Transmittance Prediction and Inverse Design of Microring Resonator Channel Dropping Filters With Deep Learning

Guoping Chen and Chun Jiang

***Abstract*—We propose a method to use deep learning to achieve transmittance prediction and inverse design of microring resonator channel dropping filters. We transform the transmittance prediction and inverse design into model training problems, which learn and approximate the intrinsic interactions from the geometric parameter space to transmittance space and the transmittance space to geometric parameter space. The test loss and mean square error for the transmittance prediction case are 3.94888×10^{-2} and 4.68901×10^{-3}, respectively; the test loss and mean square error for the inverse design case are 7.27015×10^{-3} and 4.0029×10^{-4}, respectively. The numerical results suggest that the models developed by deep learning can make an efficient prediction of the transmittance and achieve excellent performance of the inverse design for microring resonator channel dropping filters, validating the effectiveness and feasibility of the approach we propose. With generalization ability within the given design space, the well-trained models can produce fast and accurate results without the need for time-consuming numerical calculations or case-by-case design.**

***Index Terms*—Deep learning, inverse design, microring resonator channel dropping filters, transmittance prediction.**

I. Introduction

MICRORING resonator devices are essential to wavelength-division multiplexing (WDM) optical interconnect architectures, not only in transmitters and receivers but also in switching nodes [1]–[5]. They perform the multiplexing and demultiplexing of multiple wavelength channels, as well as gain equalization and dispersion compensation for different channels. Microring resonator channel dropping filters (MRR-CDFs) have been investigated with significant effort both in theory and experiment. [6]–[13] When obtaining the transmittance at the drop end, designers often rely heavily on time-consuming numerical simulations. This is meant to solve the following problem: for a given geometry, what is the MRR-CDF transmittance. In the design process of MRR-CDFs, on the other hand, it is required to get the theoretical geometric parameters for the target transmittance as a guideline before manufacturing. Inverse design methods can, in principle, explore the full design space of devices [14]–[18]. The forward calculations are well understood with analysis based on the transfer matrix [19], while solving just one instance of an inverse design problem may often involve a complicated research subject. It is very tedious to obtain the geometric parameters of an arbitrary target spectrum shape because it is a reverse mapping process from the spectrum space with a higher dimension to the geometric parameter space with a lower dimension. Although trial-based and intuition-based approaches together with simplified analytic assumptions may be effective for simple structures with a small number of free parameters, such approaches become impractical with the increase of free parameters. Even for the most common scenario involving high-order MRRs for Butterworth or Chebyshev filters, we can first get the appropriate coupling strength according to the order of the filter in terms of the prototype filters and then acquire the geometric parameters corresponding to the coupling strength configuration based on numerical simulation [20], [21]. However, there exist deviations of the resonance of the microrings from the center frequency, which arises from the coupling-induced frequency shift (CIFS) issues [22]–[29]. When synthesizing a prototype filter, the ring-bus and ring-ring couplings differ significantly, leading to different CIFS between individual resonators, which bring about different effective resonant frequencies between them within the same filter. Consequently, the resonant frequency mismatches between resonators significantly distort the filter spectra. In general, the solution is to make the effective resonant frequencies of each resonator, considering the CIFS, the same as the target frequency of the filter as a whole [23], [29]–[31], which requires tedious and nontrivial attempts and parameter adjustment via much more numerical simulations. Hence, even with regards to Butterworth or Chebyshev filters, the design process is quite cumbersome and laborious, let alone that of general filters without a prototype filter for synthesizing to achieve any target spectrum shape.

As a result, a straightforward intelligence approach in terms of reliable transmittance prediction and inverse design of MRR-CDFs makes sense. Some recent work has begun to explore the application of machine learning in photonics, which has further paved the way for the design of photonic devices and photonic

Manuscript received September 30, 2021; revised February 26, 2022; accepted March 4, 2022. Date of publication March 8, 2022; date of current version March 24, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61975119. *(Corresponding author: Chun Jiang.)*

The authors are with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: guopingchen2017@sjtu.edu.cn; cjiang@sjtu.edu.cn).

Digital Object Identifier 10.1109/JPHOT.2022.3157776

systems [32], [33]. In this paper, we employ deep learning (DL) to achieve the transmittance prediction and inverse design of MRR-CDFs. DL is one of the major tools used in machine learning [34]–[42]. It has greatly enhanced the performance of many machine-learning tasks, including speech and image recognition [36], [43]–[45]. Besides, the availability of modern computing resources in recent years has already enabled the training of much more powerful and complex learning systems. In this paper, we make use of DL to explore the relationship between the topology and the transmittance and approximate the geometric parameters based on the target transmission spectra of MRR-CDFs. In the transmittance prediction case, we feed the geometric parameters to the trained model to predict the transmittance. While in the inverse design case, given the spectra we desire, we have the model developed for inverse design predict what geometries would produce the target spectra, and then apply the predicted geometric parameters to check the spectra with the finite-difference time-domain (FDTD) numerical results to evaluate the performance of the inverse design. With generalization ability within a given design space, the well-trained model can produce fast and accurate results without the need for time-consuming numerical calculations or case-by-case design. In contrast, traditional filter design methods based on the specific prototype filters or parameter scanning need repeated numerical simulation and multi-parameter adjustment for a single design case. In this sense, the approach we propose offers both fast speeds compared to numerical simulations for the transmittance prediction and high reliability for the inverse design process.

The paper is organized as follows. Section II gives the theoretical analysis and model building process for transmittance prediction and inverse design with DL. The results are presented in section III, which show an excellent agreement with the theoretical analysis. Finally, section IV draws conclusions and presents future research lines.

II. Theoretical Analysis And Model Building With Deep Learning

In this section, we give the theoretical analysis and model building process for transmittance prediction and inverse design of MRR-CDFs based on DL. We consider the polarization-independent microring resonator filter structure [46]. MRR-CDFs are composed of two straight waveguides and several microrings. The structural diagrams with one and four microrings are depicted in Fig. 1(a) and (b), respectively. As shown, two beams of light can launch into the input and add ports independently.

When obtaining the transmission spectra at the drop end numerically, we have to manually launch time-consuming numerical simulations for each configuration with different structural parameters every time. On the other hand, when manufacturing a channel dropping filter, we hope to derive the relationship from the transmittance to geometric parameters inversely, thus obtaining the theoretical geometric parameters as a guideline based on any desired transmittance. The problem is rather involved, as the number of rings increases, especially considering the CIFS issues. We appeal to DL to achieve the transmittance

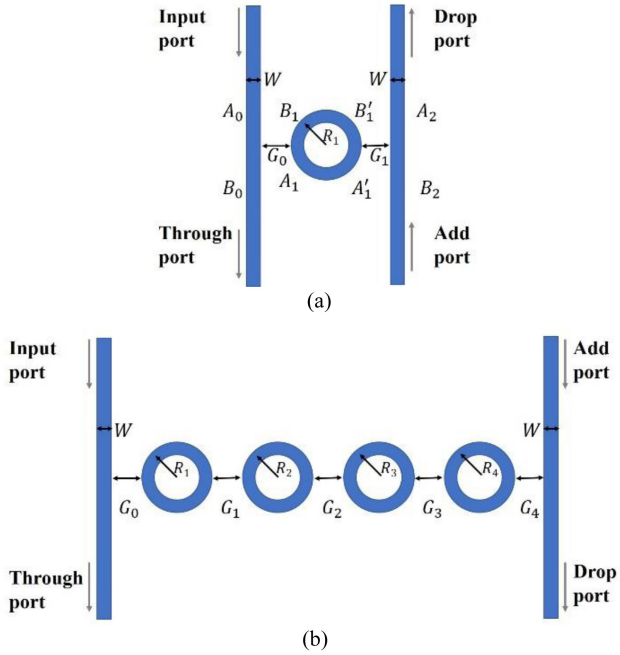

Fig. 1. Microring resonator channel dropping filters with: (a) one microring, (b) four microrings. We denote the widths of the buses and microrings as W, the radii of the microrings as R_1, R_2, R_3 and R_4, and the gaps as G_0, G_1, G_2, G_3 and G_4.

prediction and inverse design. The main goal is to build models for transmittance prediction and inverse design. We implement DL in Keras, which is one of the most popular machine learning libraries currently available based on python. The major steps involved in transmittance prediction and inverse design with DL include: selecting features (network input) and collecting response variables (network output), training the network, evaluating the generalization performance of the model, and making predictions via the model.

