SCIENCE CHINA
Physics, Mechanics & Astronomy

•Article•
Special Topic: Metasurfaces of Novel Designs and Functionalities

August 2020 Vol. 63 No. 8: 284212
https://doi.org/10.1007/s11433-020-1575-2

A data-efficient self-supervised deep learning model for design and characterization of nanophotonic structures

Wei Ma[1], and Yongmin Liu[2,3*]

[1] College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China;
[2] Department of Mechanical and Industrial Engineering, Northeastern University, Massachusetts 02115, USA;
[3] Department of Electrical and Computer Engineering, Northeastern University, Massachusetts 02115, USA

Received March 3, 2020; accepted May 7, 2020; published online June 22, 2020

With its tremendous success in many machine learning and pattern recognition tasks, deep learning, as one type of data-driven models, has also led to many breakthroughs in other disciplines including physics, chemistry and material science. Nevertheless, the supremacy of deep learning over conventional optimization approaches heavily depends on the huge amount of data collected in advance to train the model, which is a common bottleneck of such a data-driven technique. In this work, we present a comprehensive deep learning model for the design and characterization of nanophotonic structures, where a self-supervised learning mechanism is introduced to alleviate the burden of data acquisition. Taking reflective metasurfaces as an example, we demonstrate that the self-supervised deep learning model can effectively utilize randomly generated unlabeled data during training, with the total test loss and prediction accuracy improved by about 15% compared with the fully supervised counterpart. The proposed self-supervised learning scheme provides an efficient solution for deep learning models in some physics-related tasks where labeled data are limited or expensive to collect.

nanophotonics, metasurfaces, self-supervised, deep learning

PACS number(s): 78.67.-n, 78.20.Bh, 07.05.Mh

Citation: W. Ma, and Y. Liu, A data-efficient self-supervised deep learning model for design and characterization of nanophotonic structures, Sci. China-Phys. Mech. Astron. **63**, 284212 (2020), https://doi.org/10.1007/s11433-020-1575-2

1 Introduction

Nanophotonics, the study of light-matter interactions at the nanoscale, enables us to control the flow of light within the dimension far below the optical wavelength. It spawns a plethora of novel applications, such as miniaturized flat optics [1], perfect absorption [2], sub-diffraction-limited imaging [3] and extreme light concentration [4]. In all these applications, artificially designed structures play a crucial role in engineering the light-matter interactions [5,6]. Conventionally, such artificial structures as metamaterials/metasurfaces, photonic crystals and plasmonic nano-structures are designed based on expert experience. Certain empirical templates are used as initial guess, from which a limited set of design parameters are adjusted to optimize the design, either by analytical models, semi-analytical models or numerical simulations. To overcome this inefficient trial-and-error procedure, inverse design methods are proposed to better exploit larger degrees of freedom in the design space [7]. However, traditional inverse design approaches, either gradient-based or gradient-free, are still restricted in runtime speed, because the iterative optimization steps in these algorithms rely on numerical calculations to evaluate the cost function.

*Corresponding author (email: y.liu@northeastern.edu)

Recently, in light of the revolutionary revival of deep learning and its unprecedented success in machine learning and pattern recognition tasks [8], researchers are actively seeking solutions from deep learning for the challenging tasks in other disciplines, including materials science [9], chemistry [10], laser physics [11], particle physics [12] and quantum mechanics [13]. As a data-driven method, deep learning allows a computational model composed of multiple layers of processing unit to learn multiple levels of abstraction in given data, thus circumventing a direct interaction with human intervention or underlying physical laws in a conventional optimization problem. For the study of nanophotonics, on one hand, well-trained deep learning models can be used as a fast simulator to predict the optical response of a structure. On the other hand, it can also function as an optimization tool for the inverse design of structures from given requirements in various scenarios, including topological photonics [14], integrated silicon photonics [15,16], color generation from nanostructures [17,18], metamaterials/metasurfaces [19-23], plasmonics [24-29], and photonic crystals [30].

At present, most of the studies on deep learning for nanophotonics are developed under the supervised learning routine, where enormous training data comprising pairs of photonic designs and corresponding optical responses are collected in advance. The collection of training data requires numerical simulations or experimental measurements of individual structures, which remains the bottleneck of training deep learning models, since the quantity of training data is crucial in data-driven models to guarantee their performances. To alleviate the burden of data acquisition, unsupervised learning or semi-supervised learning strategies can be adopted to assist the model in discovering patterns in data [31], where the unlabeled data, usually empirical design patterns for nanophotonic structures, can be conveniently obtained with little effort [23]. However, the unlabeled data, which is also collected beforehand and fixed during training, need to be carefully chosen with appropriate quantities to balance training accuracy and efficiency. Other solutions to reduce the dependence on labeled data include physics-informed modeling [32] or transfer learning [33], resorting to learning from physical laws or other relevant tasks. Nevertheless, these approaches rely on explicit embedding of physical laws or surrogate learning tasks, which are not often applicable to a general end-to-end deep learning model for specific photonics structures like metamaterials or metasurfaces.

