MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Application of Deep Learning for Nanophotonic Device Design

Kojima, Keisuke; Tang, Yingheng; Koike-Akino, Toshiaki; Wang, Ye; Jha, Devesh K.; TaherSima, Mohammad; Parsons, Kieran

TR2020-182 March 06, 2021

Abstract

We present three different approaches to apply deep learning to inverse design for nanophotonic devices. The forward models use device parameters as inputs and device responses as outputs. This model works as a fast approximation method which can be integrated in the optimization loop, and can accelerate the optimization. The network is updated as we obtain more simulation data on the fly for better approximation. The inverse modeling uses a network trained with the device responses as inputs, and the device parameters as outputs. This way the network outputs the device structure given the target optical response. This network can also be updated as we obtain more data during the optimization and validation. The generative model we use is a variant of a conditional variational autoencoder, and the network learns the statistical characteristics of the device structure, and it generates a series of improved designs given the target device responses. By using these three models, we demonstrate how to design nanophotonic power splitters with multiple splitting ratios.

SPIE Photonics West 2021

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Application of deep learning for nanophotonic device design

Keisuke Kojima[a], Toshiaki Koike-Akino[a], Yingheng Tang[a,b], Ye Wang[a], Devesh K. Jha[a], and Kieran Parsons[a]

[a]Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA, USA.
[b]Electrical and Computer Engineering Dept., Purdue University, West Lafeyette, IN, USA

ABSTRACT

We present three different approaches to apply deep learning to inverse design for nanophotonic devices. The forward models use device parameters as inputs and device responses as outputs. This model works as a fast approximation method which can be integrated in the optimization loop, and can accelerate the optimization. The network is updated as we obtain more simulation data on the fly for better approximation. The inverse modeling uses a network trained with the device responses as inputs, and the device parameters as outputs. This way the network outputs the device structure given the target optical response. This network can also be updated as we obtain more data during the optimization and validation. The generative model we use is a variant of a conditional variational autoencoder, and the network learns the statistical characteristics of the device structure, and it generates a series of improved designs given the target device responses. By using these three models, we demonstrate how to design nanophotonic power splitters with multiple splitting ratios.

Keywords: Nanophotonics, inverse design, power splitter, WDM filter, deep learning, neural network, generative model, adversarial block

1. INTRODUCTION

Nanostructured materials with subwavelength dimensions can be used to control incident electromagnetic fields into specific transmitted and reflected wavefronts. Recent nanophotonic devices have used such complex structures to enable novel applications in optics, integrated photonics, sensing, and computational metamaterials in a compact and energy-efficient form.[1–13] However, optimizing nanostructures with a large number of possible combinations of parameters is a very challenging task.

There have been increased attention to the application of neural networks (NNs) and specifically deep neural networks (DNNs) for improving the automation of nanophotonic device designs. There are multiple approaches, and this paper overviews three major categories. The first category is forward modeling, where the DNN takes the device topology as an input and the optical response is the output. In this case, for inverse design, we would use the trained DNN in surrogate optimization, e.g., optimization metric calculation in meta-heuristic optimizers such as direct binary search (DBS) and evolutionary strategies (ES). The forward-modeling DNN can skip time-consuming electromagnetic simulations, such as the finite-difference time-domain (FDTD) method, to predict the optical response. An NN-based forward modeling in conjunction with an optimization method has been used for designing nanophotonic power splitters,[14,15] nanophotonic mode converters,[16] and optical attenuators.[11]

The second category is inverse modeling,[15,17] where the optical response is the input and the device topology is the output. In this case, trained network is used to directly generate the desired device topology given the target optical response as input. This inverse modeling can be more useful for inverse design than forward

Further author information: (Send correspondence to Keisuke Kojima)
Keisuke Kojima: E-mail: kojima@merl.com, Telephone: 1 617 621 7598
Toshiaki Koike-Akino: E-mail: kojima@merl.com, Telephone: 1 617 621 7543
Yingheng Tang: E-mail: tang96@purdue.edu, Telephone: 1 765 637 6785
Ye Wang: E-mail: yewang@merl.com, Telephone: 1 617 621 7521
Devesh K. Jha: E-mail: jha@merl.com, Telephone: 1 617 621 7513
Kieran Parsons: E-mail: parsons@merl.com, Telephone: 1 617 621 7529

modeling as we do not need to use external meta-heuristic optimizers. Note that the inverse-modeling DNN usually provides only one topology candidate per target response, which limits the capability to explore better geometries.