The data features that we employ to train our models have a huge influence on the performance we can achieve. Irrelevant or partially relevant features can negatively impact the model performance. Thus when solving a specific problem with DL, it is one of the critical steps to analyze all the input variables that affect the network output. For the integrity of the solution we propose, we first analyze the factors that affect the transmittance of MRR-CDFs in detail before preparing datasets used in DL in Appendix A. Based on the analysis in Appendix A, given the material and cross-section of the waveguide, the factors affecting the transmittance are the radii of the rings and the gaps between each component. For a given material and cross-section, the radii of the rings affect the free spectral range (FSR), which is an essential indicator in practice. FSR affects the maximum bandwidth that can be achieved by the WDM system [10], [47], [48]. Therefore, we first determine the initial radii that match the FSR requirement. In order to automatically compensate for the spectrum distortion caused by the CIFS in the filter design process, we regard the fine-tuning margins of the radii as one of the factors that affect the transmittance. As a result, we take the coupling gaps and the fine-tuning margins of the radii as the geometric parameters that affect the transmittance. After analyzing

the constituent elements of geometric parameters, we randomly sample the geometric parameters within their boundaries and then collect the corresponding transmittance. Then we bind the collected geometric parameters and the transmittance together as the dataset for model training and selection.

The efficient boundaries of the adjustment margins of the radii are given in (18) in Appendix B. The lower bounds of all the gaps are set to zero. When determining the upper bounds of the gaps, we first obtain the cross-coupling strength with varied gaps between individual structures with numerical simulations and then set the upper boundaries of the gaps to the value as the coupling strength decays almost to zero. Since inverse design is the reverse problem of transmittance prediction, we use the sampled transmittance as the features and geometric parameters as the response variables in the inverse design case.

We employ feed-forward and fully connected models to implement our scheme. The network architecture for the transmittance prediction case is depicted in Fig. 2(a), which is composed of input and output layers, as well as hidden layers consisting of nodes that bring connection between the input and the output. The architecture has as its features the geometric parameters and as its response variables the sampled transmittance. The geometric parameters G are collections of the offset dR_k $(k = 1, \ldots, K)$ from an initial radius R and the gaps G_j $(j = 0, \ldots, J)$, that is $[dR_1, \ldots, dR_K, G_0, \ldots, G_J]$. It is assumed that the center point of the whole filter is the coordinate origin, and the direction along the arrangement of the rings is the direction of the x-axis. The structure of the general filter is symmetrical about the y-axis. Since this symmetrical structure can effectively reduce the dimension of geometric parameters, our filter structure retains this symmetry about the y-axis direction. As a result, when the number of the rings is even, $K = N/2$, $J = N/2$; when the number of the rings is odd, $K = \text{ceil}(N/2)$, $J = \text{floor}(N/2)$, (ceil refers to rounding up to an integer, floor means rounding down to an integer, and N is the number of rings). The sampled transmittance T has M elements ($[t_1, \ldots, t_M]$) shown in Fig. 2(b), where M is the sampling number in the transmission spectra. In contrast, the network architecture for the inverse design case is shown in Fig. 3, with the sampled transmittance as its features and the geometric parameters its response variables.

After feature selection and response variables collection, we come to the training of the network. In the transmittance prediction case, the network is trained to approximate the function mapping from the geometric parameters to the transmission spectra. While in the inverse design case, the network is trained to approximate the function that takes the transmission spectra into exactly one spot in the geometric parameter space inversely. Before model training, we first shuffle all the data in the dataset and separate the dataset into three parts: a training dataset, a validation dataset, and a test dataset. The training dataset is used to fit different models, and the performance on the validation dataset is assessed for the model selection. The advantage of setting aside a test dataset that the model has not seen before during the training and model selection process is that we can acquire a less biased estimate of its generalization ability. To find the optimal weights of the model, we define the cost function J

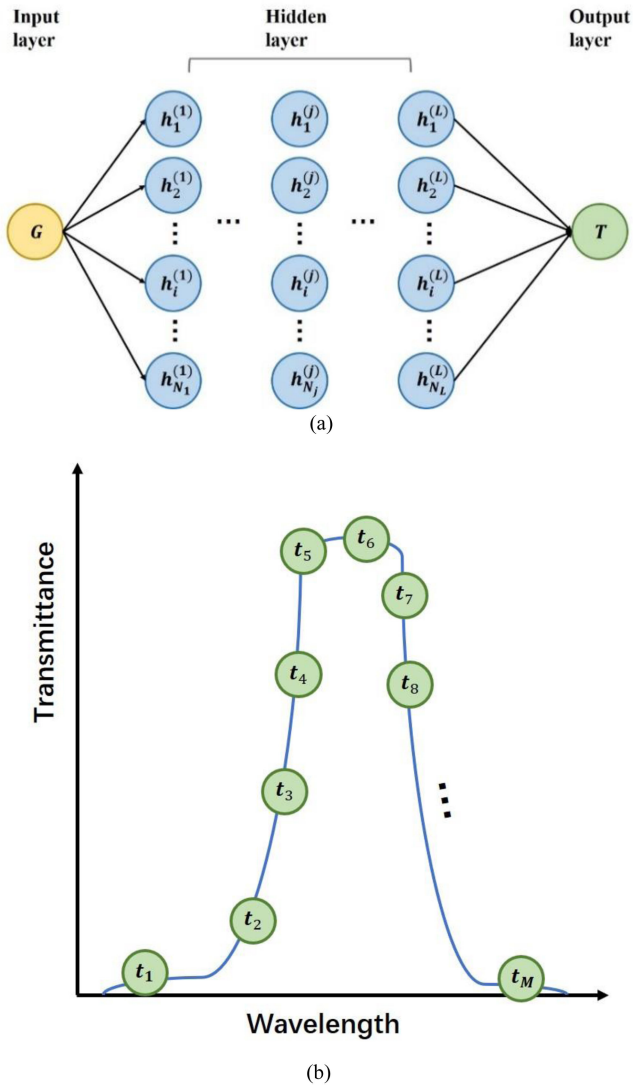

Fig. 2. (a) The network architecture for the transmittance prediction case. The number of hidden layers and the number of nodes for each hidden layer in the actual network depend on the result of model training and evaluation. (b) The sampling of the transmission spectrum as the output of the network for the transmittance prediction case and as the input of the network for the inverse design case.

to learn the weights as the mean absolute error (MAE) between the predicted and the true outcome as follows:

$$J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}|y_i - y_i'| \tag{1}$$

where $y_i{}'$ is the ith predicted result, y_i is the true value corresponding to the ith input example, and m is the sample number in the training dataset. L2 regularization is also applied to tune the complexity of the model to tackle overfitting by adding a simple regularization term to the cost function in (1). Then the regularized cost function reads as:

$$J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}|y_i - y_i'| + \frac{\lambda}{2m}\|\mathbf{w}\|^2 \tag{2}$$

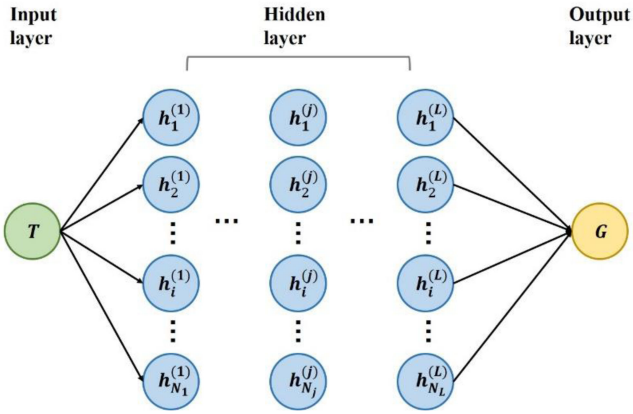

Fig. 3. Network architecture for the inverse design case. The number of hidden layers and the number of nodes for each hidden layer in the actual network depend on the outcome of model training and evaluation.