In this paper, we propose an efficient deep learning model for the design and characterization of nanophotonic structures with limited availability of labeled data. The key idea is to employ a self-supervised learning strategy, where unlabeled data are generated in an online manner and fed to the model in company with pre-collected labeled data during the training process. The training data contain empirical design geometries like arc, bowtie, cross, split ring, H-shape, rectangle, L-shape and ellipse, which are represented as two-dimensional images in general with randomly varied design parameters for each geometry. For instance, the arm width, length and rotation angle are varied for a cross resonator. With online training, some of the training data are not predefined but dynamically supplemented as unlabeled data when the training proceeds, enabling the model to learn from human experience as much as possible. After the model is fully trained, it can predict the optical response of a photonic structure in arbitrary shapes and retrieve possible designs from a given optical response at the same time. The dynamically generated unlabeled designs, albeit not able to provide direct supervision on the optical response, can help to significantly improve the prediction accuracy via deliberately designed model architecture and loss functions. In a comparative study, by introducing the self-supervised learning mechanism, the model performance surpasses its counterparts with either fully supervised or semi-supervised learning strategy. Leveraging only a limited number of labeled data and human experience on empirical designs, the proposed deep learning model creates a new framework for data-efficient design and characterization of nanophotonic structures in arbitrary geometries.

2 Model architecture

The proposed deep learning model with the self-supervised learning mechanism is schematically shown in Figure 1(a). The basic framework of the model is a mutant of a variational auto-encoder (VAE), comprising an encoder and a decoder to realize both the forward prediction of the optical responses of a certain structure and the inverse design of nanophotonic structures from predesignated optical properties. The nanophotonic structure under investigation is metasurfaces, which are two-dimensional metamaterials with thickness of only a few tens of nanometers [34,35]. By tailoring the geometry of the building blocks of metasurfaces and engineering their spatial distribution, we can realize the full control of the amplitude, polarization, phase and trajectory of light. Many optical functionalities have been demonstrated based on metasurfaces, including ultra-thin lenses [36-39], wave plates [40,41], beam deflection [42,43] and holograms [44,45]. Metasurfaces have also been used to manipulate near-field evanescent waves [46-50].

The designed metasurface works in the reflective configuration, which is composed of a continuous metallic ground plane at bottom, a dielectric spacer in the middle and metallic resonators on top. Aiming at an operating frequency in the mid-infrared region (from 40 to 100 THz), we choose the thickness of the resonator and spacer to be 50 and 100 nm

respectively, and a lattice constant of 2 μm. The design object is the top resonator. It is represented as a two-dimensional binary image with pixels of 64×64, in which 1 and 0 represent metal and air, respectively. We consider eight representative geometries for the resonator, including arc, bowtie, cross, split ring, H-shape, rectangle, L-shape and ellipse, as shown in the inset of Figure 1(a). Due to the optical reciprocity, we only need to consider one cross-polarization. Therefore, the optical response of the meta-atom is fully described by three reflection spectra, namely, R_{xx} (x-polarization in and x-polarization out), R_{yy} (y-polarization in and y-polarization out), and R_{xy} (x-polarization in and y-polarization out). Each spectrum is discretized into 61 data points in a step of 1 THz.

Generally, the model follows a VAE configuration, where the encoder compresses the input design patterns into a latent space and then the decoder reconstructed the input design from the latent variable sampled from the latent space conditioned on the corresponding reflection spectra [23,51]. To holistically optimize the forward prediction and inverse de-