The third category is generative modeling such as conditional autoencoders (CAE) and generative adversarial networks (GAN), wherein the network is trained using the device topology and optical response. Once the network is trained, we give a target optical response as well as random numbers, and the network generates a series of improved topologies. Generative models have been recently used for designing materials[18] and plasmonic devices,[19,20] and later for nanophotonic devices.[21,22]

2. DEVICE STRUCTURE

Integrated photonic beam splitters have been widely used to equally divide the power into the output ports. In this section, we consider a nanostructured power splitter with an arbitrary and fixed splitting ratio towards two output ports, targeting flat response with low insertion loss. Such a power splitter can be used as a building block of many types of photonic integrated circuits for optical communications and various other applications.

We use a silicon-on-insulator (SOI)-based structure with one input and two output ports having 0.5μm wide waveguides connected using a taper to the 2.25μm wide square power splitter design as shown in Fig. 1. Each hole is a circle with a maximum diameter of 72 nm that is easily fabricable using well-established lithography methods.[23,24] In order to comply with the fabrication limit, we also put a constraint that the minimum hole diameter is 40 nm, when continuous variables are used for the design process as in Section 6.

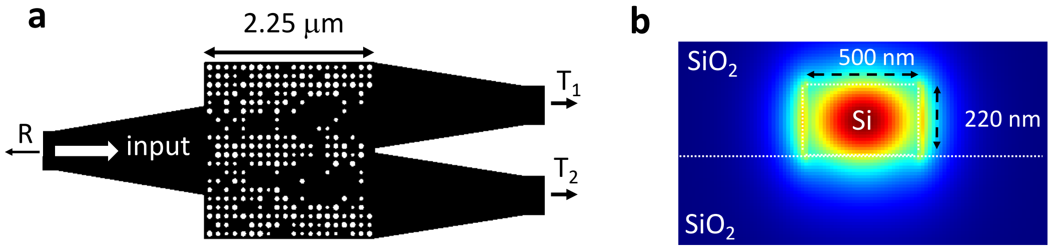

Figure 1: Schematic of the SOI-based power splitter. a) Top view, where T_1 and T_2 denote the modal transmissions of output ports 1 and 2, and R denotes the reflection at input port. b) Cross-section of the input/output waveguide.[22] By optimizing the binary sequence of positions of etch hole it is possible to adjust light propagation into either of the ports. In Section 6, continuous variables are used to represent the variable hole sizes.

3. SIMULATION AND DNN MODELING PROCEDURES

Lumerical FDTD simulations are used to generate labeled data for training and for verifications. The fundamental transverse electric (TE) mode is used at input source and TE mode output power is recorded for transmission and reflection.

The data for each structure consists of a hole vector (HV) of size 20 × 20 and labels of its spectral response (SPEC) at the two output ports and reflection at the input port whose size is 63 (3 data sets × 21 frequency points). In the HV, each hole can be represented by a binary state of '1' for etched (n_{Silicon}) and '0' for not etched (n_{Silica}) for the experiments described in Section 4 and 5, or continuous variables associated with hole area relative to the maximum size of 72 nm for the experiments described in Section 6. We use random patterned initial HVs and optimize them using heuristic optimization approaches for various optimization metrics to collect a diverse set of labeled training data for supervised learning.

We use the open source machine learning framework of Keras using Tensorflow as a backend in Section 4, and PyTorch in Sections 5 and 6 in the Python language to build and test our DNNs. The DNN training, testing, and subsequent FDTD simulations are conducted on a computer with a graphics processing unit (GPU) board.

4. FORWARD MODELING

We first describe the forward regression model, using a DNN to predict the transmission and reflection spectra. Given the two-dimensional binary array (20×20), we train a DNN using the corresponding transmission and reflection spectra vector, consisting of spectral data for transmission at both port 1, port 2, and reflection at the input port each at 21 discrete wavelengths from 1300 to 1800 nm. Once the DNN is trained, it is used as the predicted (also called "surrogate") spectra within a DBS optimization loop. Our forward modeling network consists of three 2D convolutional layers followed by one fully connected layer.[15]

In the DBS optimization process, we minimized the following loss function:

$$\mathsf{Metric} = \left|T_1 - T_1^\star\right|^2 + \left|T_2 - T_2^\star\right|^2 + \alpha\left|R - R^\star\right|^2, \tag{1}$$

where T_1 and T_2 are the lowest transmitted power within the spectral range of 1300 nm and 1800 nm, R is the largest reflection power, and $\alpha = 10$ is a weighting factor. We let $[\cdot]^\star$ denote the corresponding target values. We start with random patterns of 20×20 binary HVs, and generated about 11,000 training data by using the standard DBS, with four target splitting ratios. Due to the symmetry, the actual number of distinct data used in the training is about 22,000. We then try to design a power splitter with $T_1^\star = 0.27$ and $T_2^\star = 0.73$, starting from the three initial conditions indicated by the red circles in Fig. 2.

