Designing Silicon Photonic Devices using Artificial Neural Networks

Alec M. Hammond, Ryan M. Camacho

Abstract

We develop and experimentally validate a novel neural network design framework for silicon photonics devices that is both practical and intuitive. The framework is applicable to nearly all known integrated photonics devices, but as case studies we consider simple waveguides and chirped Bragg Gratings. By using artificial neural networks, we decrease the computational cost relative to traditional design methodologies by more than 4 orders of magnitude. We also demonstrate the abstraction of the device models to a few simple input and output parameters relevant to designers. We then apply the results to various design problems and experimentally compare fabricated devices to the neural network's predictions.

1 Introduction

Silicon photonics has become a viable technology for integrating a large number of optical components in a chip-scale format. Driven primarily by telecommunications applications, a growing number of CMOS fabrication facilities dedicated to silicon photonics are now in operation and available for researchers and engineers to submit photonic integrated circuit (PIC) designs to be fabricated [1].

Designing the silicon photonics components themselves, however, remains a major bottleneck. Current design flows are impeded by computational tractability and the need for designers with extensive experience. [2]. Unlike their electronic counterparts, photonic integrated circuits require computationally expensive simulation routines to accurately predict their optical response functions. Instead of abstracting the circuit in terms of simple voltages and currents, Maxwell's equations must often be solved numerically using techniques such as the Finite Difference Time Domain (FDTD) method or the Finite Element Method (FEM)[3], which are computationally expensive. In fact, the typical time to design integrated photonic devices now often exceeds the time to manufacture and test them.

To address this challenge, we propose and experimentally validate a new design paradigm for silicon photonics that leverages artificial neural networks (ANN). Artificial neural networks model the relationship between their inputs and outputs by cascading various computational units, known as neurons [4]. ANNs learn the desired relationship between inputs and outputs by tuning each neuron in response to a training set. Training algorithms, such as backpropagation, are used to strategically introduce these datasets into the neural network and gradually update the network's parameters until a convergence criteria is met [5]. If the network architecture is sufficiently large, any arbitrary relationship manifested in the dataset can be modeled [6]. Importantly, while the training itself can be computationally expensive, once the ANN is properly trained, it is computationally inexpensive to predict the response of a new unknown system anywhere within the parameter space of the training set.

The primary task in implementing an ANN design framework for silicon photonics components, then, is developing training sets and then training ANN architectures that are intuitive and correctly predict the optical response of integrated optical components. Owing to the complexity of field propagation within integrated photonic devices, this is difficult and often requires several training attempts. Furthermore, significant complexity exists in choosing and training a suitable training set, since the ANN may inadvertently learn unintended biases manifested in the data.

Despite these challenges, photonic components are ideal candidates for ANN modeling since existing methods such as FDTD and FEM can be used to construct robust training sets. With the growing availability of vast cloud-based computational resources, several million training simulations can now be run in hours or days. [7]. Once trained, a neural network can reliably interpolate between training data, is compact and easily shared with the community, and can even continue to learn on new datasets via transfer learning [8]. Thus, the computational complexity inherent in designing integrated photonic devices can be moved to the front end of the design process, allowing individual designers to work with abstracted components whose optical response can be rapidly calculated.

ANNs are now commonly used by radio frequency

(RF) engineers, who train networks to model transistors, oscillators, and antennas [9, 10, 11]. In addition, recent theoretical results have shown that it is possible for ANNs to be trained to model nanophotonic structures. For example, Ferreira et al. demonstrated that ANNs could assist with the numerical optimization of a waveguide coupler by randomly assigning radii and refractive index values in a large array of micropillars [12]. Using a similar method, Tahersima et al. demonstrated that by randomly assigning a refractive index to grid points in a 2D-array, the splitting ratio of an integrated photonic splitter could be optimized [13]. In both cases the input parameter space was the entire 2D array of grid points, showing the power of ANNs in blind "black box" approach, though limiting the designer's ability to intuitively adjust input parameters.

A related approach has also been applied to periodic photonic structures, which are often difficult to efficiently to model. Inampudi et al. showed that a diffraction grating could be optimized by allowing an ANN to train on a large array of simulated geometric "units" with arbitrary shapes [14]. Similarly, Ferreira et al. showed that the dispersion relation of photonic crystals can be modeled by allowing the ANN to train over varying sizes of several circular shapes within a rectangular unit cell [15]. While not offering an intuitive design flow, both of these results showed the ability of ANNs to optimize previously intractable simulation problems.