Where λ is the regularization parameter, **w** is the network weight vector. The regularization parameter λ is used to regulate the goodness-of-fit for the training data. By increasing the regularization parameter λ, we shrink the weights towards zero to reduce the dependence on the training data of the model.

After the definition of the cost function, we apply the adaptive moment estimation (Adam) optimizer to minimize the cost function. We also specify the mean square error (MSE) as a metric to judge the performance of our model. To estimate the generalization performance of the machine learning models, we perform cross-validation with the validation dataset to repeatedly evaluate the performance of the models, diagnose the problems of the machine learning algorithm, and fine-tune the machine learning models with different hyperparameter values. To further reduce the possible overfitting, we adopt the early stopping method.

Once we are satisfied with the loss and MSE metric on both the training and validation dataset, we apply the trained models to estimate the generalization performance on the test dataset. If the models perform well on the test dataset, we use the trained models to predict the transmittance given the geometric parameters or suggest a device geometry for a target transmittance. Otherwise, we reiterate the training and validation process of the models by fine-tuning the hyperparameter values.

III. Results And Discussion

In this part, we give the results and discussion of transmittance prediction and inverse design by using DL under the configuration of four-microring filters (The structure is depicted in Fig. 1(b)). The 2D simulations are much quicker than 3D simulations, and they illustrate the essential features. Thus, we employ 2D simulations to demonstrate the scheme we propose. We use Meep [1] to implement the finite-difference time-domain (FDTD) parallel numerical simulations [49]. Suppose the filter has a FSR of 21 nm and a resonant wavelength of near 1550 nm. The straight waveguides and microrings are composed of silicon

[1][Online]. Available: http://ab-initio.mit.edu/meep

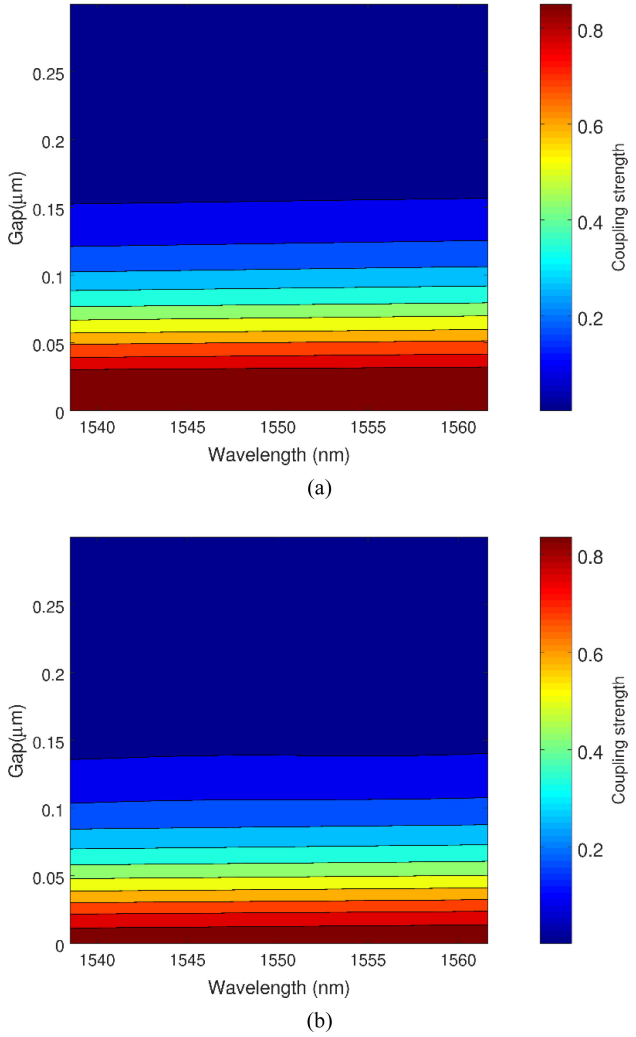

Fig. 4. (a) Contour plots of cross-coupling strength between the straight waveguide and the ring curve (R = 5 μm) in the x-y plane with the electric field along the z-axis. (b) Contour plots of cross-coupling strength between adjacent ring curves (R = 5 μm) in the x-y plane with the electric field along the z-axis.

with a width of 0.3 μm. Only the fundamental transverse magnetic (TM) mode with electric field oriented along the z-direction is considered. If the input signal is TE mode, we can convert the TE mode into TM mode before feeding it into the straight waveguide [46]. The computational domain is truncated with perfectly matched layers (PMLs). These performance indicators result in an initial radius R of about 5 μm.

We first determine the bounds of geometric parameters based on these performance indicators. Under this configuration, the value of dR_{max} is 0.083 μm from (20) in Appendix B. To have a little slack for adjustment, we set the lower and upper bounds of dR_i to -0.1 μm and 0.1 μm, respectively. Referring to [50], we employ the structures in Fig. 5 and Fig. 6 to figure out the fields between the straight waveguide and the ring curve and between adjacent ring curves. The mode light source is excited only in the TM mode at point S, and the self-coupling and cross-coupling signals are along with the arrow directions in the figures. The

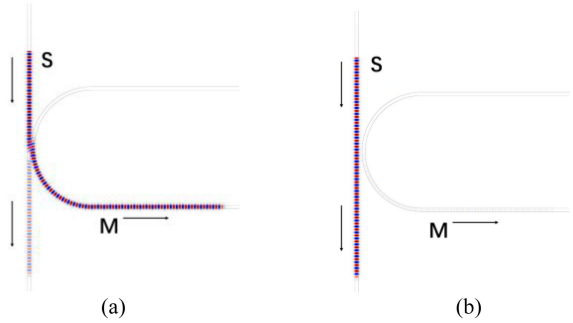

Fig. 5. Field distribution between the straight waveguide and the ring curve: (a) With the gap set to 0 μm. (b) With the gap set to 0.3 μm.

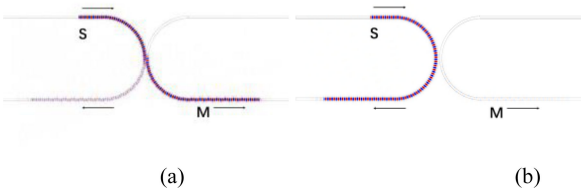

Fig. 6. Field distribution between two adjacent ring curves: (a) With the gap set to 0 μm. (b) With the gap set to 0.3 μm.

monitoring point is placed at point *M*. The fields are absorbed when approaching the PMLs. Under the configurations, we can obtain the changes of the cross-coupling strength (the square of the magnitude of the cross-coupling coefficient in Appendix A) with different gaps and wavelengths by dividing the flux at the monitoring point by the flux of the mode light source before coupling. In this section, since our working frequency band is around 1550nm and the FSR is 21nm, we consider the cross-coupling strength within the band spanning two free spectral ranges centered at 1550nm. On the contrary, if our working band is in other ranges, the boundary analysis of coupling gaps needs to be carried out in the corresponding ranges.

The contour plots of the cross-coupling strength between the straight waveguide and the ring curve, as well as the cross-coupling strength between the adjacent ring curves as a function of the gaps and wavelengths, are depicted in Fig. 4(a) and (b), respectively. In the contour plots, the blue colors correspond to lower values of strength and the red colors to higher values. From these figures, we can clearly see that the cross-coupling strength is almost zero when the gaps between the straight waveguide and the ring curve and between the adjacent ring curves exceed 0.3 μm. The field distributions between the straight waveguide and the ring curve with the gaps between them set to 0 μm and 0.3 μm are shown in Fig. 5(a) and (b), respectively. Besides, the field distributions between the adjacent ring curves with the gaps between them set to 0 μm and 0.3 μm are also given, respectively, in Fig. 6(a) and (b). It is also evident from Fig. 5(b) and Fig. 6(b) that when the gaps reach 0.3 μm, the fields coupled from the straight waveguide to the ring curve and from the left ring curve to the right one nearly decay to zero. We thus set the lower and upper boundaries of both the gaps between the straight waveguide and the ring and between the adjacent rings to 0 μm and 0.3 μm, respectively.