In the DNN-assisted DBS, we first train the DNN with 300 epochs using the initial training data. Next, we virtually flip each of the 400 holes, using the output of the DNN as a predicted spectral response. We then select a new pattern with a flipped hole corresponding to the lowest metric and verify the spectra via an FDTD simulation. When the actual FDTD result is better as expected, the flipped pattern is retained, otherwise, we try to flip another hole corresponding to the next best metric and verify with an FDTD simulation. We train with one epoch of the original training data and 15 epochs of the accumulated newly acquired data. This is essentially an accelerated DBS using metric values predicted by the DNN.

Fig. 3 shows a comparison between a conventional DBS and a DNN-assisted DBS (denoted as DL-DBS), plotting the metric as a function of the number of FDTD runs. This confirms that the DNN-assisted DBS optimizes the device structure faster than the conventional DBS and leads to better device designs overall. Note that each FDTD run takes about two minutes, while the additional training (active learning) for each FDTD run takes 20 seconds. Hence, there is an overhead of about 20% per FDTD. The total transmittance greater than 87.5% is achieved with a low reflection below 0.5%.

The key is to take advantage of the very fast forward model computation to predict performance of numerous candidates, which can save time-consuming FDTD simulations.

5. INVERSE MODELING

The inverse modeling takes the spectral response as the input and directly generates the device topology as the output. We here consider an inverse modeling DNN consisting of three fully connected layers and two 2D deconvolution layers. The deconvolutional layers, not included in our original inverse modeling paper,[17] improves the performance. Note that if we use a very small number of performance metrics such as only transmitted power at a single wavelength, we will face a severe issue of non-bijectivity for the forward function, where multiple geometries $\boldsymbol{x}$ may end up having the identical performance $\boldsymbol{y}$, making the inverse design challenging. Therefore, it is desirable to have a relatively large number of performance metrics (in our case $m = 63$) to increase the entropy of $\mathbb{H}(\boldsymbol{y})$ in comparison to $\mathbb{H}(\boldsymbol{x})$ for the HV dimension of $n = 400$.

The DNN is trained to predict HV in favor of minimizing the binary cross-entropy (BCE) loss as follows:

$$\mathsf{BCE} = -\sum_{i=1}^{n} \Big[x_i \log \hat{x}_i + (1 - x_i)\log(1 - \hat{x}_i)\Big], \tag{2}$$

where n is the maximum number of holes, x_i denotes the ith HV value of the training data, and $\hat{x}_i$ is the output of DNN as the corresponding estimate of x_i. The predicted HVs $\hat{x}_i$ can take any value from 0 to 1 from a Bernoulli

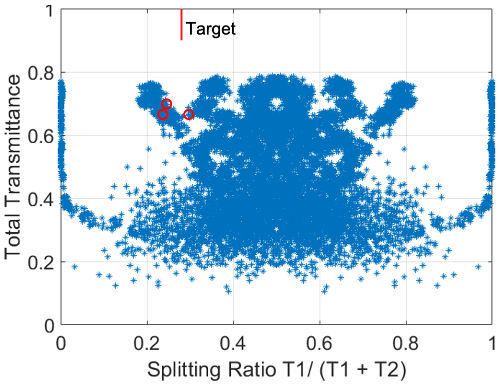

Figure 2: Total transmittance of 22,000 training data as a function of the splitting ratio. The red circles indicate the three initial conditions used for training the forward model, and the line shows the target splitting ratio.[15]

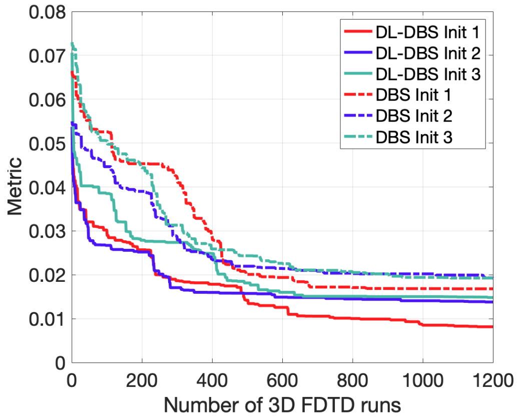

Figure 3: Metric as a function of the number of 3D FDTD runs, for the conventional DBS and the DNN-assisted DBS methods. Three different initial conditions are used.[15]

distribution classifier. The classification tends to converge to either 0 or 1 as the loss reduces by increasing the number of training epochs. We train the network using the same data as used in Section 4.

