Deep Transfer Learning for Nanophotonic Device Design

Kojima, Keisuke; Jung, Minwoo; Koike-Akino, Toshiaki; Wang, Ye; Brand, Matthew E.; Parsons, Kieran

TR2022-107 September 29, 2022

Abstract

Applying a transfer-learning technique for generative deep neural networks, we demonstrate a very time-efficient inverse design framework for photonic integrated circuit devices, when there are new demands for structural/material parameters from an existing device library.

Conference on Lasers and Electro-Optics (CLEO) Pasific Rim 2022

Deep Transfer Learning for Nanophotonic Device Design

Keisuke Kojima[1], Minwoo Jung[1,2], Toshiaki Koike-Akino[1,*], Ye Wang[1], Matthew Brand[1], Kieran Parsons[1]

[1]*Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.*
[2]*Department of Physics, Cornell University, Ithaca, NY 14853, USA.*
**koike@merl.com, brand@merl.com*

Abstract: Applying a transfer-learning technique for generative deep neural networks, we demonstrate a very time-efficient inverse design framework for photonic integrated circuit devices, when there are new demands for structural/material parameters from an existing device library.

1. Introduction: In recent years, generative deep neural networks (DNN) have been successfully applied to time-efficient inverse design of photonic devices [1,2]. For example, a DNN model optimized for a discrete set of parameters can generate devices with arbitrary parameters analogous to interpolation. When the DNN is trained with a sufficient dataset (device topologies and corresponding transmission spectra), high-performance device topologies are generated given a target spectrum. However, if the process or material parameters are changed, the DNN will often fail to generate high-quality device topologies appropriate for the new domain. Transfer learning addresses this domain shift issue [3]. The DNN can adapt to the new domain (e.g., different waveguide thickness) by updating a set of layers of network using a small amount of training data from the new domain. In this work, we take nanophotonic wavelength splitters as an example, and apply transfer learning to accommodate a different waveguide thickness. With a small number of additional training data, the updated DNN generates better devices achieving higher extinction ratios (ER).

2. Wavelength Splitter: Our target device structure is based on the compact on-chip wavelength de-multiplexer (1 input and 2 output ports) as shown in Fig. 1 (a). This is a silicon-on-insulator (SOI) structure, with a Si waveguide thickness of 220 nm. The optimization area is a 4.5×4.5 $\mu\mathrm{m}^2$ square, represented by 181×181 pixels. Using the LumOpt numerical package [4] provided by Lumerical, we first perform the adjoint optimization method [5] for the following target responses: $T_1(\lambda_1) = T_2(\lambda_2) = T_{\mathrm{high}}$ and $T_1(\lambda_2) = T_2(\lambda_1) = T_{\mathrm{low}}$, where $T_{1/2}$ is the transmission of the waveguide mode from the input port to the output ports 1/2. Ideally, $T_{\mathrm{high}} = 1$ and $T_{\mathrm{low}} = 0$ are desired for the best performance. All transmission values are averaged over 5nm bandwidth around the target wavelengths $\lambda_{1,2} \pm 2.5$nm. Fig. 1(a) shows an example device topology for $\bar{\lambda} = (\lambda_1 + \lambda_2)/2 = 1550$nm and $\Delta\lambda = |\lambda_1 - \lambda_2| = 50$nm, and the resulting device response is drawn in Fig. 1(b). The ER is defined as the average of the extinction ratios at two wavelength bands.

3. DNN Model: The goal of the inverse design in this device is to generate a useful device topology (denoted as T), given a desired device performance, e.g., transmission spectra (denoted as S). The network is called an ACVAE (Adversarial Conditional Variational Encoder) [1], which is an extension of the well-established CVAE [6]. Since we treat the input topology as an image (i.e. pixels), we use convolutional neural networks (CNNs) for the encoder and the decoder. Further, cycle consistency was added to improve the inverse design capability [7].

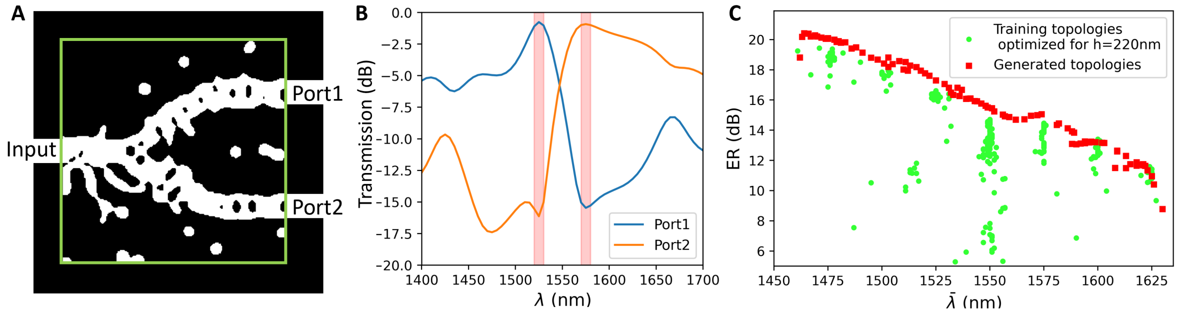

Fig. 1. (A) Optimized device topology (black: silicon oxide background, white: silicon waveguide, box: optimization area of 4.5×4.5 $\mu\mathrm{m}^2$) and (B) transmission spectra of an example device generated by DNN. (C) Performance of densely generated topologies (red squares) from DNN trained with a sparse set of topologies optimized for 220 nm-thick devices (green dots).