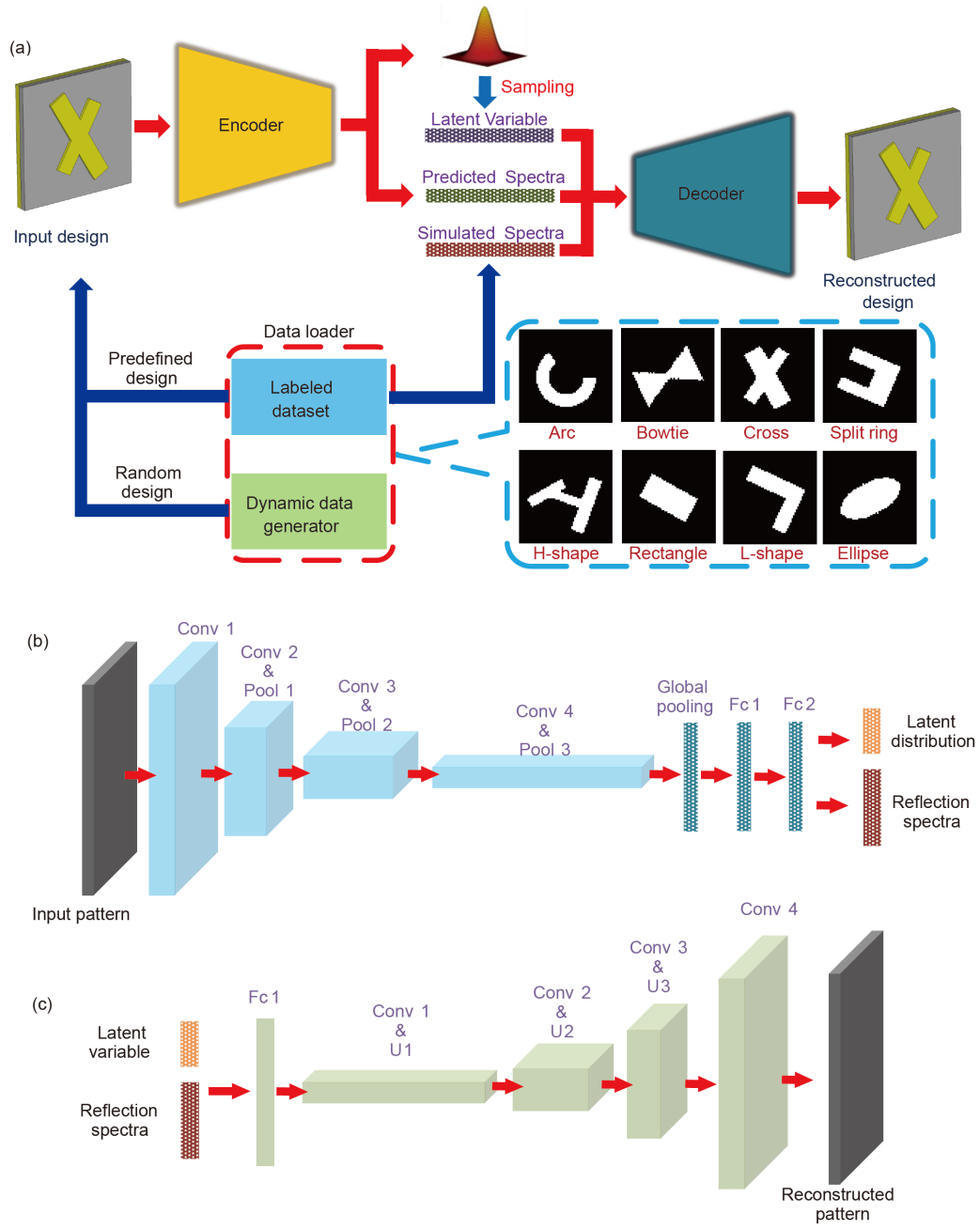

Figure 1 (Color online) (a) The proposed deep learning model with self-supervised learning mechanism for both the forward prediction and inverse design of nanophotonic structures. The network architecture of encoder (b) and decoder (c). Conv stands for the convolutional block containing three convolution operations with kernel size of 1×1, 3×3 and 1×1, respectively, each followed by a batch-normalization layer. Pool denotes the pooling layer to halve the lateral dimension while U denotes the up-sampling layer to double the lateral dimension. Fc stands for the fully connected layer.

sign capability, the model is trained in an end-to-end manner by minimizing the following loss terms. Denoting the input design pattern as x, the corresponding spectra as y and latent variable as z, we have a VAE loss as [52]:

$$L_{\mathrm{VAE}} = L_{\mathrm{KL}} + L_{\mathrm{recon}} = KL\left[q_\phi(z \mid x) \,\middle\|\, P(z \mid x)\right] - E_{q_\phi(z|x)}\left[\log P_\theta(x \mid y, z)\right], \quad (1)$$

where the probability density function $q_\phi(z|x)$ is the encoder parameterized by learnable parameters ϕ using a deep neural network, while $P_\theta(x|y, z)$ is the decoder parameterized by learnable parameters θ using another deep neural network. $P(z|x)$ is the true posterior distribution of the latent variable z, which in our case, is simplified to a standard Gaussian distribution $N(0, I)$ of a dimension of 20. The 20-dimensional latent space is expressive enough for encoding the 64×64 image, while enabling effective decoding together with the reflection spectra. The probability density $q_\phi(z|x)$ is also chosen to be an isotropic Gaussian with the same dimension as $P(z|x)$, whose mean vector and diagonal covariance matrix are predicted by the encoder network depending on the input pattern x. Therefore, the VAE loss of our model consists of the KL divergence loss L_{KL} and the reconstruction loss L_{recon}, which is formulated as the negative log-likelihood of the reproduced input pattern at the output of the decoder. The KL divergence loss measures how the encoder-generated distribution is different from the assumed prior distribution of z, while the reconstruction loss indicates how accurate the decoder can replicate the input pattern from its latent code and corresponding reflection spectra.

To account for the forward prediction ability of the proposed model, we allow the encoder to output the reflection spectra of the input pattern in addition to the parameters of its latent distribution. Then another prediction loss, L_{pred}, is added to measure the discrepancy of the predicted spectra y and the ground-truth spectra $\hat{y}$ by their mean squared error (MSE).

$$L_{\mathrm{pred}} = (y - \hat{y})^2. \quad (2)$$

Therefore, the total loss to train the deep learning model is given by

$$L_{\mathrm{total}} = L_{\mathrm{KL}} + L_{\mathrm{recon}} + \alpha L_{\mathrm{pred}}, \quad (3)$$

where a weight factor α=10^5 is introduced to balance the VAE loss and prediction loss. We use convolutional neural networks (CNN) to construct both the encoder and decoder. The detailed model architectures are illustrated in Figure 1(b) and (c). We first create the basic convolution block, which is composed of three convolution layers with the kernel size of 1×1, 3×3, and 1×1, respectively, each followed by a batch-normalization layer before fed to rectified linear unit (Relu) activation. For the encoder network, the input design pattern passes through alternating layers of four convolution blocks and three pooling layers to reduce the size from 64×64 to 8×8. Then a global average pooling layer converts the feature maps into a vector, before further processed by two fully connected layers. The first output of the decoder is mean vector and diagonal covariance matrix of the latent distribution, each in the size of 20. The second output of the decoder is the reflection spectra with the size of 61×3. The architecture of the decoder network follows roughly the reversed configuration compared with the encoder. The decoder takes the latent variable, sampled from the 20-dimensional Gaussian distribution obtained from the encoder, together with the reflection spectra as input, and gradually increases the output size to 64×64 by alternating layers of four convolution blocks and three up-sampling layers.