Two recent theoretical papers by Zhang et al [16], and Peurifoy et al. [17] go beyond simply optimizing over a large parameter space, and used ANNs to calculate complicated spectra using a smaller number of intuitive, smoothly varying input parameters. These results are the closest scientifically to this work. Zhang et al. predict the response of plasmonic waveguide cavities using a small number of geometric input parameters such as waveguide widths and lengths. Peurifoy et al. train an ANN to model light scattering from a perfect 8-layer sphere with input parameters consisting of layer thicknesses and refractive index. Even though in both cases each wavelength point in the calculated spectra required its own ANN output neuron, the usefulness of using ANNs to model systems with intuitive input parameters was demonstrated.

In this work we develop a new design framework for silicon photonics using ANNs and report on the first experimental validation of ANNs as a design tool by fabricating and testing a variety of devices. We also demonstrate for the first time the use of ANNs with a small number of descriptive input *and* output neurons corresponding to descriptive design parameters. In contrast, all previous work known to the authors using ANNs to desing photonic devices is theoretical and requires an output neuron for each wavelength point of interest. While the ANNs we describe are more difficult to train, once developed these ANNs have two primary advantages: (1) A continuous spectral output response allowing for detection of sharp features and built-in interpolation, and (2) A simplified constraint space for device designers.

To motivate our approach, we first describe a neural network that models the effective index of a silicon photonic strip waveguide with various widths and thicknesses. While waveguide simulation is already straightforward from a designer's perspective, the model illustrates the advantages of our approach and is a key building block for more advanced ANN models described below which are less straightforward using existing techniques. This includes the ANN's computational speedup of over 4 orders of magnitude, and the simplification and speedup of other complicated simulation routines that rely on effective index calculations. A more advanced example that is computationally intractable via traditional methods is then given, in which we demonstrate an ANN that models the complex relationship between a chirped silicon photonic Bragg grating's design parameters and its corresponding spectral response. Many designers leverage silicon photonic chirped Bragg gratings to equalize optical amplifier gain [18], compensate for semiconductor laser dispersion [19, 20], and enable nonlinear temporal pulse compression [21, 22].

Finally, we illustrate various forward design and inverse design applications using the developed ANNs. It is hoped that our proposed design methodology will become a powerful tool for future innovation within integrated optics platforms and encourage a new level of collaboration and participation within the community.

2 Results

Overview

We present two different device ANNs with forward and inverse design applications. Figure 1 illustrates the new design methodology. First, we iterated between generating an appropriate dataset and training the ANN until the model adequately characterized the device. Next, we used the ANN to simulate circuits and solve inverse design problems. Finally, we fabricated devices to validate the results.

Waveguide Neural Network

We first report on a simple waveguide neural network capable of estimating the effective index of any arbitrary silicon photonic waveguide geometry for a variety of modes. Specifically, we modeled the relationship between the waveguide's width, thickness, and operating wavelength and the effective index for the first two TE and TM modes. We note that including wavelength as an input parameter is a unique and enabling strategy not previously adopted (see Discussion section below). Figure 2 (f) compares the ANN's predicted effective index to a its corresponding simulation. The first TE and TM mode for any silicon photonic waveguide with a width between 350 nm and 1000 nm and a thickness between 150 nm and 350 nm are demonstrated. The network estimates a smooth response for both modes simultaneously, even for data points outside of its training set.

We implemented various tests to validate the network's accuracy. First, we split the initial dataset into a training set and a validation set. While the network evaluated both sets after each epoch (i.e. training iteration) only the training set's results were used to update the network's weights. We monitored the validation set's results to assess overfitting. To better understand the network's performance after each iteration, we recorded each epoch's mean-square-error (MSE) and coefficient of determination (R^2). Figure 2 (a) and (b) illustrate the MSE and R^2 respectively after each epoch. To prevent overfitting, we stopped training at 100 epochs, where the MSE and R^2 appear to converge. At this point, the network demonstrated a MSE of 1.323×10^{-4} for the training set and 7.490×10^{-5} for the validation set. The final R^2 values for the training data and validation data were 0.9996 and 0.9997 respectively. The MSE and R^2 evolution for both the training set and validation set converge well, indicating little to no overfitting. Figure 2 (c) illustrates the relative error for both the training and validation sets after the final epoch. Both the training set errors and validation set errors seem similarly distributed and tightly bounded between -1% and 1%, once again indicating little to no overfitting.

