection efficiency, as shown in Figure 4(b).

To address these fabricability concerns, we propose a method for enforcing the MFS constraint within the PIRL framework by modifying the reward function. In the MFS-enforced PIRL, fabricability is incorporated into the reward by subtracting a penalty function $\alpha\Delta B$ from the original reward $\Delta\eta$. Here, B represents the number of pillars or gaps that fall below the MFS limit, and α is a penalty constant that determines the level of enforcement. The value of α is empirically determined by selecting the minimum value that ensures the device satisfies the MFS condition. It's important to note that with this reward setting, the undiscounted return corresponds to the net change in efficiency over the trajectory, assuming the final structure doesn't violate the MFS constraint.

Figure 4(c) showcases the optimized structure obtained using the MFS-enforced PIRL for $N = 256$ and MFS = 16 cells, with α set to 0.1. It achieves a deflection efficiency of 74.0 %, significantly surpassing the optimal structure among the subset of $N = 16$ (Figure 4(d)). Our method also discovered a device with an efficiency of 85.1 % for the problem of $N = 256$ and MFS = 8 cells, outperforming the optimum found for $N = 32$ by 1.3 %.

Another approach to enforcing the MFS constraint is by reducing the number of grid cells, N, to match the required

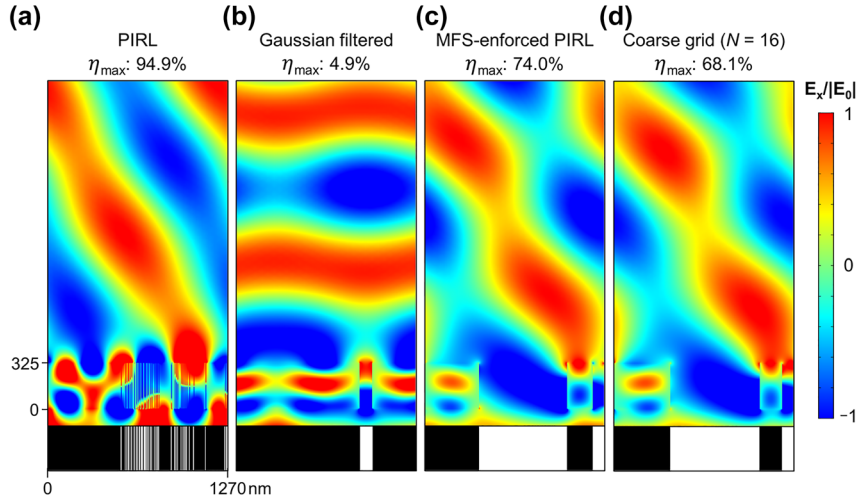

Figure 4: Electric field profiles from beam deflectors designed for a target condition of $\lambda = 1100$ nm and $\theta = 60°$. All electric field profiles are normalized to the electric field intensity of the incident wave. (a) Structure of the highest deflection efficiency device found with PIRL along with the corresponding electric field distribution. However, this device cannot be fabricated even with cutting-edge fabrication techniques. (b) Structure and electric field resulting from the Gaussian-filtered device. The device found with PIRL is filtered using a Gaussian filter with a standard deviation $\sigma = 4$ and then binarized. Although the minimum feature size is increased to eighteen, the deflection efficiency drops by 86.45 %p. (c) Electric field profile and device structure of the beam deflector obtained from MFS-enforced PIRL. The smallest feature in the device is a gap of 16 cells. (d) Electric field profile and device structure of the beam deflector obtained from the resolution-limited device (DOF = 16). An exhaustive search was conducted to find the true global optimum.

MFS. However, this coarse grid approach is also undesirable because it confines the design space to a tiny subset of possible outcomes as gaps between features smaller than the minimum feature size can be fabricated in reality. Even the global optimum structure obtained from an exhaustive search within this limited design space has a significantly lower deflection efficiency. For example, the global optimum structure obtained for $N = 16$ (MFS $\sim$ 80 nm), as depicted in Figure 4(d), achieves a deflection efficiency of 68.1 %. This is more than 5 %p lower than the optimal structure found for the same MFS with $N = 256$ in Figure 4(c), which is found using MFS-enforced PIRL.