For each of the eight representative geometries, we use Monte Carlo sampling to vary the design parameters, and then perform numerical simulations on the samples to collected 2000 labeled data. The 16000 data in total constitute our labeled training dataset. Similarly, we also collect 1000 data for each geometry as the test dataset. To introduce the self-supervised learning into the model, we incorporate a dynamic data generator in the data loader. During training, each data in a batch is either randomly sampled from the labeled dataset or online generated by the dynamic data generator, with equal probability of 50%. The encoder takes all the patterns, regardless of their origins, to output the latent distribution and the predicted spectra. When data flow to the decoder, the training samples from labeled dataset use the ground-truth spectra as the input to the decoder, while the dynamically generated samples use the predicted spectra from the encoder instead. In other words, for unlabeled data, the decoder takes the predicted spectra from the encoder as pseudo-labels. In this way of dealing with unlabeled data, the model is actually supervised by the encoder of itself. Such self-supervised learning mechanism is widely studied in other tasks like image classification, which uses intermediate output from the model or surrogate tasks as the supervisory signal [53]. Considering the loss terms in eq. (3), unlabeled data only contribute to the VAE loss but neglect prediction loss. In the engineering implementation, when feeding a batch of training data to the model, we use the encoder predicted spectra as the fake ground-truth together with the real ground-truth for labeled data. In this way, the prediction loss for unlabeled data is always zero during training. However, as the network weights are shared when VAE loss and prediction loss back propagate from the decoder to encoder, the unlabeled data under such self-supervised learning configuration can help to improve the overall model performance.

3 Results and discussion

To systematically investigate the function of the proposed

self-supervised learning mechanism, we make a comparative study of four different models, namely, the model trained using only labeled data (U0), the model trained using a fixed set of unlabeled data with the ratio of total number of unlabeled data to that of labeled data to be 1:1 (U1) and 4:1 (U4), and the model trained with a dynamic unlabeled data generator (U_dynamic). During the training process, we use an initial learning rate of 0.01 and a reduce-on-plateau update strategy. Specifically, if the validation loss no longer decreases in 10 consecutive epochs, we reduce the learning rate to 10% of the original one. In addition, we apply an early stopping strategy to prevent overfitting, which terminate the training if the validation loss no longer decreases in 20 consecutive epochs. As illustrated in the left panel of Figure 2(a), the total test loss of model U_dynamic is smaller than that of the other three models, especially with a decrease of 14.4% compared with fully supervised model U0, indicating that the self-supervised learning indeed helps to improve the overall performance and generalization ability of the model. On the right panel of loss evolution over training steps, we notice that by introducing self-learning mechanism, the loss drops faster compared with the case of fully supervised learning. With self-supervised learning using dynamically generated data, the loss takes more training epochs before meeting the stop criteria, yielding a smaller loss value at last. The longer training time is naturally caused by the random unlabeled data fed to the model, where the stochastic variations of the unlabeled data in each batch force the model to generalize better and thus converge slower.

In Figure 2(b) and (c), we plot the reconstruction loss and prediction loss of the four different models. We can observe that by using unlabeled data, the reconstruction loss is always smaller than the case that only uses labeled data. This is because unlabeled data, which provide more similar design patterns as the labeled dataset but without corresponding reflection spectra, directly contribute to the reconstruction loss according to eq. (1). As for the prediction loss, the self-learning mechanism learns useful information by a weight