With confidence in the waveguide neural network's prediction accuracy, we benchmarked its speed and found that a single neural network evaluation was 10^4 times faster on average than the corresponding finite difference eigenmode simulation. Figure 2 (d) compares the computation speed for the ANN to the eigenmode solver, Meep Photonic Bands (MPB). This significant speedup enables many simulation techniques, like the layered dielectric media transfer matrix method (LDMTMM) [23] or the eigenmode expansion method (EMM) [24], where photonic components are discretized into individual waveguides. Using the ANN, a transfer matrix for each waveguide can be quickly generated and cascaded to formulate a fairly accurate response for the device. In addition, modeling fabrication variations is now much quicker since existing Monte Carlo sampling routines can leverage the ANN's speed.

Bragg Grating Neural Network

Modeling the relationship between a Bragg grating's physical design parameters and its corresponding responses is difficult since no one-to-one mapping exists. Consequently, many designers resort to black-box optimization routines that strategically search the parameter space for viable design options. As a result, inverse design problems — where a simulation needs to run each iteration — become intractable for even modest size gratings. If a full 3D FDTD simulation is performed, for example, each optimization iteration can take between 8-12 hours on typical desktop computing systems. In addition, the optimization routines tend to inefficiently simulate redundant test scenarios for different design problems. We train and demonstrate a Bragg ANN, however, that can predict a grating's response on the order of milliseconds on the same system, enabling much faster solutions to more complex design problems. We fabricate various test devices and validate our neural network's predictions.

Using the waveguide neural network, we generated a dataset to train our Bragg grating neural network to predict the reflection spectrum and group delay response of a silicon photonic, sidewall-corrugated, linearly chirped Bragg grating, as illustrated in Figure 3 (d). We note that generating the training dataset was approximately 2 orders of magnitude fast using the waveguide ANN reported above rather than traditional methods. To smooth apodization dependent ringing, we pre-processed the training data (see methods). We parameterized the gratings by length of the first period (a_0), the length of the last period (a_1), the number of grating periods NG, and the corrugation width difference($\Delta w = w_1 - w_0$). We designed the network to receive these four parameters along with a wavelength point as inputs. The network has two outputs: optical reflection and group delay. To further validate the Bragg ANN, we fabricated and measured several silicon photonic Bragg gratings with different chirping patterns and compared their reflection spectra to the neural network's response, as shown in Figure 3 (e). The results agree well with the experimental data. Small discrepancies in the grating responses are largely attributed to manufacturing limitations and variations [1].

Similar to the waveguide network, we divided the dataset into a training set and validation set. We tracked both the MSE and the R^2 metrics after each epoch. The Bragg training set was much larger than the waveguide training set, owing to the larger parameter space. Consequently, the MSE converged within the first few epochs and we stopped training after just five epochs to prevent overfitting. The final MSE for the training and validaton sets was 1.8451e-04 and 1.6768e-04 respectively. The final R^2 was 0.9975 and 0.9977 respectively. Once again, the MSE and R^2 evolution for both the training set and validation set converge well, indicating little to no overfitting. Figure 3 (a-b) illustrates the network's MSE and R^2 evolution. Figure 3 (c) illustrates the absolute error for both the training sets and validation sets. We calculated the absolute error because several training samples were at or near zero and skewed the relative error.

We note that calclulating Bragg grating response with the ANN is much more computationally efficient than previously demonstrated methods. This is because the Bragg ANN linearly increases in computation complexity with added grating parameters, while LDMTMM and all other methods known to these authors increase at least quadratically.

Forward design

The neural network's speed and flexibility enable forward design exploration. For example, Figure 4 illustrates a graphical user interface (GUI) built with slider bars to adjust the Bragg grating's design parameters (i.e. corrugation widths, grating length, chirp pattern, etc). The plots dynamically update, calling the neural network every time the user modifies the input, and display the corresponding reflection and group delay profiles. Because wavelength is included as an input to the ANN rather than an output, arbitrary wavelength sampling within the domain is allowed. Computing these responses in real time is not possible using traditional techniques. This capability is valuable and allows novice designers the ability to rapidly gain device intuition without necessarily understanding the underlying numerical techniques.