To test the generalization capabilities of the network, we investigate the design performance on arbitrary and unseen target cases. We try to optimize the device at around a target splitting ratio of 0.27 as an example, where there are no good samples in the training dataset, as shown in Fig. 4. In the design stage, the target splitting ratios are chosen to be $0.23, 0.24, \ldots, 0.32$, and the total transmittances are $0.80, 0.82, 0.84$, and 0.86, and hence we obtain 40 combinations of input spectrum to feed in the inverse-modeling DNN. The DNN output are quantized with a threshold of 0.5, and the binary sequence is then fed back into the FDTD solver. The results are shown as the red circles in Fig. 4. In this first round, some data points fill the gap, while many are overlapping with the original training data clouds.

In the second run, we add these 40 new data points to the training data, retrain the network, and repeat this active learning process. The results are shown as the dark blue triangles, and those from the third round are indicated by the green squares in Fig. 4. These results show that the inverse modeling has the capability of generating the unseen results out of the training data, and the results further improve with active learning.

In this process, the training takes about half an hour at each round on a high-performance computer with a GPU board. Once trained, the inverse design process is instantaneous (less than 1 s) for 40 devices.

6. GENERATIVE MODELING

Another methodology based on generative modeling has been proposed for photonic devices including metasurfaces and plasmonics.[19,20,25–27] The generative network can produce a series of improved designs from the training data, based on random number sampling, in a more explicit and systematic way. We applied generative deep learning models based on an improved version of convolutional variational autoencoder (CVAE) to generate integrated photonics devices,[21] and later added a concept of active learning to further improve the performance.[22]

The CVAE network models the distribution of the device topologies associated with target spectrum characteristics. In our power splitter application, we define the topology pattern using variable-size holes (rather than binary holes).

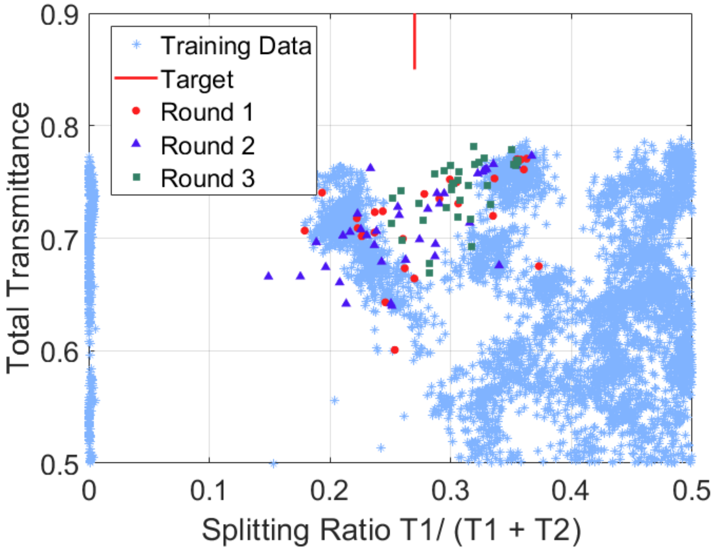

Figure 4: Demonstration of inverse design for a power splitter to fill gap around a splitting ratio of 0.27. The light blue asterisks, red circles, dark blue triangles, and green squares the training data, the first, the second, and the third round results, where the total transmittance is plotted against the splitting ratio. The red line denotes the target splitting ratio of 0.27.[15]

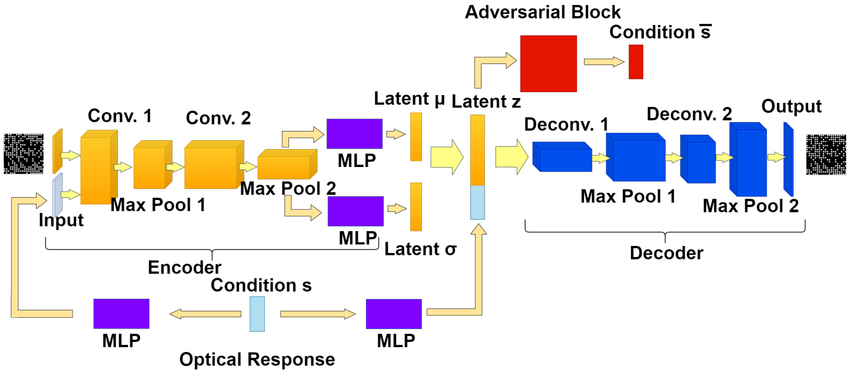

Figure 5: The A-CVAE model structure[22]