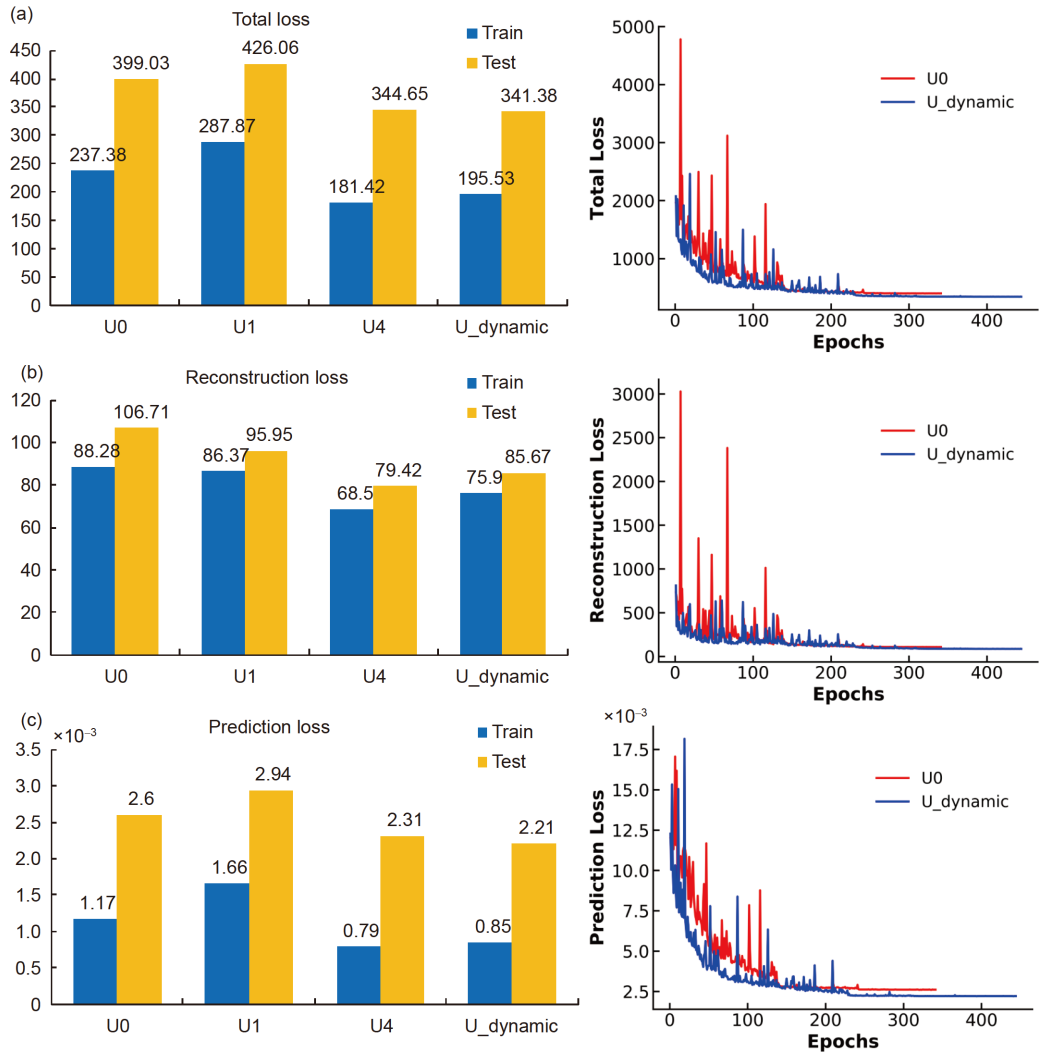

Figure 2 (Color online) The total loss (a), reconstruction loss (b), and prediction loss (c) of four models (left), together with the loss evolution with training epochs (right).

sharing of the encoder when dealing with labeled and unlabeled data, so model U_dynamic reduces the prediction error by 15% compared with model U0.

To get a more concrete picture of how unlabeled data improve prediction accuracy under the self-supervised learning scheme, we pick three design patterns from the test dataset, namely bowtie, ellipse and split ring. As illustrated in Figure 3, we plot the forward prediction results (hollow circles) of three test samples by model U0 without self-supervised learning and model U_dynamic with self-supervised learning. The ground-truth reflection spectra for each sample are plotted as solid curves as references. Obviously, by introducing self-supervised learning, model U_dynamic produces more accurate prediction of the reflection spectra from the metasurface, especially around resonance frequencies. This result manifests that during the training process, unlabeled data without corresponding ground-truth spectra can still contribute to improving the accuracy of forward prediction with the aid of self-supervised learning.

Apart from the decrease in total loss and improvement in forward prediction accuracy, the benefit of introducing self-supervised learning can also be verified by checking the structure of the latent space. Through the encoding process, the nanophotonic structural design is compressed into a 20-dimensional latent space, which is expected to contain useful information about the meta-atom. To visualize the latent space, we use t-distributed stochastic neighbor embedding (t-SNE) method to reduce the dimension to 2 while keeping the relative spatial relationship among data. In Figure 4(a) and (b), we plot the 2D distribution of the test data from the eight geometry groups (i.e., arc, bowtie, cross, split ring, H-shape, rectangle, L-shape and ellipse), encoded by model U0 and model U_dynamic, respectively. In both cases, the designs are clearly separated into eight clusters according to the design geometries in training dataset. Although no labels for the geometries are provided during training, the models can automatically learn to distinguish the eight geometries through the encoding-decoding process. However, the clustering effect of model U_dynamic is apparently better than that of model U0, with smaller intra-class variations and larger inter-class gaps. The better clustering results confirm that, aided with self-supervised learning, the model can discover more intricate features among different geometries and separate them further apart into isolated clusters. It is noteworthy that the encoder does not take any information about geometry classes as input, and the clustering is merely a result of the probabilistic learning procedure. So not surprisingly, similar geometries may have a large portion of overlap in the latent space, such as bowtie and H-shape in Figure 4(b).

To test the inverse design ability of the proposed model, we use the decoder of model U_dynamic to produce possible designs with given requirements in a probabilistically generative manner. In this way, we can get multiple retrieved designs from the same given requirements in the reflection spectra, thus solving the one-to-many mapping issue in photonic designs that is intractable for a deterministic model.