Inverse design

This new approach also enables an entirely new set of inverse design problems. For example, we used the neural network in conjunction with a truncated Newtonian optimization algorithm to design a temporal pulse compressor. Designers often rely on dispersive Bragg gratings to generate short, optical pulses for high-capacity communications [21]. In this particular case study, we assumed an arbitrary source generates a 20 ps wide chirped pulse with 4 nm of bandwidth. Figure 5 illustrates the optimization routine's evolution, the resulting grating response, and the pulse shape before and after the Bragg grating. Such optimization algorithms run much quicker than previously known methods. The agnostic nature of the neural network interface works well with typical optimization routines, especially since any arbitrary wavelength sampling is allowed. Depending on the cost function formulation, gradient-based methods could directly evaluate the Jacobian and Hessian tensors from the ANN without any extra sampling or discretization.

3 Discussion

Our method demonstrates a new, viable platform for silicon photonic circuit design. Without much ANN architecture tuning, we successfully modeled silicon photonic waveguides and silicon photonic chirped Bragg gratings. Future work could explore new network architectures (e.g. different activation functions, layer connections, etc.) and training algorithms. Several other devices, like ring resonators [25], can also be modeled. One could subsequently cascade several ANNs that model the scattering parameters of different devices, opening the door to large-scale optimization problems.

An important feature of this work is the choice to model the wavelength as a continuous input parameter rather than fix each output neuron at a specific wavelength point. The waveguide ANN, for example, outputs effective index values and the Bragg ANN outputs reflection and group delay values across the entire input spectrum. This approach, while more difficult to train, is more convenient for the designer. For example, an optimization routine tasked with designing a Bragg filter can focus more on parameters like the bandwidth and shape, rather than an arbitrarily sampled wavelength profile. Furthermore, this method doesn't require the training spectra to have the same sampling. Training sets for structures like ring resonators, whose features may require finer wavelength resolution than other devices, can now be strategically simulated to highlight these features. Assuming the network is trained correctly across a suitable domain, the ANN will seamlessly interpolate between both design parameters and wavelength points without any additional routines.

Unlike traditional simulation methods, training an arbitrary device ANN requires large datasets that are too time-intensive for most typical computers. Fortunately, several affordable and parallelizable high performance computing platforms are widely available. Once the dataset is generated and the ANN sufficiently trained, the ANN can be easily packaged and delivered for widespread collaboration. Furthermore, designers can build off of existing ANNs via transfer learning. These features motivate a new culture within the PIC community.

As is always the case with deep learning applications, the network's utility is limited by biases induced by factors like the training set, the network architecture, or even the training process itself [26]. Fortunately, we can anticipate these biases by extracting the model's prediction uncertainty without modifying our network architecture. Dropout inference techniques leverage models that rely on dropout layers to mitigate over-fitting (a form of network bias) [27]. Even pre-trained networks can use dropout inference to extract prediction uncertainties without any modifications to the network. This particular network design methodology opens the door to many more applications, like training on fabricated device data. Foundry's that develop process design kits (PDKs), for example, can use this technique to model their fabrication processes while preserving their trade secrets.

4 Methods

Training data generation and preprocessing

We generated our waveguide neural network's training set on a high performance computing cluster (HPC) using MEEP Photonic Bands (MPB) [28], a finite difference eigenmode solver. The solver simulated 31 different waveguide widths from 350 nm to 1500 nm and 31 waveguide thicknesses ranging from 150 nm to 400 nm resulting in 961 different geometries. The solver simulated 200 distinct wavelength points in the range of 1400 nm to 1700 nm. The total number of training samples fed into the neural network was 192,200. 70% of the data set was used as training samples and the remaining 30 % was used as validation samples. Each sample had three inputs (width, thickness, and wavelength) and four corresponding outputs (effective indices for the first two TE and TM modes). No postprocessing was performed on the waveguide training data.