We constructed a new generative deep learning model based on a CVAE[19,28] and an adversarial block.[20,29] The issue of a conventional CVAE is that the latent variables tend to be correlated with the condition SPEC s, which will result in degradation of the device performance for the pattern generation because random latent space sampling can adversely impact the target spectra. To address this issue, an Adversarial CVAE (A-CVAE) is introduced as shown in Fig. 5, where a separate branch to the adversary block is used for isolating the latent variable z from the nuisance variations s (the target SPEC).[30–32] Our A-CVAE model also has two convolutional layers both for the encoder and the decoder networks(similar to the CVAE model). The encoder is followed by one fully connected layer to obtain the latent variables. The latent variables are then concatenated with the SPEC performance data and is fed into the decoder to generate the output HV. The validations are calculated by using the FDTD simulation to verify a figure of merit (FOM) of generated patterns. The FOM is calculated by:

$$\mathsf{FOM} = 1 - 10 \sum_{\lambda=\lambda_{\min}}^{\lambda_{\max}} \Big[\big|T_1(\lambda) - T_1^\star(\lambda)\big|^2 + \big|T_2(\lambda) - T_2^\star(\lambda)\big|^2 + \alpha \big|R(\lambda) - R^\star(\lambda)\big|^2 \Big], \tag{3}$$

where $\alpha = 4$ is used as a weighting factor to balance between the contributions from transmission and the reflection. We take the average of FOM over the FDTD simulation spectral range. As the SPEC performance approaches the target, the FOM increases towards 1, in which case we obtain an ideal power splitter without excess loss for $R^\star(\lambda) = 0$ and $T_1^\star(\lambda) + T_2^\star(\lambda) = 1$.

We use an encoder structure feeding the performance SPEC s projected on a 20×20 matrix which is combined with the original 20×20 hole vector to form a 2-channel input, then process it through two convolution layers. In A-CVAE, the latent z variable will also be fed into an adversarial block to estimate the SPEC $\bar{s} := (\bar{s}_1, \ldots, \bar{s}_n)$. The loss function for the A-CVAE model is shown as follows:

$$\begin{aligned} \mathsf{Loss} = &- \sum_{i=1}^{n} \Big[y_i \log x_i + (1 - y_i) \log(1 - x_i) \Big] \\ &+ \frac{1}{2} \sum_{j=1}^{J} \Big[\mu_{zj}^2 + \sigma_{zj}^2 - \log(\sigma_{zj}^2) - 1 \Big] - \frac{\beta}{n} \sum_{i=1}^{n} (s_i - \bar{s}_i)^2. \end{aligned} \tag{4}$$

The loss function has three parts. The first loss function term is the VAE reconstruction loss in the BCE criterion and the second term is the Kullback–Leibler (KL) divergence. The last term is a regularization term which is the MSE loss of the adversarial block. Since the condition information contained in the latent variable z shall be minimized, the MSE loss between s and $\bar{s}$ needs to be maximized. A complete update of the network generally requires alternating updates in two iteration cycles. The first iteration updates the CVAE model based on the loss function in (4). The second iteration updates the adversarial block solely based on the MSE loss between s and $\bar{s}$. The total training time using a computer with a GPU board is around 5 minutes long.

The initial training data do not contain any data close to the splitting ratios of 6 : 4, 7 : 3, and 8 : 2, and it is challenging to generate very good designs around these splitting ratios. So we employ an active learning, i.e., for the second round of training, we use the new data generated from the first trained model. When we train the second model, the original BCE is replaced with the MSE loss since the training data now contains non-binary data. After training the proposed machine learning model, we test the decoder output of the final A-CVAE model by sampling random latent variable z combined with different splitting conditions to generate nanopatterned power splitter devices according to the generated HVs. Fig. 6 shows the comparison of the performance among the devices generated by the CVAE model, by the A-CVAE model without active learning, and by the A-CVAE model with active learning, for four different splitting ratios (5 : 5, 6 : 4, 7 : 3, 8 : 2). The FOM is calculated for 20 randomly generated devices from the trained CVAE, A-CVAE models, and A-CVAE models with active learning. This figure shows that the adversarial censoring and active learning generates devices with much better performance across a very broad bandwidth (from 1250μm to 1800μm), compared with the conventional CVAE model. The devices generated by our final A-CVAE model with active learning can fit the target splitting ratio better with excellent total transmission. The average FOMs for the CVAE model, A-CVAE model, and A-CVAE model with active learning are 0.771, 0.888, and 0.9009, respectively.

Figure 7 shows the results generated by the A-CVAE model along with the FDTD verification. The solid lines show the transmission and reflection spectra for the four types of devices that are generated by the A-CVAE model. The dashed lines show the transmission and the reflection for the best devices in the training data. The generated devices for 6 : 4, 7 : 3 and 8 : 2 power splitting ratios have significantly improved transmission across a wide range of wavelengths compared to the best training data for those splitting ratios. The generated device for the 5 : 5 power splitting ratio is close to the best training data because the training data already include near optimal solution of that specific power splitting ratio.

Therefore, it is demonstrated that the adversarial censoring and especially with active learning further improves the performance of the model with high stability.