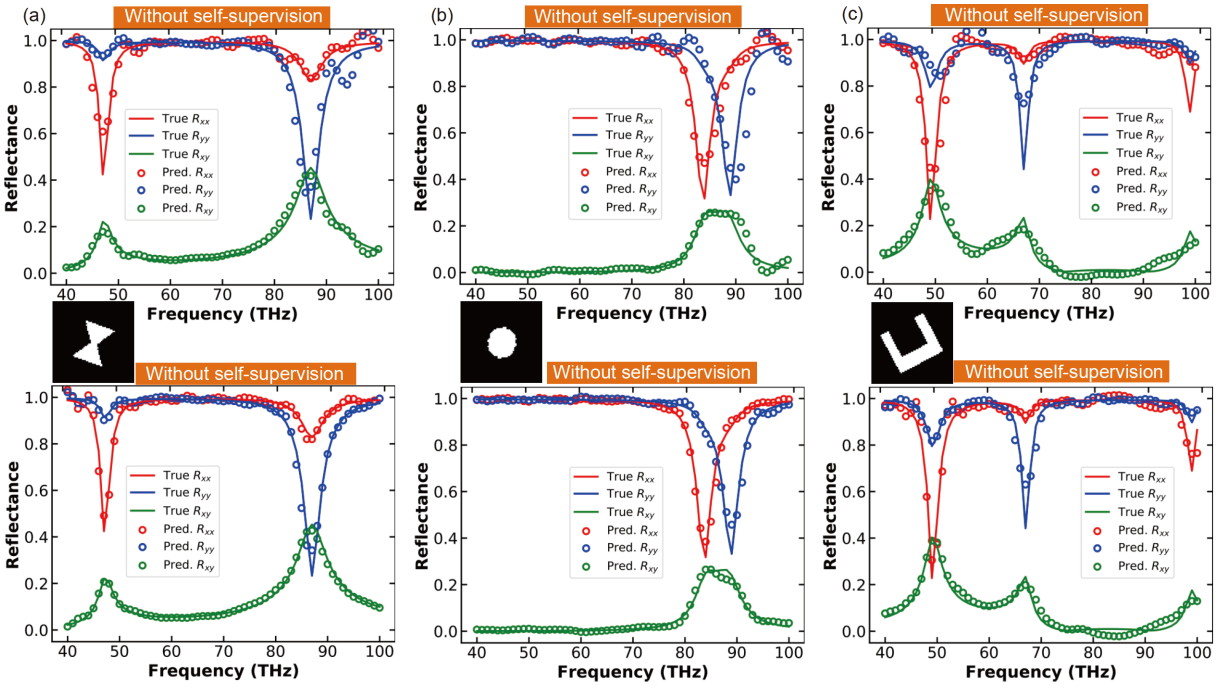

Figure 3 (Color online) Forward prediction of three samples, bowtie (a), ellipse (b) and split ring (c) from the test dataset. Top panels show the prediction without self-supervised learning (model U0) and bottom panels show the prediction with self-supervised learning (model U_dynamic).

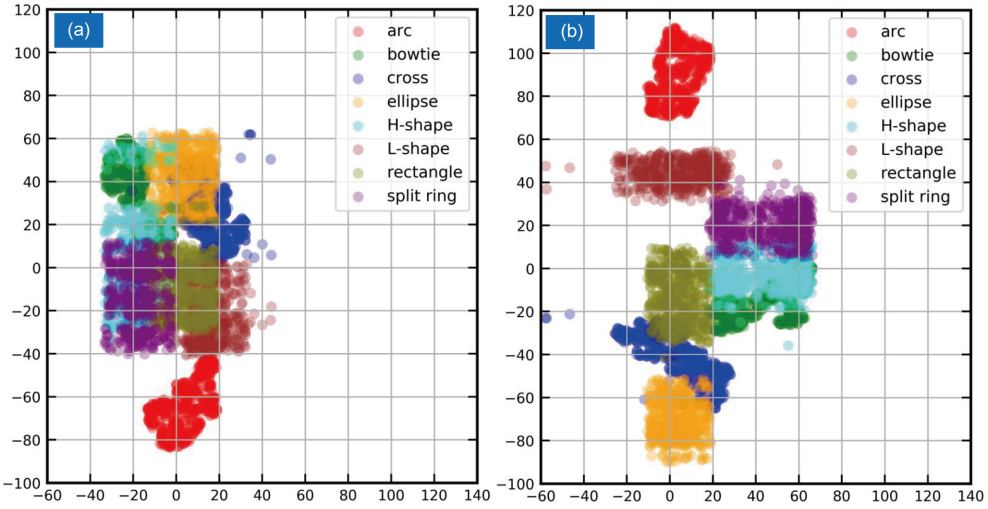

Figure 4 (Color online) Visualization of the latent space by reducing the dimension from 20 to 2 using t-SNE. The distribution of the nanophotonic structures from test dataset encoded by model U0 (a) and model U_dynamic (b), respectively.

Specifically, according to the training configuration, we first sample a 20-dimensional latent variable from a standard Gaussian prior distribution, $P(z)\sim N(0, I)$, and then feed the latent variable together with the required reflection spectra to the decoder of the model.

Since there is a discrepancy between the true latent distribution and the standard Gaussian prior, as described by the KL-divergence in eq. (1), we use the encoder to filter the retrieved patterns to guarantee the inverse design accuracy. Specifically, a mean-squared-error threshold of 10^{-3} is utilized to check the required spectra and the predicted spectra by feeding the generated design patterns back to the encoder, where the retrieved designs with error lower than this threshold are kept. We provide five retrieval results in Figure 5(b)-(f), with the required reflection spectra from a H-shape structure in the test dataset presented in Figure 5(a). From the simulated spectra in Figure 5(b)-(f), we can clearly see that the retrieved design patterns, albeit in very different geometries as shown in the insets, reproduce the corresponding input reflection spectra with high fidelity. These results unambiguously prove that the proposed U_dynamic model with self-learning can effectively link the nanophotonic structures and their corresponding optical responses through the probabilistic representation in the latent space.