On the same HPC, we generated our Bragg training set by simulating 104,131 different gratings with the layered dielectric media transfer matrix method (LDMTMM) [23]. The LDMTMM method models each individual section of the Bragg grating as an ideal waveguide and cascades each sections's corresponding transfer matrix to estimate the grating's response for each wavelength point of interest. We calculated each individual waveguide's effective index using the waveguide neural network. Our simulations swept through 10 different corrugation widths from 10 nm to 100 nm, 11 different grating lengths from 100 periods to 2000 periods, and 32 different chirping patterns.

Once the grating spectra were generated, we curve fit the results to reduce ringing and to generalize the grating's response to arbitrary apodization profiles. We found that without curve-fitting, the resulting oscillations significantly complicate the training process and restrict the network's domain to a single apodization. We fit both the reflection spectrum and group delay responses to generalized Gaussians and resampled the results with 250 wavelength points from 1.45 μm to 1.65 μm. Since the nonlinear curve fitting routine occasionally failed, not all of the simulated gratings were appropriate for testing. After filtering through the results, we generated a database of 26,032,750 training samples.

Neural network design and training

Both neural networks were trained on the same HPC cluster using Keras [29] and Tensorflow [30]. Several hundred different architectures were tested. To gauge the effectiveness of each architecture, the mean-squared-error and coefficient of determination ($R**2$) metrics were used. The waveguide neural network that worked best had 4 hidden layers with 128 neurons, 64, neurons, 32 neurons, and 16 neurons. Each neuron used a hyperbolic tangent activation function. The Bragg grating neural network was designed with 10 hidden layers and 128 neurons each. RELU activation functions were used. Both networks were trained with 16 sample batch sizes. While the waveguide neural network was trained with 100 epochs, the Bragg grating neural network only needed about 5 epochs to reach sufficient results, primarily due to the large training set. The Bragg training set was normalized to improve the network's expressive capabilities.

Simulation benchmarks

We performed all benchmarks using a quad-core Intel(R) i5-2400 CPU clocked at 3.10GHz with 12 GB of RAM. To evaluate the waveguide ANN's speed, we simulated various waveguide parameters in serial using both the ANN and MPB. To evaluate the BG ANN's speed, we simulated various grating's in serial using both the ANN and the LDMTMM. We linearly fit each method's results and compared the slopes to examine the speedup.

Device fabrication

The silicon photonic Bragg gratings were fabricated by Applied Nanotools Inc (Edmonton, Canada) using a direct-write 100 keV electron beam lithographic process. Silicon-on-Insulator wafers with 220 nm device thickness and 2 μm thick insulator layer were used. The devices were patterned with a raster step of 5 μm and etched with a ICP-RIE etch process. A 2.2 μm oxide cladding was deposited using a plasma-enhanced chemical vapour deposition (PECVD) process.

Device measurement

Each device was measured using an automated process at the University of British Colombia (UBC). An Agilent 81600B tunable laser was used as the input source and Agilent 81635A optical power sensors as the output detectors. The wavelength was swept from 1500 to 1600 nm in 10 pm steps. A polarization maintaining (PM) fibre was used to maintain the polarization state of the light, to couple the TE polarization in and out of the grating couplers. Several dembedded test structures were used to normalize out the coupler profiles.

5 Conclusion

We theoretically and experimentally demonstrate silicon photonic circuit design using artificial neural networks. A waveguide ANN that models the effective index for various silicon photonic waveguide geometries shows 4 orders of magnitude speedup versus traditional vectorial methods. A chirped Bragg grating network shows similar gains, and is used to solve both forward and inverse design problems. Corresponding fabrication measurments align with the ANN's predictions.

References

[1] L. Chrostowski and M. Hochberg, *Silicon Photonics Design: From Devices to Systems*, 1st ed. Cambridge ; New York: Cambridge University Press, May 2015.

[2] W. Bogaerts and L. Chrostowski, "Silicon Photonics Circuit Design: Methods, Tools and Challenges," *Laser & Photonics Reviews*, vol. 12, no. 4, p. 1700237, Apr. 2018.

[3] K. F. Warnick, *Numerical Methods for Engineering: An introduction using MATLAB and computational electromagnetics examples*. Raleigh, NC: Scitech Publishing, Dec. 2010.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[5] Y. le Cun, "A theoretical framework for back-propagation," *Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA*, pp. 21–28, 1988.

[6] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[7] C. Vecchiola, S. Pandey, and R. Buyya, *High-Performance Cloud Computing: A View of Scientific Applications*. New York: IEEE, 2009.