4 Conclusions

This work represents the initial endeavor to incorporate physical information (adjoint gradient) into RL for the design of highly complex optical devices. In this work, we introduce PIRL, a method that integrates physical information from the adjoint-based method into RL for designing highly complex optical devices. By initializing the RL agent's network with the figure-of-merit gradient, we significantly enhance the sample efficiency for optimizing the structure, surpassing the previous work by more than an order of magnitude [33]. To demonstrate the effectiveness of PIRL, we directly compare it with other existing methods, ranging from conventional genetic algorithms to deep generative models [39]. Furthermore, we show that transfer learning can further improve the sample efficiency of PIRL by successfully transferring networks between design problems with different target conditions. Additionally, we address the need for fabrication feasibility by modifying PIRL to enforce a minimum feature size in devices through reward engineering.

The optimization framework can be applied to optimize other devices where the adjoint gradient can be calculated, such as a two-dimensional metagrating, or a meta lens. Furthermore, the developed method can be extended to other techniques as long as the simulation tool allows for the calculation of local gradients of each design element within a limited number of simulations. For instance, automatic differentiation-enabled RCWA tools have the capability to compute local gradients of design elements in a comparable timescale to the adjoint method [49], [55]–[57]. By combining such tools with PIRL, the optimization of devices with intricate figure-of-merit functions becomes feasible. We anticipate that this optimization method will empower RL to address seemingly intractable problems in photonics device design.

Research funding: This work was supported by the LX Semicon – KAIST Future Research Center. This research was also supported by the Ministry of Trade, Industry & Energy (MOTIE) (1415180303), the Korea Semiconductor Research Consortium (KSRC) (20019357), and the Ministry of Science

and ICT (MSIT) (NRF-2022R1A2C2092095). The work of C.Y.P. was performed in part at Aspen Center for Physics, which is supported by National Science Foundation grant PHY-2210452.

Author contributions: M.S.J. conceived the presented idea. Chaejin P., S.K., and C.Y.P. contributed to the further development of the idea. Chaejin P., J.P., and D.S. designed the model and the computational framework. A.W.J. did algorithm parallelization and resolved the technical problems in RL. Y.K. developed the electromagnetic simulation Python package. Chanhyung P. performed numerical simulations on COMSOL multiphysics. Chaejin P. and S.K. carried out simulations and analyzed the data. Chaejin P., S.K., and A.W.J. wrote the manuscript with comments and revisions from C.Y.P. and M.S.J. C.Y.P. and M.S.J. supervised the project.

Conflict of interest: The authors declare no conflict of interest.

Informed consent: Informed consent was obtained from all individuals included in this study.

Ethical approval: The conducted research is not related to either human or animals use.

Data availability: The source code is available from the following GitHub repository. https://github.com/jLabKAIST/Physics-Informed-Reinforcement-Learning.