4 Conclusions

To conclude, we propose a self-supervised learning mechanism to improve the performance of deep learning

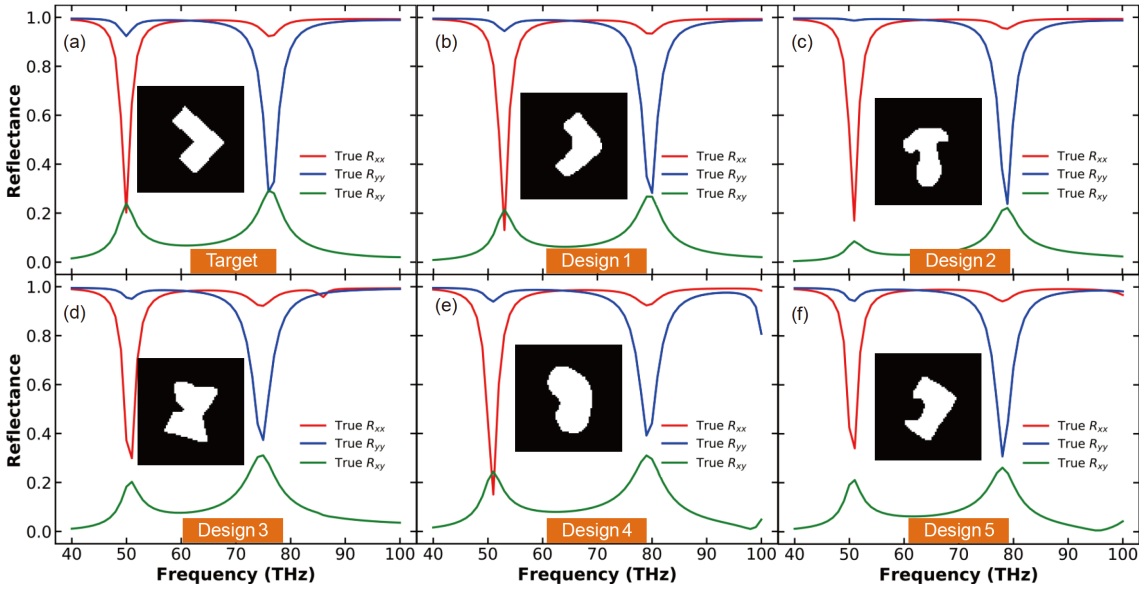

Figure 5 (Color online) Inverse design by the U_dynamic model. (a) Required spectra and the ground-truth design. (b)-(f) Retrieved designs and their corresponding reflection spectra.

models for nanophotonic structure designs. The deep learning model is composed of an encoder that compresses design patterns into a latent space, from which the patterns are reconstructed by another decoder together with corresponding reflection spectra. The self-supervised learning strategy allows generating random unlabeled data online during training, and the missing reflection spectra are replaced by the encoder output. In this way, the model performance is significantly improved in terms of total loss, prediction accuracy and latent space structure. The demonstrated deep learning model would serve as a very useful tool to design, characterize and optimize nanophotonic structures, which will find practical applications in building nanophotonic devices and systems for imaging, sensing and information processing. Moreover, the proposed self-learning strategy is flexible and widely applicable, which can be adapted to construct other data-efficient deep learning models for other research fields like physics and materials science, especially when it is difficult or expensive to collect labeled data.

This work was supported by the National Science Foundation (Grant No. ECCS-1916839).