[8] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[9] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design - From theory to practice," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003.

[10] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 420–435, Jan. 2004.

[11] A. Zaabab, Q. Zhang, and M. Nakhla, "A Neural-Network Modeling Approach to Circuit Optimization and Statistical Design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 6, pp. 1349–1358, Jun. 1995.

[12] A. da Silva Ferreira, C. H. da Silva Santos, M. S. Gonçalves, and H. E. Hernández Figueroa, "Towards an integrated evolutionary strategy and artificial neural network computational tool for designing photonic coupler devices," *Applied Soft Computing*, vol. 65, pp. 1–11, Apr. 2018.

[13] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, "Deep Neural Network Inverse Design of Integrated Nanophotonic Devices," *arXiv:1809.03555 [physics.app-ph]*, Sep. 2018.

[14] S. Inampudi and H. Mosallaei, "Neural network based design of metagratings," *Applied Physics Letters*, vol. 112, no. 24, p. 241102, Jun. 2018.

[15] A. D. Silva Ferreira, G. N. Malheiros-Silveira, and H. E. Hernandez-Figueroa, "Computing Optical Properties of Photonic Crystals by Using Multilayer Perceptron and Extreme Learning Machine," *Journal of Lightwave Technology*, vol. 36, no. 18, pp. 4066–4073, Sep. 2018.

[16] T. Zhang, Q. Liu, J. Dai, X. Han, J. Li, Y. Zhou, and K. Xu, "Spectrum prediction and inverse design for plasmonic waveguide system based on artificial neural networks," *arXiv:1805.06410 [physics]*, May 2018.

[17] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljacic, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Science Advances*, vol. 4, no. 6, p. eaar4206, Jun. 2018.

[18] M. Rochette, M. Guy, S. LaRochelle, J. Lauzon, and F. Trepanier, "Gain equalization of EDFA's with Bragg gratings," *IEEE Photonics Technology Letters*, vol. 11, no. 5, pp. 536–538, May 1999.

[19] D. T. H. Tan, K. Ikeda, R. E. Saperstein, B. Slutsky, and Y. Fainman, "Chip-scale dispersion engineering using chirped vertical gratings," *Optics Letters*, vol. 33, no. 24, pp. 3013–3015, Dec. 2008.

[20] M. J. Strain and M. Sorel, "Design and Fabrication of Integrated Chirped Bragg Gratings for On-Chip Dispersion Control," *IEEE Journal of Quantum Electronics*, vol. 46, no. 5, pp. 774–782, May 2010.

[21] D. T. H. Tan, P. C. Sun, and Y. Fainman, "Monolithic nonlinear pulse compressor on a silicon chip," *Nature Communications*, vol. 1, p. 116, Nov. 2010.

[22] N. M. L. B. J. Eggleton, G. Lenz, "Optical Pulse Compression Schemes That Use Nonlinear Bragg Gratings," *Fiber and Integrated Optics*, vol. 19, no. 4, pp. 383–421, Oct. 2000. [Online]. Available: https://doi.org/10.1080/014680300300001725

[23] R. Helan, "Comparison of methods for fiber Bragg gratings simulation," ser. International Spring Seminar on Electronics Technology ISSE. IEEE, 2006, pp. 175+.

[24] D. F. G. Gallagher and T. P. Felici, "Eigenmode expansion methods for simulation of optical propagation in photonics - Pros and cons," in *Integrated Optics: Devices, Materials, and Technologies Vii*, Y. S. Sidorin and A. Tervonen, Eds. Bellingham: Spie-Int Soc Optical Engineering, 2003, vol. 4987, pp. 69–82.

[25] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. K. Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, Jan. 2012.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[27] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *arXiv:1506.02142 [cs, stat]*, Jun. 2015.

[28] S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis," *Optics Express*, vol. 8, no. 3, pp. 173–190, Jan. 2001.