References

[1] H. A. Atwater and A. Polman, "Plasmonics for improved photovoltaic devices," *Nat. Mater.*, vol. 9, no. 3, pp. 205–213, 2010.
[2] N. Mohammadi Estakhri, B. Edwards, and N. Engheta, "Inverse-designed metastructures that solve equations," *Science*, vol. 363, no. 6433, pp. 1333–1338, 2019.
[3] H. Kwon, D. Sounas, A. Cordaro, A. Polman, and A. Alù, "Nonlocal metasurfaces for optical signal processing," *Phys. Rev. Lett.*, vol. 121, no. 17, p. 173004, 2018.
[4] J. B. Pendry, "Negative refraction makes a perfect lens," *Phys. Rev. Lett.*, vol. 85, no. 18, pp. 3966–3969, 2000.
[5] Z. Jacob, L. V. Alekseyev, and E. Narimanov, "Optical Hyperlens: far-field imaging beyond the diffraction limit," *Opt. Express*, vol. 14, no. 18, pp. 8247–8256, 2006.
[6] Z. Liu, H. Lee, Y. Xiong, C. Sun, and X. Zhang, "Far-field optical hyperlens magnifying sub-diffraction-limited objects," *Science*, vol. 315, no. 5819, p. 1686, 2007.
[7] W. T. Chen, *et al*., "A broadband achromatic metalens for focusing and imaging in the visible," *Nat. Nanotechnol.*, vol. 13, no. 3, pp. 220–226, 2018.
[8] S. Han, S. Kim, S. Kim, T. Low, V. W. Brar, and M. S. Jang, "Complete complex amplitude modulation with electronically tunable graphene plasmonic metamolecules," *ACS Nano*, vol. 14, no. 1, pp. 1166–1175, 2020.
[9] J. Park, *et al*., "All-solid-state spatial light modulator with independent phase and amplitude control for three-dimensional LiDAR applications," *Nat. Nanotechnol.*, vol. 16, no. 1, pp. 69–76, 2021.
[10] J. Y. Kim, *et al*., "Full 2π tunable phase modulation using avoided crossing of resonances," *Nat. Commun.*, vol. 13, no. 1, p. 2103, 2022.
[11] J. Park, S. Kim, D. W. Nam, H. Chung, C. Y. Park, and M. S. Jang, "Free-form optimization of nanophotonic devices: from classical methods to deep learning," *Nanophotonics*, vol. 11, no. 9, pp. 1809–1845, 2022.
[12] D. Sell, J. Yang, S. Doshay, R. Yang, and J. A. Fan, "Large-angle, multifunctional metagratings based on freeform multimode geometries," *Nano Lett.*, vol. 17, no. 6, pp. 3752–3757, 2017.
[13] J. S. Jensen and O. Sigmund, "Systematic design of photonic crystal structures using topology optimization: low-loss waveguide bends," *Appl. Phys. Lett.*, vol. 84, no. 12, pp. 2022–2024, 2004.
[14] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," *Opt. Express*, vol. 21, no. 18, pp. 21693–21701, 2013.
[15] H. Chung and O. D. Miller, "High-NA achromatic metalenses by inverse design," *Opt. Express*, vol. 28, no. 5, pp. 6945–6965, 2020.
[16] S. Jafar-Zanjani, S. Inampudi, and H. Mosallaei, "Adaptive genetic algorithm for optical metasurfaces design," *Sci. Rep.*, vol. 8, no. 1, p. 11040, 2018.
[17] J. Li, *et al*., "Inverse design of multifunctional plasmonic metamaterial absorbers for infrared polarimetric imaging," *Opt. Express*, vol. 27, no. 6, pp. 8375–8386, 2019.
[18] J. Park, S. Kim, J. Lee, S. G. Menabde, and M. S. Jang, "Ultimate light trapping in a free-form plasmonic waveguide," *Phys. Rev. Appl.*, vol. 12, no. 2, p. 024030, 2019.
[19] J. Peurifoy, *et al*., "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, no. 6, p. eaar4206, 2018.
[20] S. Kim, *et al*., "Inverse design of organic light-emitting diode structure based on deep neural networks," *Nanophotonics*, vol. 10, no. 18, pp. 4533–4541, 2021.
[21] P. R. Wiecha and O. L. Muskens, "Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3D nanostructures," *Nano Lett.*, vol. 20, no. 1, pp. 329–338, 2019.
[22] M. H. Tahersima, *et al*., "Deep neural network inverse design of integrated photonic power splitters," *Sci. Rep.*, vol. 9, no. 1, p. 1368, 2019.
[23] S. Inampudi and H. Mosallaei, "Neural network based design of metagratings," *Appl. Phys. Lett.*, vol. 112, no. 24, p. 241102, 2018.
[24] S. An, *et al*., "Multifunctional metasurface design with a generative adversarial network (advanced optical materials 5/2021)," *Adv. Opt. Mater.*, vol. 9, no. 5, p. 2001433, 2021.
[25] J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, "Free-Form diffractive metagrating design based on generative adversarial networks," *ACS Nano*, vol. 13, no. 8, pp. 8872–8878, 2019.
[26] W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy," *Adv. Mater.*, vol. 31, no. 35, p. 1901111, 2019.
[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MIT press, 2018.
[28] S. Greengard, "Better algorithms through faster math," *Commun. ACM*, vol. 66, no. 6, pp. 11–13, 2023.