Within the bounds of the geometric parameters obtained above, we randomly generate 25000 sets of geometric parameters ($[dR_1, dR_2, G_0, G_1, G_2]$) in micron and then collect the corresponding spectra ($[t_1, \ldots, t_M]$) uniformly sampled between 1527 nm and 1574 nm through the FDTD numerical method. We employ 8 servers for parallel numerical simulations at the same time. Each server contains 128 vcpu, and the processor type is *AMD EPYC ROME 7H12*. The time to train the model for transmittance prediction is about 10 minutes, while the time to train the model for inverse design is about 5 minutes with one server. It should be pointed out that there exist real-world limitations of fabrication constraints, such as minimum feature size and gap size unique to each foundry. In the actual fabrication process, these constraints should be considered. However, we currently focus on theoretical and numerical solutions for transmission spectrum prediction, as well as for inverse design to automatically solve the CIFS problem.

In the transmittance prediction case, all input samples (the geometric parameters) can be written as a matrix $\mathbf{In}_{25000\times5}$, and all output samples (the sampled transmittance) can be written as a matrix $\mathbf{Out}_{25000\times100}$. Firstly, we shuffle all the data in the dataset $(\mathbf{In}_{25000\times5}, \mathbf{Out}_{25000\times100})$ by rows and split the dataset by rows into three parts: a training dataset, a validation dataset, and a test dataset (81%, 9%, and 10%, respectively). Next, we train the network using the samples in the training dataset with a batch size of 100 and an Adam optimizer. In the Adam optimizer, we set the learning rate to 0.001, the exponential decay rate for the first-order moment estimate to 0.9, the exponential decay rate for the second-order moment estimate to 0.999, and the small constant for numerical stability to 10^{-8}. To initialize the network weights, we set all weights to samples from a uniform distribution within [-limit, limit], where $\text{limit} = \text{sqrt}(6/(\text{fan\_in} + \text{fan\_out}))$ (fan_in is the number of input units in the weight tensor and fan_out is the number of output units) and set all biases to zeros. To avoid overfitting, we set the L2 regularization parameter $\lambda = 0.0000001$ in the third hidden layer during the training. The training converges after 1333 epochs.

In the inverse design case, all input samples (the sampled transmittance) can be expressed as a matrix $\mathbf{In}_{25000\times100}$, and all output samples (the geometric parameters) can be written as a matrix $\mathbf{Out}_{25000\times5}$. We shuffle all the data in the dataset $(\mathbf{In}_{25000\times100}, \mathbf{Out}_{25000\times5})$ by rows and divide the dataset by rows into three categories: a training dataset, a validation dataset, and a test dataset (81%, 9%, and 10%, respectively). Next, we train the network using the samples in the training dataset with a batch size of 500 and an Adam optimizer. In the Adam optimizer, we set the learning rate to 0.001, the exponential decay rate for the first-order moment estimate to 0.9, the exponential decay rate for the second-order moment estimate to 0.999, and the small constant for numerical stability to 10^{-8}. We initialize all weights to values from a uniform distribution within [-limit, limit], and all biases to zeros. To avoid overfitting, we set the L2 regularization parameter $\lambda = 0.00000008$ in the second hidden layer and $\lambda = 0.000000001$ in the third hidden layer during the

TABLE I
PARAMETERS OF THE NETWORK ARCHITECTURES

Parameters	Transmittance prediction	Inverse design
Number of hidden layers	3	3
Node number of hidden layer 1	2048	2048
Node number of hidden layer 2	2048	2048
Node number of hidden layer 3	2048	1024
The activation function of hidden layer 1	Relu	Relu
The activation function of hidden layer 2	Relu	Relu
The activation function of hidden layer 3	Relu	Relu
The activation function of the output layer	Relu	Tanh
The dimension of a single input sample	[1, 5]	[1, 100]
The dimension of a single output sample	[1, 100]	[1, 5]
Total example number in the dataset	25,000	25,000
Total parameters	8,609,892	6,506,501

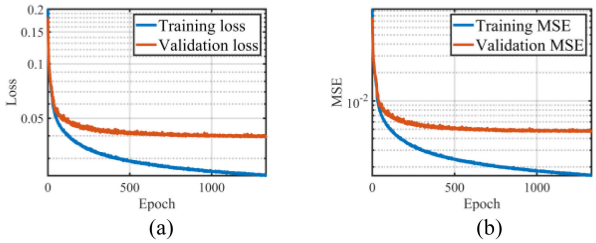

Fig. 7. (a) Loss curves for the transmittance prediction case. (b) MSE curves for the transmittance prediction case.

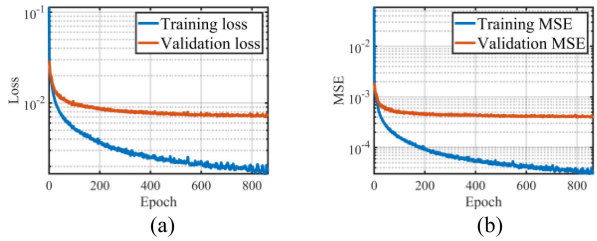

Fig. 8. (a) Loss curves for the inverse design case. (b) MSE curves for the inverse design case.

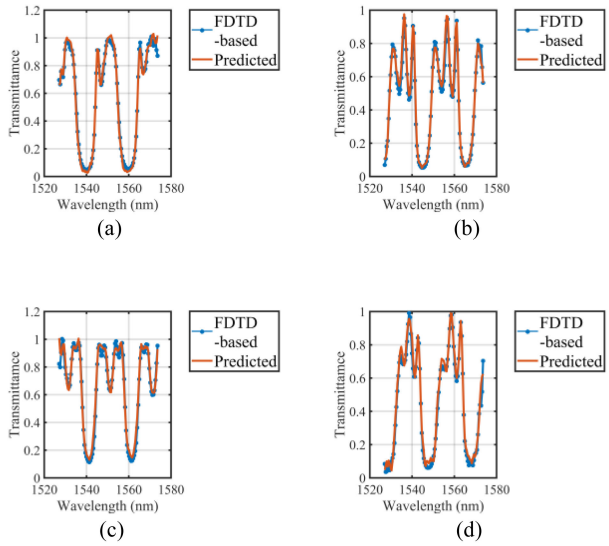

Fig. 9. Comparison of the FDTD-based transmittance in the test dataset and the corresponding predicted transmittance in the transmittance prediction case for (a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4.

training. The training achieves convergence after 864 epochs. We can see that the inverse-design model takes fewer epochs to train. This is because the input has more parameters and the output has fewer parameters.

Finally, we achieve two networks for the two cases. The parameters of the network architectures are illustrated in Table I in Appendix C. The driving criterion in optimizing the network is to reduce the loss on both the training and validation dataset and to yield a gap as narrow as possible between the training and validation loss curves after training. We also specify the MSE as a metric to quantify the performance of our model. We plot the loss, including the training and validation loss, as well as the MSEs on both the training and validation dataset. The loss and MSEs with logarithmic scaling (Y-axis) over 1333 training epochs for the transmittance prediction and over 864 training epochs for the inverse design are given in Fig. 7(a)–(b) and Fig. 8(a)–(b), respectively. As we can see, both the training and validation loss in Fig. 7(a) and Fig. 8(a) goes down pretty quickly to a stable state, and the final gaps between the training and validation loss are quite narrow. Simultaneously, the training and validation MSEs in both Fig. 7(b) and Fig. 8(b) also decrease with the training epochs, and the final separations between the training and validation MSEs are very small. The results in Fig. 7(a)–(b) and Fig. 8(a)–(b) reveal that the models we train in both cases perform well in both the training and validation datasets, and are likely to generalize well to new data, for example, the test dataset, which we use for the final model evaluation.