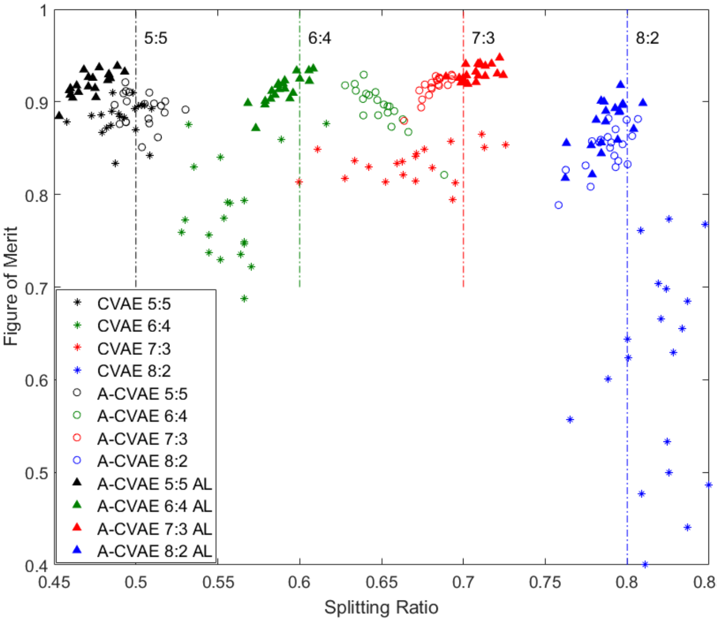

Figure 6: FOM comparison for different CVAE models: conventional CVAE (star marker), A-CVAE (round marker) and A-CVAE with active learning (triangle marker). Four different splitting ratios are used as a target value to test the model performance (marked with dashed lines). The devices generated by the active learning assisted CVAE model can fit the target splitting ratio better with excellent total transmission. The average FOM for the three models are: 0.771, 0.888, 0.901, respectively.

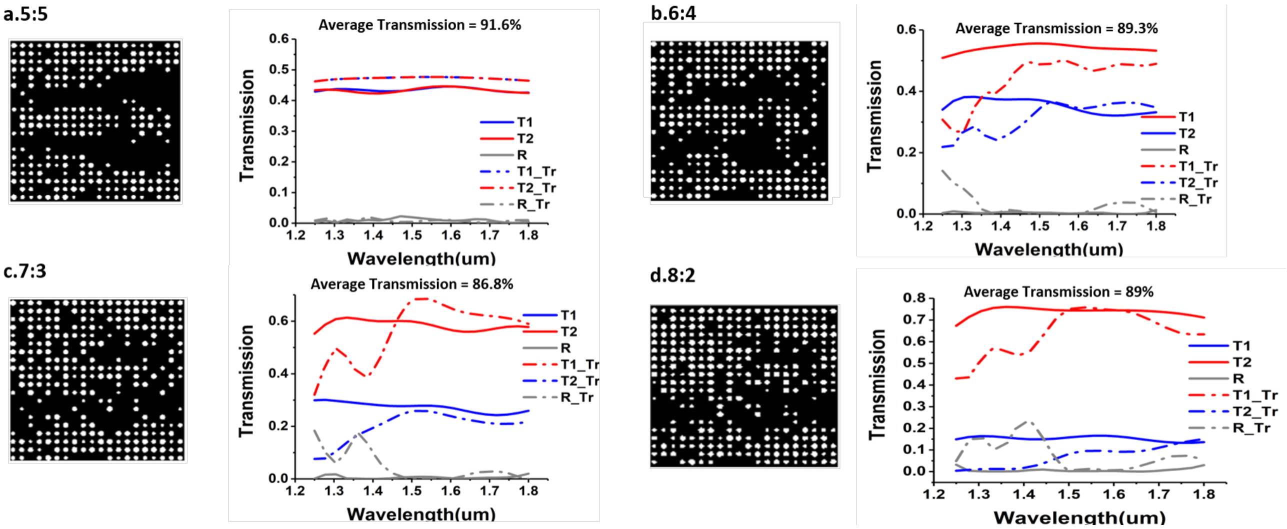

Figure 7: FDTD results (transmission and reflection) of the generated patterns via the active learning-assisted A-CVAE model. The dashed lines show the results of the best training devices around the target splitting ratio.

7. DISCUSSIONS

The adjoint method is widely used in the inverse design of nanophotonic devices.[8,33,34] Given an ideal initial condition for parameters, the optimization process can be efficiently done in a small number of iterations (tens of FDTD simulations). However, the initial condition needs to be carefully chosen in order to obtain the optimal result or many initial conditions need to be tried. DNNs are different in many sense. They are trained from a library (training data) of FDTD simulation results, which may come from previous imperfect optimization results with multiple target conditions (splitting ratio and bandwidth, in our case). Then the DNNs, especially the inverse and the generative models, will try to learn/generalize from the library, and generate a series of improved results for a given condition. Once the model is trained, inverse designs for multiple conditions can be generated in almost no time. Further FDTD simulations are not required, and can be used only for verification purposes.