1 N. Yu, P. Genevet, M. A. Kats, F. Aieta, J. P. Tetienne, F. Capasso, and Z. Gaburro, Science **334**, 333 (2011).
2 C. M. Watts, X. Liu, and W. J. Padilla, Adv. Mater. **24**, OP98 (2012).
3 X. Zhang, and Z. Liu, Nat. Mater. **7**, 435 (2008).
4 J. A. Schuller, E. S. Barnard, W. Cai, Y. C. Jun, J. S. White, and M. L. Brongersma, Nat. Mater. **9**, 193 (2010).
5 W. X. Tang, Z. L. Mei, and T. J. Cui, Sci. China-Phys. Mech. Astron. **58**, 127001 (2015).
6 X. Luo, ACS Photon. **5**, 4724 (2018).
7 S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, Nat. Photon. **12**, 659 (2018).
8 Y. Lecun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
9 B. Sanchez-Lengeling, and A. Aspuru-Guzik, Science **361**, 360 (2018).
10 G. B. Goh, N. O. Hodas, and A. Vishnu, J. Comput. Chem. **38**, 1291 (2017).
11 T. Zahavy, A. Dikopoltsev, D. Moss, G. I. Haham, O. Cohen, S. Mannor, and M. Segev, Optica **5**, 666 (2018).
12 P. Baldi, P. Sadowski, and D. Whiteson, Nat. Commun. **5**, 4308 (2014).
13 J. Carrasquilla, and R. G. Melko, Nat. Phys. **13**, 431 (2017).
14 L. Pilozzi, F. A. Farrelly, G. Marcucci, and C. Conti, Commun. Phys. **1**, 1 (2018).
15 D. Melati, Y. Grinberg, M. Kamandar Dezfouli, S. Janz, P. Cheben, J. H. Schmid, A. Sánchez-Postigo, and D. X. Xu, Nat. Commun. **10**, 1 (2019).
16 M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, Sci. Rep. **9**, 1 (2019).
17 O. Hemmatyar, S. Abdollahramezani, Y. Kiarashinejad, M. Zandehshahvar, and A. Adibi, Nanoscale **11**, 21266 (2019).
18 I. Sajedian, T. Badloe, and J. Rho, Opt. Express **27**, 5874 (2019).
19 J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, ACS Nano **13**, 8872 (2019).
20 Z. Liu, D. Zhu, K.-T. Lee, A. S. Kim, L. Raju, and W. Cai, Adv. Mater. **32**, 1904790 (2020).
21 W. Ma, F. Cheng, and Y. Liu, ACS Nano **12**, 6326 (2018).
22 Y. Chen, J. Zhu, Y. Xie, N. Feng, and Q. H. Liu, Nanoscale **11**, 9749 (2019).
23 W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, Adv. Mater. **31**, 1901111 (2019).
24 J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, Sci. Adv. **4**, eaar4206 (2018).
25 S. So, J. Mun, and J. Rho, ACS Appl. Mater. Interfaces **11**, 24264 (2019).
26 I. Sajedian, J. Kim, and J. Rho, Microsyst. Nanoeng. **5**, 27 (2019).
27 Z. Liu, D. Zhu, S. P. Rodrigues, K. T. Lee, and W. Cai, Nano Lett. **18**, 6570 (2018).
28 I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, Light Sci Appl **7**, 1 (2018).
29 T. Zhang, J. Wang, Q. Liu, J. Zhou, J. Dai, X. Han, Y. Zhou, and K. Xu, Photon. Res. **7**, 368 (2019).
30 T. Asano, and S. Noda, Opt. Express **26**, 32704 (2018).
31 D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, in *37th Advances in Neural Information Processing Systems*, Montreal, 9-13 December 2014. pp. 3581-3589.
32 M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019).
33 Y. Qu, L. Jing, Y. Shen, M. Qiu, and M. Soljačić, ACS Photon. **6**, 1168 (2019).
34 A. V. Kildishev, A. Boltasseva, and V. M. Shalaev, Science **339**, 1232009 (2013).
35 N. Yu, and F. Capasso, Nat. Mater. **13**, 139 (2014).
36 F. Aieta, P. Genevet, M. A. Kats, N. Yu, R. Blanchard, Z. Gaburro, and F. Capasso, Nano Lett. **12**, 4932 (2012).
37 W. Ma, Z. Huang, X. Bai, P. Zhan, and Y. Liu, ACS Photonics **4**, 1770 (2017).
38 S. Wang, P. C. Wu, V. C. Su, Y. C. Lai, M. K. Chen, H. Y. Kuo, B. H. Chen, Y. H. Chen, T. T. Huang, J. H. Wang, R. M. Lin, C. H. Kuan, T. Li, Z. Wang, S. Zhu, and D. P. Tsai, Nat. Nanotech. **13**, 227 (2018).
39 W. T. Chen, A. Y. Zhu, V. Sanjeev, M. Khorasaninejad, Z. Shi, E. Lee, and F. Capasso, Nat. Nanotech. **13**, 220 (2018).
40 F. Aieta, P. Genevet, N. Yu, M. A. Kats, Z. Gaburro, and F. Capasso, Nano Lett. **12**, 1702 (2012).
41 A. Pors, and S. I. Bozhevolnyi, Opt. Express **21**, 2942 (2013).
42 C. Pfeiffer, N. K. Emani, A. M. Shaltout, A. Boltasseva, V. M. Shalaev, and A. Grbic, Nano Lett. **14**, 2491 (2014).
43 J. Neu, R. Beigang, and M. Rahm, Appl. Phys. Lett. **103**, 041109 (2013).
44 X. Ni, A. V. Kildishev, and V. M. Shalaev, Nat. Commun. **4**, 2807 (2013).
45 G. Zheng, H. Mühlenbernd, M. Kenney, G. Li, T. Zentgraf, and S. Zhang, Nat. Nanotech. **10**, 308 (2015).
46 S. Sun, Q. He, S. Xiao, Q. Xu, X. Li, and L. Zhou, Nat. Mater. **11**, 426 (2012).
47 Y. Liu, and X. Zhang, Appl. Phys. Lett. **103**, 141101 (2013).
48 A. A. High, R. C. Devlin, A. Dibos, M. Polking, D. S. Wild, J. Perczel, N. P. de Leon, M. D. Lukin, and H. Park, Nature **522**, 192 (2015).
49 Z. Wang, K. Yao, M. Chen, H. Chen, and Y. Liu, Phys. Rev. Lett. **117**, 157401 (2016).
50 Z. Su, F. Cheng, L. Li, and Y. Liu, ACS Photon. **6**, 1947 (2019).
51 Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, in *30th Advances in Neural Information Processing Systems*, Barcelona, 2016. pp. 2352-2360.
52 D. P. Kingma and M. Welling, in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, Banff, 2014. pp. 14-16.
53 A. Kolesnikov, X. Zhai, and L. Beyer, in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 15-20 June 2019. pp. 1920-1929.