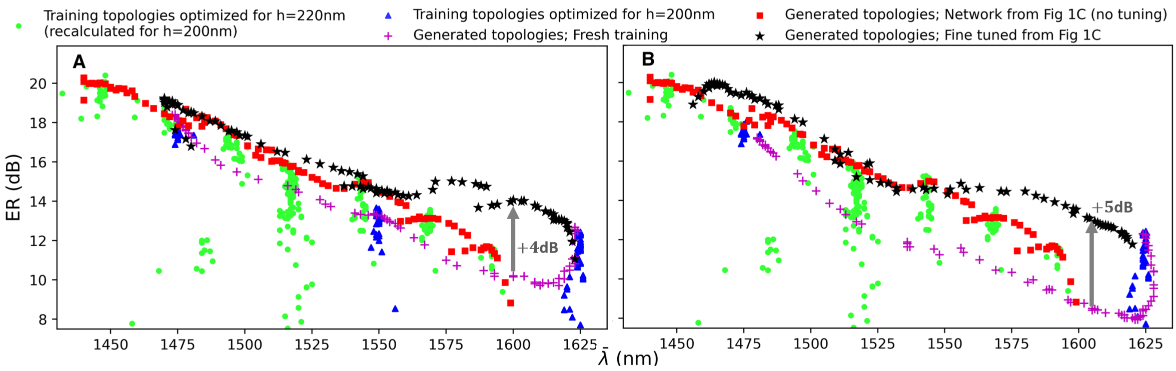

Fig. 2. Average extinction ratio vs. center wavelength $\bar{\lambda}$. Green dots and blue triangles show the training topologies for $h = 220$ nm and $h = 200$ nm, respectively, purple plus signs and red squares indicate the generated devices with the DNN trained with the data for $h = 220$ nm and $h = 200$ nm, respectively. The black starts are the generated devices using the transfer learning. (A) contains three series and (B) contains two series of the training data for $h = 200$ nm. All the ERs are evaluated with FDTD for $h = 200$ nm.

4. Baseline Training Data and Validation: The baseline DNN training dataset was prepared by the following procedure: (1) optimize a device topology targeted for $\bar{\lambda} = 1550$nm starting from a random initial condition, given a separation $\Delta\lambda = 50$ nm and waveguide thickness $h = 220$ nm; (2) take the topology optimized at $\bar{\lambda} = 1550$nm as the initial condition for the optimization targeting for $\bar{\lambda} = 1575$nm; and (3) cascade the previous step for every 25nm of target $\bar{\lambda}$ values. The steps (1)–(3) ensure smoothness in T across varying S' within a series of such cascades. The adjoint optimization produces many ($\sim$100) intermediate sub-optimal results en route to the final optimal design. Using these training data, the DNN was trained, and by sweeping the target center wavelength $\bar{\lambda}$ from 1475nm to 1625nm, we generate 91 devices and validate by finite-difference time-domain (FDTD) simulations, demonstrating that they are filling the gap and generally exceeding the training data, as shown in Fig. 1 (c).

5. Transfer Learning: We now consider to design devices for a new domain with $h = 200$ nm, by transferring the knowledge at $h = 220$ nm. The green dots in Fig. 2 shows the previous baseline training set evaluated at $h = 200$ nm. The center wavelength is shifted towards the shorter wavelength, and ER is generally degraded. The generated devices from the DNN using only the baseline training data are shown as red squares, also showing shifted and degraded performances. The blue triangles show new training data for the 200 nm thickness at $\bar{\lambda} = 1475$ nm, 1550 nm, and 1625 nm for Fig. 2 (a) and $\bar{\lambda} = 1475$ nm and 1625 nm for Fig. 2 (b). The purple plus sign makers show the generated devices when the DNN was using only the new training set, indicating the limitation when the training data are sparse, especially at the longer wavelength region. Now, the transfer learning is applied to the original DNN, with unfreezing all the layers while training only with the new training data set. The validation results of the transfer learning are shown as the black starts. This clearly demonstrate that transfer learning outperforms the DNN trained with the baseline training data, or with only the new training data, which is especially clear at the longer wavelength. One FDTD simulation takes 1 minute, and one adjoint method optimization run takes 2–3 hours using a cluster with 36 cores. So it takes one and a half day to collect the baseline and additional training dataset, respectively. Once the training dataset are collected, to train the DNN takes 6 minutes using a graphic processsing unit, and 1 second to generate the whole 91 devices.

6. Summary: In designing photonic devices, we often need to deviate from the original conditions, such as structural or material parameters. Focusing on nanophotonic wavelength splitters as an example, we demonstrated that transfer learning can effectively adapt to the new waveguide thickness with fewer training data. This demonstrates the versatility and the effectiveness of the DNN framework.

References

1. Y. Tang et. al., Laser Photonics Rev. **14**(12), 2000287 (2020).
2. K. Kojima et. al., IEEE J. Lightwave Techol. **39** (4), 1010 (2021).
3. S. J. Pan, Q. Yang, IEEE Transactions on knowledge and data engineering **22** (10), 1345 (2010).
4. C. Lalau-Keraly. https://lumopt.readthedocs.io, 2018.
5. A. Y. Piggott et. al., Nat. Photon. **9**, 374-7 (2015).
6. K. Sohn et al., in: Advances in neural information processing systems, 3483-3491 (2015).
7. Y. Tang et al., Paper SW4E, CLEO 2021.