8. CONCLUSION

We have reviewed three different ways of using DNNs for designing nanophotonic devices: forward, inverse, and generative modeling.

DNNs can be used to take device structure data (shape, depth, and permittivity) to predict the optical response of the nanostructure in the forward modeling framework. In this case DNNs, can be used as a viable counterpart for fast approximation of the optical response, in comparison to the use of computationally heavy FDTD simulations.

Another way to use DNNs is taking an optical response to provide the user with an approximate solution of nanostructure in the inverse modeling framework, which does not rely on external meta-heuristic optimizers unlike forward modeling. Although the DNN initially needs a sufficient amount of data for the training purpose, it is possible to process several heuristic optimization metrics in parallel on a computing cluster to speed up generating the training data. We can design the nanostructured geometry in a fraction of second once the network is trained to represent the topology as optical response and vice versa.

The generative model, on the other hand, implicitly integrates forward and inverse models. Once the network is trained, the generation of new designs takes practically little time. The use of adversarial censoring further improves the design capability.

Overall, DNNs have the capability of learning the training data with a high degree of generalizability.

REFERENCES

[1] Ni, X., Wong, Z. J., Mrejen, M., Wang, Y., and Zhang, X., "An ultrathin invisibility skin cloak for visible light," *Science* **349**(6254), 1310–1314 (2015).

[2] Alù, A. and Engheta, N., "Achieving transparency with plasmonic and metamaterial coatings," *Physical Review E* **72**(1), 016623 (2005).

[3] Monticone, F., Estakhri, N. M., and Alù, A., "Full control of nanoscale optical transmission with a composite metascreen," *Phys. Rev. Lett.* **110**, 203903 (May 2013).

[4] Arbabi, E., Arbabi, A., Kamali, S. M., Horie, Y., and Faraon, A., "Multiwavelength polarization-insensitive lenses based on dielectric metasurfaces with meta-molecules," *Optica* **3**(6), 628–633 (2016).

[5] Khorasaninejad, M., Chen, W. T., Devlin, R. C., Oh, J., Zhu, A. Y., and Capasso, F., "Metalenses at visible wavelengths: Diffraction-limited focusing and subwavelength resolution imaging," *Science* **352**(6290), 1190–1194 (2016).

[6] Krasnok, A., Tymchenko, M., and Alù, A., "Nonlinear metasurfaces: A paradigm shift in nonlinear optics," *Materials Today* **21**(1), 8 – 21 (2018).

[7] Azad, A. K., Kort-Kamp, W. J., Sykora, M., Weisse-Bernstein, N. R., Luk, T. S., Taylor, A. J., Dalvit, D. A., and Chen, H.-T., "Metasurface broadband solar absorber," *Scientific Reports* **6**(20347) (2016).

[8] Lalau-Keraly, C. M., Bhargava, S., Miller, O. D., and Yablonovitch, E., "Adjoint shape optimization applied to electromagnetic design," *Optics express* **21**(18), 21693–21701 (2013).

[9] Motayed, A., Aluri, G., Davydov, A. V., Rao, M. V., Oleshko, V. P., Bajpai, R., Zaghloul, M. E., Thomson, B., Wen, B., Xie, T., et al., "Highly selective nanostructure sensors and methods of detecting target analytes," (May 29 2018). US Patent 9,983,183.

[10] Silva, A., Monticone, F., Castaldi, G., Galdi, V., Alù, A., and Engheta, N., "Performing mathematical operations with metamaterials," *Science* **343**(6167), 160–163 (2014).

[11] Chu, Z., Liu, Y., Sheng, J., Wang, L., Du, J., and Xu, K., "On-chip optical attenuators designed by artifical neural networks," in [*2018 Asia Communications and Photonics Conference (ACP)*], 1–3, IEEE (2018).

[12] Liu, Z., Liu, X., Xiao, Z., Lu, C., Wang, H.-Q., Wu, Y., Hu, X., Liu, Y.-C., Zhang, H., and Zhang, X., "Integrated nanophotonic wavelength router based on an intelligent algorithm," *Optica* **6**(10), 1367–1373 (2019).

[13] Teng, M., Honardoost, A., Alahmadi, Y., Polkoo, S. S., Kojima, K., Wen, H., Renshaw, C. K., LiKamWa, P., Li, G., Fathpour, S., et al., "Miniaturized silicon photonics devices for integrated optical signal processors," *Journal of Lightwave Technology* **38**(1), 6–17 (2019).