After selecting the two models fitted on the training dataset, we evaluate the generalization performance of the models by calculating the loss and the MSE metric on the test dataset. The test loss and test MSE for the transmittance prediction case are 3.94888×10^{-2} and 4.68901×10^{-3}, respectively; the test loss and test MSE for the inverse design case are 7.27015×10^{-3} and 4.0029×10^{-4}, respectively. Hence, we can conclude that our models not only yield good performance in the training and validation dataset but also generalize well to new data. Next, in the transmittance prediction case, we randomly select four samples of FDTD-based transmittance in the test dataset shown in Fig. 9(a)–(d) with the legend "FDTD-based" and the corresponding predicted transmittance shown in Fig. 9(a)–(d) with the legend "Predicted" for comparison. Also, in the inverse design case, we randomly select four sets of target transmittance in the test dataset shown in Fig. 10(a)–(d) with the legend "Target", then use the four sets of predicted geometric parameters that map the four sets of target transmittance in the test dataset to perform FDTD simulations to check the transmittance for the inverse design as shown in Fig. 10(a)–(d) with the legend "Inverse-designed". It is evident from Fig. 9 and Fig. 10 that

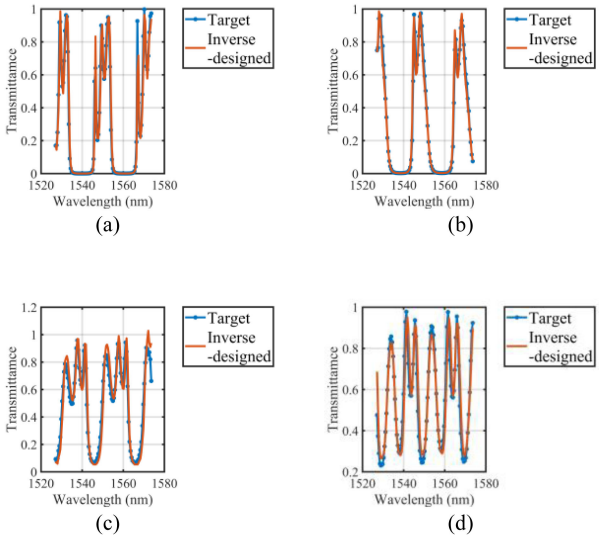

Fig. 10. Comparison of the target transmittance in the test dataset and the corresponding inverse designed transmittance in the inverse design case for (a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4.

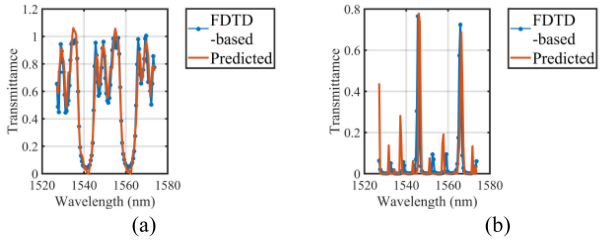

Fig. 11. Comparision of the FDTD-based and the predicted transmittance: (a) With the randomly generated geometric parameter [0.080571, -0.092486, 0.043606, 0.012873, 0.014656] with the unit of micron. (b) With the randomly generated geometric parameter [0.052099, 0.011391, 0.099598, 0.075295, 0.018692] with the unit of micron.

our models achieve good performance with randomly selected samples. Finally, since we are satisfied with the generalization performance of the networks for both cases in the test dataset, we next let the two models we train loose on the real world to make predictions on new data.

In the transmittance prediction case, we randomly regenerate two sets of geometric parameters ([0.080571, -0.092486, 0.043606, 0.012873, 0.014656] and [0.052099, 0.011391, 0.099598, 0.075295, 0.018692] with the unit of micron) within the geometric parameter boundaries and let the model predict the transmittance. The transmittance from FDTD numerical results with the two newly generated geometric parameters are illustrated in Fig. 11(a) and (b) with the legend "FDTD-based". Then we feed the two sets of geometric parameters to the trained model to predict the transmittance, which results in the two curves, as shown in Fig. 11(a) and (b) with the legend "Predicted". By comparing the FDTD-based and predicted spectra in Fig. 11(a) and (b), we can conclude that the model we achieve can indeed make accurate and efficient predictions about the spectra given the geometric parameters.

Following the performance evaluation of the model for the transmittance prediction, we assess the performance of the

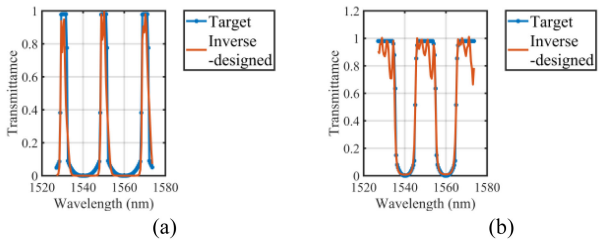

Fig. 12. Evaluation of inverse design: (a) With a 3 dB bandwidth of about 3.3 nm. (b) With a 3 dB bandwidth of about 10 nm.

model for the inverse design. In this case, we draw a relatively narrow spectrum with a 3 dB bandwidth of about 3.3 nm and a relatively wide spectrum with a 3 dB bandwidth of around 10 nm centered at 1550 nm, as illustrated in Fig. 12(a) and (b) with the legend "Target" as the target spectra. Then, we realize the inverse design of the MRR-CDF through the trained model. Given the spectra we desire, we have the model predict what geometries would produce the target spectra. We feed the target spectra in Fig. 12(a) and (b) with the legend "Target" as the input to the trained model and get two sets of output [0.0784117, 0.0691013, 0.0478859, 0.0970026, 0.141379] and [0.07115, 0.0685247, 0.0368173, 0.037674, 0.0380903] with the unit of micron as the predicted geometric parameters. To verify the validity of the inverse design, we apply the predicted geometric parameters to generate spectra with FDTD simulations, as shown in Fig. 12(a) and (b) with the legend "Inverse-designed". The comparisons between the target and inverse-designed results show that the model we obtain can make a good prediction of the geometric configuration parameters, thus producing the spectra that resemble the target spectra as closely as possible.

It should be pointed out that the transmittance based on numerical simulations is occasionally slightly greater than 1 at some frequency points far from the center spectrum in the numerical simulation. The reason is that the flux is being computed at a frequency that is too far away from the center of the source bandwidth (Since the FSR is an important characteristic of filters in WDM systems, we simulate within the bandwidth spanning more than two FSRs to retain more spectral information, resulting in a wide frequency band in our simulation.). In such cases, the flux values at frequencies that are far away from the center of the bandwidth are too small and may be dominated by rounding errors. At the same time, the transmittance predicted based on the model may be occasionally slightly greater than 1 at the frequency point where the actual transmission spectrum is close to 1, which is caused by a small prediction error.

It is demonstrated from all the results above that when appropriately trained and evaluated, the models can approximate the spectra very well and gain a good effect for the inverse design of MRR-CDFs. The trained model actually holds a meaningful mapping from the geometric parameter space to the transmittance space or the transmittance space to the geometric parameter space. Therefore, DL is a practical solution to the problems of transmittance prediction and inverse design. If we want to achieve even better performance, we can further reduce the sampling interval of the transmission spectra, and increase

the number of samples in the collected dataset. Limited by time and other factors, we will further improve this aspect in future work.

Although it takes some time to collect data, due to the availability of modern computing resources, our data collection process is based on automated large-scale parallel simulations without human intervention. More importantly, once the data collection and network training are completed, the transmission spectrum prediction and the inverse design of the filters are once and for all (getting rid of the design process of the filter from scratch for the desired spectrum shape each time).

IV. Conclusion

To conclude, our work based on DL provides a complete solution for transmittance prediction and the inverse design of MRR-CDFs from scratch, including problem analysis, input-output feature extraction, boundary analysis of input parameters, data collection based on automated massively parallel simulations, model training, and model evaluation. The excellent agreement between the predicted and numerical results confirms the approach presented in this work. The process of data collection and model building is a once-and-done operation. Once we have completed the model selection, all we have to do is to load the model, and then feed the geometric parameters to the model for transmittance prediction or input the target transmittance to the model for the filter design. With the well-trained models, we can directly achieve the transmittance prediction and quickly acquire the geometric parameters closest to the shape of the target spectrum within the design space. Especially for the design of the filter, it avoids the repeated process of parameter traversal and parameter adjustment based on numerical simulations for every desired spectrum shape and automatically solves the problem of spectrum distortion caused by CIFS. By contrast, the traditional filter design methods demand repeated numerical simulations and multi-parameter adjustment for a single design case, which takes a lot of repetitive manual work. In this regard, the approach we propose offers a more efficient and effective alternative for transmittance prediction and filter design.