[29] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," arXiv preprint arXiv:1611.09940, 2016.
[30] W. Kool, H. V. Hoof, and M. Welling, *Presented at the International Conference on Learning Representations*, 2018.
[31] D. Silver, *et al*., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484—489, 2016.
[32] A. Mirhoseini, *et al*., "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207—212, 2021.
[33] D. Seo, D. W. Nam, J. Park, C. Y. Park, and M. S. Jang, "Structural optimization of a one-dimensional freeform metagrating deflector via deep reinforcement learning," *ACS Photonics*, vol. 9, no. 2, pp. 452—458, 2021.
[34] I. Sajedian, T. Badloe, and J. Rho, "Optimisation of colour generation from dielectric nanostructures using reinforcement learning," *Opt. Express*, vol. 27, no. 4, pp. 5874—5883, 2019.
[35] M. Chen, *et al*., "High speed simulation and freeform optimization of nanophotonic devices with physics-augmented deep learning," *ACS Photonics*, vol. 9, no. 9, pp. 3110—3123, 2022.
[36] S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, "Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks," *J. Fluid Mech.*, vol. 915, p. A102, 2021.
[37] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686—707, 2019.
[38] Z. Fang and J. Zhan, "Deep physical informed neural networks for metamaterial design," *IEEE Access*, vol. 8, pp. 24506—24513, 2019.
[39] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," *Nano Lett.*, vol. 19, no. 8, pp. 5366—5372, 2019.
[40] G. Gokhale, B. Claessens, and C. Develder, "PhysQ: a physics informed reinforcement learning framework for building control," arXiv:2211.11830, 2022.
[41] D. Cao, *et al*., "Physics-informed graphical representation-enabled deep reinforcement learning for robust distribution system voltage control," *IEEE Trans. Smart Grid*, vol. 15, no. 1, pp. 233—246, 2024.
[42] A. Ramesh and B. Ravindran, "Physics-informed model-based reinforcement learning," arXiv:2212.02179, 2022.
[43] C. Xie, S. Patil, T. Moldovan, S. Levine, and P. Abbeel, "Model-based reinforcement learning with parametrized physical models and optimism-driven exploration," arXiv:1509.06824, 2015.
[44] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, Hershey, PA, IGI global, 2010, pp. 242—264.
[45] P. Collaboration, *et al*., "Planck intermediate results XXIV. Constraints on variations in fundamental constants?," *A. & A.*, vol. 580, pp. 1—25, 2015.
[46] E. J. Rothwell and M. J. Cloud, *Electromagnetics*, shey, Boca Raton, PA, CRC Press, 2018.
[47] M. P. Bendsoe and O. Sigmund, *Topology optimization: Theory, Methods, and Applications*, Springer Science & Business Media, 2003.
[48] O. Ronneberger, P. Fischer, and T. Brox, *Presented at the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Munich, 2015, (unpublished).
[49] Y. Kim, *et al*., *An Electromagnetic Simulation Software*, 2024. Available at: https://github.com/kc-ml2/meent.
[50] P. J. Huber, "Robust estimation of a location parameter," *Ann. Stat.*, vol. 53, no. 1, pp. 73—101, 1964.
[51] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[52] E. Liang, *et al*., *Presented at the International Conference on Machine Learning*, 2018.
[53] V. Mnih, *et al*., *Presented at the International Conference on Machine Learning*, 2016.
[54] C. Finn, P. Abbeel, and S. Levine, *Proceedings of the 34th International Conference on Machine Learning, PMLR Proceedings of Machine Learning Research*, vol. 70, P. Doina and T. Yee Whye, Eds., 2017, pp. 1126—1135.
[55] W. Jin, W. Li, M. Orenstein, and S. Fan, "Inverse design of lightweight broadband reflector for relativistic lightsail propulsion," *ACS Photonics*, vol. 7, no. 9, pp. 2350—2355, 2020.
[56] C. Kim and B. Lee, "TORCWA: GPU-accelerated Fourier modal method and gradient-based optimization for metasurface design," *Comput. Phys. Commun.*, vol. 282, p. 108552, 2023.
[57] S. Colburn and A. Majumdar, "Inverse design and flexible parameterization of meta-optics using algorithmic differentiation," *Commun. Phys.*, vol. 4, no. 1, p. 65, 2021.
[58] J. Jiang and J. A. Fan, "Simulator-based training of generative neural networks for the inverse design of metasurfaces," *Nanophotonics*, vol. 9, no. 5, pp. 1059—1069, 2019.
[59] V. Mnih, *et al*., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529—533, 2015.

Supplementary Material: This article contains supplementary material (https://doi.org/10.1515/nanoph-2023-0852).