[14] Kojima, K., Wang, B., Kamilov, U., Koike-Akino, T., and Parsons, K., "Acceleration of FDTD-based inverse design using a neural network approach," in [*Integrated Photonics Research, Silicon and Nanophotonics*], ITu1A–4, Optical Society of America (2017).

[15] Kojima, K., Tahersima, M. H., Koike-Akino, T., Jha, D., Tang, Y., Wang, Y., and Parsons, K., "Deep neural networks for inverse design of nanophotonic devices (invited)," *Journal of Lightwave Technology* **39**(4), 1010–1019 (2021).

[16] Teng, M., Kojima, K., Koike-Akino, T., Wang, B., Lin, C., and Parsons, K., "Broadband SOI mode order converter based on topology optimization," in [*2018 Optical Fiber Communications Conference and Exposition (OFC)*], 1–3 (March 2018).

[17] Tahersima, M. H., Kojima, K., Koike-Akino, T., Jha, D., Wang, B., Lin, C., and Parsons, K., "Deep neural network inverse design of integrated photonic power splitters," *Scientific Reports* **9**(1), 1368 (2019).

[18] Sanchez-Lengeling, B. and Aspuru-Guzik, A., "Inverse molecular design using machine learning: Generative models for matter engineering," *Science* **361**(6400), 360–365 (2018).

[19] Ma, W., Cheng, F., and Liu, Y., "Deep-learning enabled on-demand design of chiral metamaterials," *ACS Nano* **12**(6), 6326–6334 (2018).

[20] Liu, Z., Zhu, D., Rodrigues, S. P., Lee, K.-T., and Cai, W., "Generative model for the inverse design of metasurfaces," *Nano letters* **18**(10), 6570–6576 (2018).

[21] Tang, Y., Kojima, K., Koike-Akino, T., Wang, Y., Wu, P., Tahersima, M. H., Jha, D., Parsons, K., and Qi, M., "Generative deep learning model for a multi-level nano-optic broadband power splitter," in [*2020 Optical Fiber Communications Conference and Exhibition (OFC)*], Th1A.1 (March 2020).

[22] Tang, Y., Kojima, K., Koike-Akino, T., Wang, Y., Wu, P., Xie, Y., Tahersima, M. H., Jha, D. K., Parsons, K., and Qi, M., "Generative deep learning model for inverse design of integrated nanophotonic devices," *Laser & Photonics Reviews* **14**(12), 2000287 (2020).

[23] Lu, L., Zhang, M., Zhou, F., and Liu, D., "An ultra-compact colorless 50: 50 coupler based on PhC-like metamaterial structure," in [*Optical Fiber Communications Conference and Exhibition (OFC), 2016*], 1–3, IEEE (2016).

[24] Piggott, A. Y., Petykiewicz, J., Su, L., and Vučković, J., "Fabrication-constrained nanophotonic inverse design," *Scientific Reports* **7**(1), 1786 (2017).

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

[26] Ma, W., Cheng, F., Xu, Y., Wen, Q., and Liu, Y., "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy," *Advanced Materials* **31**(35), 1901111 (2019).

[27] An, S., Zheng, B., Tang, H., Shalaginov, M. Y., Zhou, L., Li, H., Gu, T., Hu, J., Fowler, C., and Zhang, H., "Multifunctional metasurface design with a generative adversarial network," *Advanced Optical Materials* , 2001433 (2021).

[28] Sohn, K., Lee, H., and Yan, X., "Learning structured output representation using deep conditional generative models," in [*Advances in neural information processing systems*], 3483–3491 (2015).

[29] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," in [*Advances in neural information processing systems*], 2672–2680 (2014).

[30] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., and Ranzato, M., "Fader networks: Manipulating images by sliding attributes," in [*Advances in neural information processing systems*], 5967–5976 (2017).

[31] Wang, Y., Koike-Akino, T., and Erdogmus, D., "Invariant representations from adversarially censored autoencoders," *arXiv preprint arXiv:1805.08097* (2018).

[32] Özdenizci, O., Wang, Y., Koike-Akino, T., and Erdoğmuş, D., "Transfer learning in brain-computer interfaces with adversarial variational autoencoders," in [*2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*], 207–210, IEEE (2019).

[33] Piggott, A. Y., Lu, J., Lagoudakis, K. G., Petykiewicz, J., Babinec, T. M., and Vučković, J., "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," *Nature Photonics* **9**(6), 374–377 (2015).

[34] Wang, K., Ren, X., Chang, W., Liu, L., Liu, D., and Zhang, M., "Inverse design of digital nanophotonic devices using the adjoint method," *Photonics Research* **8**, 528–533 (2020).