As an exploration into the design of photonic devices with DL, the scheme we propose opens new design possibilities for a large number of universal applications in components, devices, and systems based on the manipulation of networks.

Appendix A

When solving a specific problem with DL, it is one of the critical steps to collect all the input variables that affect the network output. Only by fully and accurately analyzing all the input factors that affect the output can we get the correct DL model. For the two problems of transmittance prediction and inverse design of microring resonator channel dropping filters, we first analyze the factors affecting transmittance in detail before collecting data sets used in DL. Meanwhile, in the process of deriving the transmittance, we give the physical source of CIFS, which is also an obstacle in the design of filters.

For the convenience of analysis, we get started on the one-microring case in Fig. 1(a). In Fig. 1(a), we lump the distributed coupling region into a point interaction with respect to a single reference plane. Suppose a single unidirectional mode is excited and the coupling is lossless, the coupling interaction reads as [50]:

$$\begin{bmatrix} B_0 \\ A_1 \end{bmatrix} = \mathbf{X_{01}} \begin{bmatrix} A_0 \\ B_1 \end{bmatrix} = e^{j\theta_0^\lambda} \begin{bmatrix} \kappa_{00}^\lambda & j\kappa_{10}^\lambda \\ j\kappa_{10}^\lambda & \kappa_{00}^\lambda \end{bmatrix} \begin{bmatrix} A_0 \\ B_1 \end{bmatrix} \tag{3}$$

where A_0 and B_1 are the complex mode amplitudes of the input ports for the first coupler of the microring; A_1 and B_0 are the complex mode amplitudes of the corresponding output ports; $\mathbf{X_{01}}$ is the scattering matrix of the coupler, which suggests the relationship between the inputs and outputs within the same coupler; κ_{00}^λ and κ_{10}^λ are the self-coupling and cross-coupling coefficients for the first gap G_0; and θ_0^λ is the coupling-induced phase for the first gap G_0. In order to figure out the relationship between the input and output of the filter taken as a whole in Fig. 1(a), (3) can be recast as:

$$\begin{bmatrix} A_0 \\ B_0 \end{bmatrix} = \mathbf{S_{01}} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} = \frac{1}{1j \times \kappa_{10}^\lambda} \begin{bmatrix} e^{-j\theta_0^\lambda} & -\kappa_{00}^\lambda \\ \kappa_{00}^\lambda & -e^{j\theta_0^\lambda} \end{bmatrix} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} \tag{4}$$

where $\mathbf{S_{01}}$ is the transfer matrix for the gap G_0 and can be decomposed as:

$$\begin{aligned} \mathbf{S_{01}} &= \frac{1}{1j \times \kappa_{10}^\lambda} \begin{bmatrix} e^{-j\theta_0^\lambda} & -\kappa_{00}^\lambda \\ \kappa_{00}^\lambda & -e^{j\theta_0^\lambda} \end{bmatrix} \\ &= \frac{1}{1j \times \kappa_{10}^\lambda} \begin{bmatrix} e^{\frac{-j\theta_0^\lambda}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^\lambda}{2}} \end{bmatrix} \begin{bmatrix} 1 & -\kappa_{00}^\lambda \\ \kappa_{00}^\lambda & -1 \end{bmatrix} \begin{bmatrix} e^{\frac{-j\theta_0^\lambda}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^\lambda}{2}} \end{bmatrix} \end{aligned} \tag{5}$$

The similar transfer matrix $\mathbf{S_{i(i+1)}}$ for the gap $G_i (i = 01, \ldots, 4)$ in Fig. 1(b) follows immediately with the substitute of κ_{00}^λ, κ_{10}^λ and θ_0^λ to κ_{ii}^λ, $\kappa_{(i+1)i}^\lambda$ and θ_i^λ, respectively:

$$\begin{aligned} \mathbf{S_{i(i+1)}} &= \frac{1}{1j \times \kappa_{(i+1)i}^\lambda} \begin{bmatrix} e^{-j\theta_i^\lambda} & -\kappa_{ii}^\lambda \\ \kappa_{ii}^\lambda & -e^{j\theta_i^\lambda} \end{bmatrix} \\ &= \frac{1}{1j \times \kappa_{(i+1)i}^\lambda} \begin{bmatrix} e^{\frac{-j\theta_i^\lambda}{2}} & 0 \\ 0 & e^{\frac{j\theta_i^\lambda}{2}} \end{bmatrix} \begin{bmatrix} 1 & -\kappa_{ii}^\lambda \\ \kappa_{ii}^\lambda & -1 \end{bmatrix} \\ &\quad \times \begin{bmatrix} e^{\frac{-j\theta_i^\lambda}{2}} & 0 \\ 0 & e^{\frac{j\theta_i^\lambda}{2}} \end{bmatrix} \end{aligned} \tag{6}$$

with κ_{ii}^λ and $\kappa_{(i+1)i}^\lambda$ being the self-coupling and cross-coupling coefficients for the gap G_i, and θ_i^λ the coupling-induced phase for the gap G_i. The amplitude transmission factor around the first ring in Fig. 1(a) is given by:

$$\alpha_1 = 10^{-2\pi R_1 \gamma_1 / 20} \tag{7}$$

where γ_1 is the loss factor. The relationship between the complex mode amplitudes (A_1, B_1) and (A_1', B_1') where the ring cavity is located at the first and second couplers in Fig. 1(a) can be

expressed as:

$$\begin{bmatrix} A_1 \\ B_1 \end{bmatrix} = \mathbf{P_1} \begin{bmatrix} A_1' \\ B_1' \end{bmatrix} = \begin{bmatrix} \alpha_1{}^{-1}\!/_2 e^{\frac{j\phi_1}{2}} & 0 \\ 0 & \alpha_1{}^{1}\!/_2 e^{\frac{-j\phi_1}{2}} \end{bmatrix} \begin{bmatrix} A_1' \\ B_1' \end{bmatrix} \tag{8}$$

where $\mathbf{P_1}$ stands for the transfer matrix of the first ring element and ϕ_1 is the phase shift of one turn in the first ring element, which is given by:

$$\phi_1 = \frac{4\pi^2 R_1 n_{eff}}{\lambda} \tag{9}$$

where n_{eff} is the effective index. Similarly, the transfer matrix of the ith ring element $\mathbf{P_i}(i = 1, \ldots, 4)$ in Fig. 1(b) reads as:

$$\mathbf{P_i} = \begin{bmatrix} \alpha_i{}^{-1}\!/_2 e^{\frac{j\phi_i}{2}} & 0 \\ 0 & \alpha_i{}^{1}\!/_2 e^{\frac{-j\phi_i}{2}} \end{bmatrix} \tag{10}$$

where α_i is the amplitude transmission factor around the ith ring, and ϕ_i is the phase shift of one turn in the ith ring element, which is given by:

$$\phi_i = \frac{4\pi^2 R_i n_{eff}}{\lambda} \tag{11}$$

From (4) and (8), the relationship between (A_0, B_0) and (A_2, B_2) in Fig. 1(a) can be written as:

$$\begin{bmatrix} A_0 \\ B_0 \end{bmatrix} == \mathbf{M_1} \begin{bmatrix} A_2 \\ B_2 \end{bmatrix} = \mathbf{S_{01} P_1 S_{12}} \begin{bmatrix} A_2 \\ B_2 \end{bmatrix} \tag{12}$$

where the matrix $\mathbf{M_1}$ reads as:

$$\begin{aligned} \mathbf{M_1} &= \mathbf{S_{01} P_1 S_{12}} \\ &= \tfrac{1}{1\times j\kappa_{10}^{\lambda}} \begin{bmatrix} e^{\frac{-j\theta_0^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^{\lambda}}{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & -\kappa_{00}^{\lambda} \\ \kappa_{00}^{\lambda} & -1 \end{bmatrix} \cdot \begin{bmatrix} e^{\frac{-j\theta_0^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^{\lambda}}{2}} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \alpha_1{}^{-1}\!/_2 e^{\frac{j\phi_1}{2}} & 0 \\ 0 & \alpha_1{}^{1}\!/_2 e^{\frac{-j\phi_1}{2}} \end{bmatrix} \cdot \tfrac{1}{1\times j\kappa_{21}^{\lambda}} \cdot \begin{bmatrix} e^{\frac{-j\theta_1^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_1^{\lambda}}{2}} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} 1 & -\kappa_{11}^{\lambda} \\ \kappa_{11}^{\lambda} & -1 \end{bmatrix} \cdot \begin{bmatrix} e^{\frac{-j\theta_1^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_1^{\lambda}}{2}} \end{bmatrix} \\ &= -\tfrac{1}{\kappa_{10}^{\lambda}\kappa_{21}^{\lambda}} \begin{bmatrix} e^{\frac{-j\theta_0^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^{\lambda}}{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & -\kappa_{00}^{\lambda} \\ \kappa_{00}^{\lambda} & -1 \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \alpha_1{}^{-1}\!/_2 e^{\frac{j\left(\phi_1-\theta_0^{\lambda}-\theta_1^{\lambda}\right)}{2}} & 0 \\ 0 & \alpha_1{}^{1}\!/_2 e^{\frac{-j\left(\phi_1-\theta_0^{\lambda}-\theta_1^{\lambda}\right)}{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & -\kappa_{11}^{\lambda} \\ \kappa_{11}^{\lambda} & -1 \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} e^{\frac{-j\theta_1^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_1^{\lambda}}{2}} \end{bmatrix} \end{aligned} \tag{13}$$

Under the configuration of N microrings, the following relationship is satisfied:

$$\begin{aligned} &\begin{bmatrix} A_0 \\ B_0 \end{bmatrix} = \\ &\mathbf{S_{01} P_1 S_{12} P_2 S_{23} \ldots P_i S_{i(i+1)} \ldots P_N S_{N(N+1)}} \begin{bmatrix} A_{N+1} \\ B_{N+1} \end{bmatrix} \\ &= \mathbf{M_N} \begin{bmatrix} A_{N+1} \\ B_{N+1} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} A_{N+1} \\ B_{N+1} \end{bmatrix} \end{aligned} \tag{14}$$

Similar to (13), $\mathbf{M}_N$ can be expressed as:

$$\begin{aligned} \mathbf{M}_N &= \mathbf{S_{01} P_1 S_{12} P_2 S_{23} \ldots P_i S_{i(i+1)} \ldots P_N S_{N(N+1)}} \\ &= \frac{1}{j^{N+1}\kappa_{10}^{\lambda}\kappa_{21}^{\lambda}\cdots\kappa_{(N+1)N}^{\lambda}} \begin{bmatrix} e^{\frac{-j\theta_0^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_0^{\lambda}}{2}} \end{bmatrix}, \\ &\quad \left(\prod_{i=1}^{N} \begin{bmatrix} 1 & -\kappa_{(i-1)(i-1)}^{\lambda} \\ \kappa_{(i-1)(i-1)}^{\lambda} & -1 \end{bmatrix}, \right. \\ &\quad \left. \begin{bmatrix} \alpha_i{}^{-1}\!/_2 e^{\frac{j\left(\phi_i-\theta_{i-1}^{\lambda}-\theta_i^{\lambda}\right)}{2}} & 0 \\ 0 & \alpha_i{}^{1}\!/_2 e^{\frac{-j\left(\phi_i-\theta_{i-1}^{\lambda}-\theta_i^{\lambda}\right)}{2}} \end{bmatrix} \right) \\ &\quad \cdot \begin{bmatrix} 1 & -\kappa_{NN}^{\lambda} \\ \kappa_{NN}^{\lambda} & -1 \end{bmatrix} \cdot \begin{bmatrix} e^{\frac{-j\theta_N^{\lambda}}{2}} & 0 \\ 0 & e^{\frac{j\theta_N^{\lambda}}{2}} \end{bmatrix} \end{aligned} \tag{15}$$

Assuming there exists no input at the add port, B_{N+1} is hence supposed to be zero. From (14), the transmittance can be expressed as:

$$T = \frac{1}{|M_{11}|^2} \tag{16}$$

It can be seen from (14)–(16) that when there are N rings within the filter, the distinct phase shift between individual resonators can indeed bring about different resonant frequencies within the same filter, resulting in spectrum distortions. Furthermore, the difference in the resonance frequency of each ring causes the distortion of the transmission spectrum. It is for this reason that we compensate for spectrum distortion by automatically adjusting the radii in the trained network.

Based on the above analysis, given the material and cross-section of the waveguide, the factors affecting the transmittance are the radii of the rings and the gaps between each component. At the same time, since the radius directly affects the FSR, we first determine the initial value of the radius for the required FSR. Then we take the coupling gaps between various components and the adjustment amount of the radii as the geometric parameters that affect the transmittance. After analyzing the structural parameters affecting the transmission spectrum, we randomly sample the structural parameters within the boundary of the structural parameters and collect the corresponding transmittance. Then we make the collected structural parameters and transmittance together to form the dataset for model training and selection.

APPENDIX B

In this part, we give the calculation method for determining the valid boundaries of the adjustment margins of the ring radii. Given the material and waveguide cross-section, the radii of the microrings and the gaps between the structural components have a combined effect on the transmittance shapes. Meanwhile, the radii affect FSR, which is an essential indicator in the application of filters. Therefore, we first get the initial radii of the microrings according to the FSR requirement, which is defined as [47], [48]:

$$FSR = \frac{(\lambda)^2}{2\pi R n_g} \tag{17}$$

where $\lambda_{t\arg et}$ is the target resonant wavelength, and n_g is the group refractive index. For simplicity, we assume that the initial radii of all microrings are the same denoted as R. We acquire R according to the target FSR denoted as $FSR_{t\arg et}$ from the following relationship:

$$\frac{(\lambda_{t\arg et})^2}{2\pi R n_g} = FSR_{t\arg et} \tag{18}$$

Due to the existence of CIFS, the resonance condition becomes:

$$\frac{4\pi^2 R n_{eff}}{\lambda_i} + \varphi_i = m_i \times 2\pi \tag{19}$$

where m_i is an integer, and φ_i is the coupling-induced phase shift of the ith ring, λ_i is the actual resonant wavelength of the ith ring, and n_{eff} is the effective refractive index. To keep the resonance wavelength unchanged and correct the distorted spectrum caused by CIFS, we adjust the radii as follows by the model:

$$\begin{aligned} |dR_i| &= \left| \frac{\lambda_i d\varphi_i}{4\pi^2 n_{eff}} \right| \leq \left| \frac{\lambda_i}{2\pi n_{eff}} \right| \\ &= dR_i^{\max} \end{aligned} \tag{20}$$

From (20), we can get a rough estimation of the boundaries of dR_i, within which we get random sampling values of dR_i for resonance tuning in the process of inverse design.

APPENDIX C

In this part, we give the details of networks for the transmittance prediction and inverse design. The more layers and the more neurons in each layer, the larger the capacity of the network will have. Here, the network capacity can be viewed as a measure of how readily the model can approximate the complex functions.

REFERENCES

[1] X. Xu *et al.*, "Orthogonally polarized RF optical single sideband generation and dual-channel equalization based on an integrated microring resonator," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4808–4818, 2018.

[2] S. Palermo *et al.*, "Silicon photonic microring resonator-based transceivers for compact WDM optical interconnects," in *Proc. IEEE Compound Semicond. Integr. Circuit Symp.*, 2015, pp. 1–4.

[3] P. De Heyn *et al.*, "Fabrication-tolerant four-channel wavelength-division-multiplexing filter based on collectively tuned si microrings," *J. Lightw. Technol.*, vol. 31, no. 16, pp. 3085–3092, 2013.

[4] M. S. Dahlem, C. W. Holzwarth, A. Khilo, F. X. Kartner, H. I. Smith, and E. P. Ippen, "Reconfigurable multi-channel second-order silicon microring-resonator filterbanks for on-chip WDM systems," *Opt. Exp.*, vol. 19, no. 1, pp. 306–316, 2011.

[5] M. Hossein-Zadeh and K. J. Vahala, "Importance of intrinsic-Q in microring-based optical filters and dispersion-compensation devices," *IEEE Photon. Technol. Lett.*, vol. 19, no. 14, pp. 1045–1047, Jul. 2007.

[6] S. Liao and L. Ou, "High-Q interstitial square coupled microring resonators arrays," *IEEE J. Quantum Electron.*, vol. 56, no. 4, Aug. 2020, Art. no. 6500208.

[7] W. Li *et al.*, "Ultra-compact on-chip optical filters with narrow bandwidth," *Mod. Phys. Lett. B*, vol. 34, no. 22, 2020, Art. no. 2050232.

[8] L. Jin *et al.*, "Optical multi-stability in a nonlinear high-order microring resonator filter," *APL Photon.*, vol. 5, no. 5, 2020, Art. no. 056106.

[9] J. C. C. Mak, W. D. Sacher, T. Xue, J. C. Mikkelsen, Z. Yong, and J. K. S. Poon, "Automatic resonance alignment of high-order microring filters," *IEEE J. Quantum Electron.*, vol. 51, no. 11, Nov. 2015, Art. no. 0600411.

[10] S. Xiao and M. H. Khan and H. Shen, and M. Qi, "Silicon-on-insulator microring add-drop filters with free spectral ranges over 30 nm," *J. Lightw. Technol.*, vol. 26, no. 2, pp. 228–236, 2008.

[11] Y. Yanagase and S. Suzuki and Y. Kokubun, and S. T. Chu, "Box-like filter response and expansion of FSR by a vertically triple coupled microring resonator filter," *J. Lightw. Technol.*, vol. 20, no. 8, 2002, Art. no. 1525.

[12] P. Rabiei, W. H. Steier, C. Zhang, and L. R. Dalton, "Polymer microring filters and modulators," *J. Lightw. Technol.*, vol. 20, no. 11, 2002, Art. no. 1968.

[13] B. E. Little, S. T. Chu, H. A. Haus, J. Foresi, and J.-P. Laine, "Microring resonator channel dropping filters," *J. Lightw. Technol.*, vol. 15, no. 6, pp. 998–1005, 1997.

[14] C. Sitawarin, W. Jin, Z. Lin, and A. W. Rodriguez, "Inverse-designed photonic fibers and metasurfaces for nonlinear frequency conversion," *Photon. Res.*, vol. 6, no. 5, pp. B82–B89, 2018.

[15] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vuckovic, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nature Photon.*, vol. 12, no. 11, pp. 659–670, 2018.

[16] J. Peurifoy *et al.*, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, no. 6, 2018, Art. no. 4206.

[17] A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, and J. Vuckovic, "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," *Nature Photon.*, vol. 9, no. 6, pp. 374–377, 2015.

[18] L. Yu, R. S. Kokenyesi, D. A. Keszler, and A. Zunger, "Inverse design of high absorption thin-film photovoltaic materials," *Adv. Energy Mater.*, vol. 3, no. 1, pp. 43–48, 2013.

[19] A. Yariv and P. Yeh, *Photonics: Optical Electronics in Modern Communications*. Oxford, U.K.: Oxford Univ. Press, 2007.

[20] L. Zhang *et al.*, "Cascading second-order microring resonators for a box-like filter response," *J. Lightw. Technol.*, vol. 35, no. 24, pp. 5347–5360, 2017.

[21] V. Van, *Optical Microring Resonators: Theory, Techniques, and Applications*. Boca Raton, FL, USA: CRC Press, 2016.

[22] G. Chen and C. Jiang, "Reverse design of microring resonator channel dropping filters," *Results Phys.*, vol. 19, 2020, Art. no. 103380.

[23] M. Radmilovic-Radjenovic and B. Radjenovic, "Eigenmode and frequency domain analysis of the third-order microring filters," *Opt. Quantum Electron.*, vol. 50, no. 6, pp. 1–11, 2018.

[24] G. Bellanca, P. Orlandi, and P. Bassi, "Assessment of the orthogonal and non-orthogonal coupled-mode theory for parallel optical waveguide couplers," *JOSA A*, vol. 35, no. 4, pp. 577–585, 2018.

[25] P. Xia, Y. Fu, M. Kong, Z. Liu, J. Zhou, and J. Zhou, "Theoretical analysis of the mode splitting properties in periodically patterned microring resonators," *J. Lightw. Technol.*, vol. 35, no. 9, pp. 1700–1704, 2017.

[26] B. Radjenovic, M. Radmilovic-Radjenovic, and P. Belicev, "Eigenmodes of finite length silicon-on-insulator microring resonator arrays," *Opt. Quantum Electron.*, vol. 49, no. 4, pp. 1–17, 2017.

[27] J. Torres, A. Baptista, and V. M. Machado, "Coupling analysis in concentric ring waveguides," *J. Lightw. Technol.*, vol. 31, no. 13, pp. 2140–2145, 2013.

[28] Q. Li, M. Soltani, A. H. Atabaki, S. Yegnanarayanan, and A. Adibi, "Quantitative modeling of coupling-induced resonance frequency shift in microring resonators," *Opt. Exp.*, vol. 17, no. 26, pp. 23474–23487, 2009.

[29] M. A. Popovic, C. Manolatou, and M. R. Watts, "Coupling-induced resonance frequency shifts in coupled dielectric multi-cavity filters," *Opt. Exp.*, vol. 14, no. 3, pp. 1208–1222, 2006.

[30] L. Y. Tobing, S. Darmawan, D. R. Lim, M.-K. Chin, and T. Mei, "Relaxation of critical coupling condition and characterization of coupling-induced frequency shift in two-ring structures," *IEEE J. Sel. Topics Quantum Electron.*, vol. 16, no. 1, pp. 77–84, Jan./Feb. 2010.
[31] Q. Li, M. Soltani, S. Yegnanarayanan, and A. Adibi, "Design and demonstration of compact, wide bandwidth coupled-resonator filters on a silicon-on-insulator platform," *Opt. Exp.*, vol. 17, no. 4, pp. 2247–2254, 2009.
[32] A. M. Hammond, "Machine learning methods for nanophotonic design, simulation, and operation," M. S. thesis, Dept. Elect. Comput., Brigham Young Univ., 2019.
[33] A. M. Hammond and R. M. Camacho, "Designing integrated photonic devices using artificial neural networks," *Opt. Exp.*, vol. 27, no. 21, pp. 29620–29638, 2019.
[34] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd ed. Birmingham, U.K.: Packt Publishing, 2017.
[35] D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training deep neural networks for the inverse design of nanophotonic structures," *Acs Photon.*, vol. 5, no. 4, pp. 1365–1369, 2018.
[36] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.
[37] G. E. Hinton, N. Srivastava, A. Krizhevsky, R. R. Salakhutdinov, and I. Sutskever, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, Jul. 2012.
[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[39] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proc. Int. Joint Conf. Neural Netw.*, WA, DC, USA, 1989, pp. 593–605.
[40] M. T. Hagan, H. B. Demuth, M. Beale, and O. De Jesús, *Neural Network Design*, 2nd ed. New York, NY, USA: Martin, 2014.
[41] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.
[42] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, no. 7, pp. 441–446, 2017.
[43] 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.
[44] A. Graves *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.
[45] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[46] T. Barwicz, M. R. Watts, and M. Popovic, "Polarization-transparent microphotonic devices in the strong confinement limit," *Nature Photon.*, vol. 1, no. 1, pp. 57–60, 2007.
[47] A. Noury, X. Le Roux, L. Vivien, and N. Izard, "Controlling carbon nanotube photoluminescence using silicon microring resonators," *Nanotechnology*, vol. 25, no. 21, 2014, Art. no. 215201.
[48] S. Feng, T. Lei, H. Chen, H. Cai, X. Luo, and A. W. Poon, "Silicon photonics: From a microresonator perspective," *Laser Photon. Rev.*, vol. 6, no. 2, pp. 145–177, 2012.
[49] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, "MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method," *Comput. Phys. Commun.*, vol. 181, no. 3, pp. 687–702, 2010.
[50] M. Bahadori *et al.*, "Design space exploration of microring resonators in silicon photonic interconnects: Impact of the ring curvature," *J. Lightw. Technol.*, vol. 36, no. 13, pp. 2767–2782, 2018.