[29] "Keras." [Online]. Available: https://keras.io/

[30] TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. [Online]. Available: http://tensorflow.org/

List of Figures

1 The process overview describing the new design methodology. First, datasets are generated using traditional numerical methods (described in Methods). From this dataset, a neural network is trained to characterize the device under consideration. Figures 2 & 2 illustrate this process for a strip waveguide and a chirped grating. Often, the designer iterates between these two steps until an appropriate model is developed. Once the model is ready, several design applications, like circuit simulations, are available. The designs are then fabricated the validate the model's results. From here, the model can be shared and extended. 9

2 Waveguide neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the residual errors after training (c). Panel (d) compares the computational cost for the ANN and the eigenmode solver. Panel (e) exhibits the effective index profiles as a function of a waveguide geometry at 1550 nm for the first TE and TM modes. 10

3 Bragg neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the absolute error after training (c). (d) illustrates the the different adjustable grating parameters and (e) demonstrates fabricated results compared to their corresponding ANN prediction. 11

4 Graphical user interface used to explore the design space of a chirped grating. The slider bars on the left control physical parameters like grating length (NG), grating corrugation (dw), and the grating chirp (a1 and a2). Any time the user adjusts these parameters, the program calls the ANN and reproduces the expected reflection and group delay profiles for that particular grating. Due to the ANN's speed, the program is extremely responsive. 12

5 The design of a monolithic temporal pulse compressor using a silicon photonic chirped Bragg grating (a). A truncated Newton algorithm was tasked with constructing a grating that compressed an arbitrary chirped pulse by a factor of 2. After 400 grating simulations, the optimizer sufficiently minimized a cost function (b) that compared the new pulse's width to the old pulse. The resulting grating is demonstrated in (c) and the input, output, and desired pulses for iterations 10, 100, and 400 are demonstrated in (d-f) respectively. 13

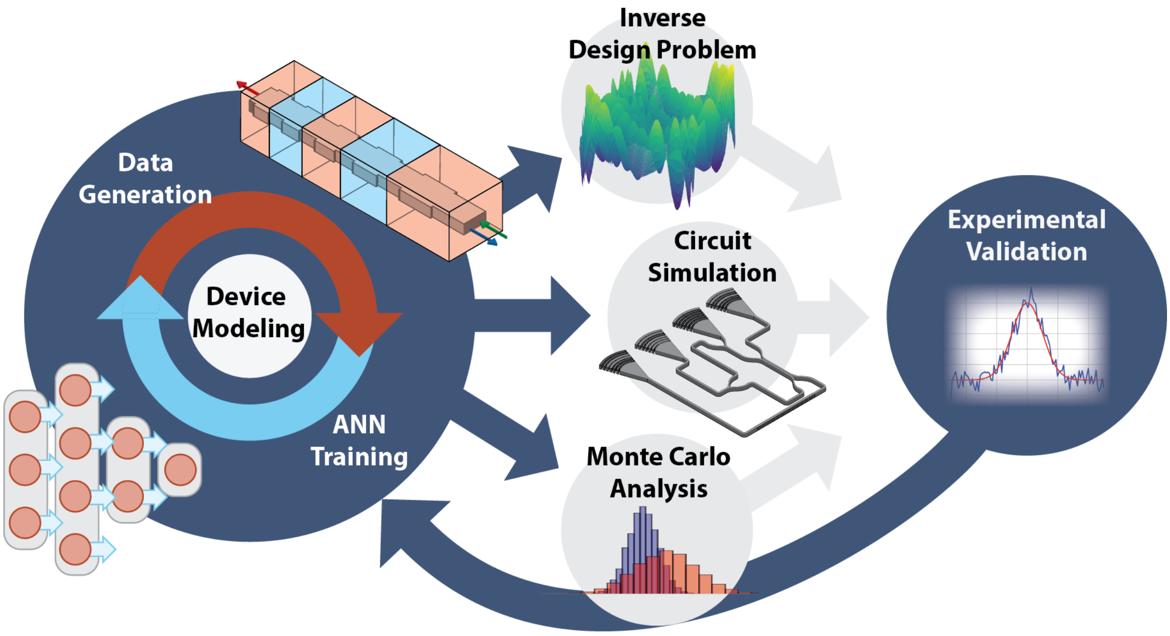

Figure 1: The process overview describing the new design methodology. First, datasets are generated using traditional numerical methods (described in Methods). From this dataset, a neural network is trained to characterize the device under consideration. Figures 2 & 2 illustrate this process for a strip waveguide and a chirped grating. Often, the designer iterates between these two steps until an appropriate model is developed. Once the model is ready, several design applications, like circuit simulations, are available. The designs are then fabricated the validate the model's results. From here, the model can be shared and extended.

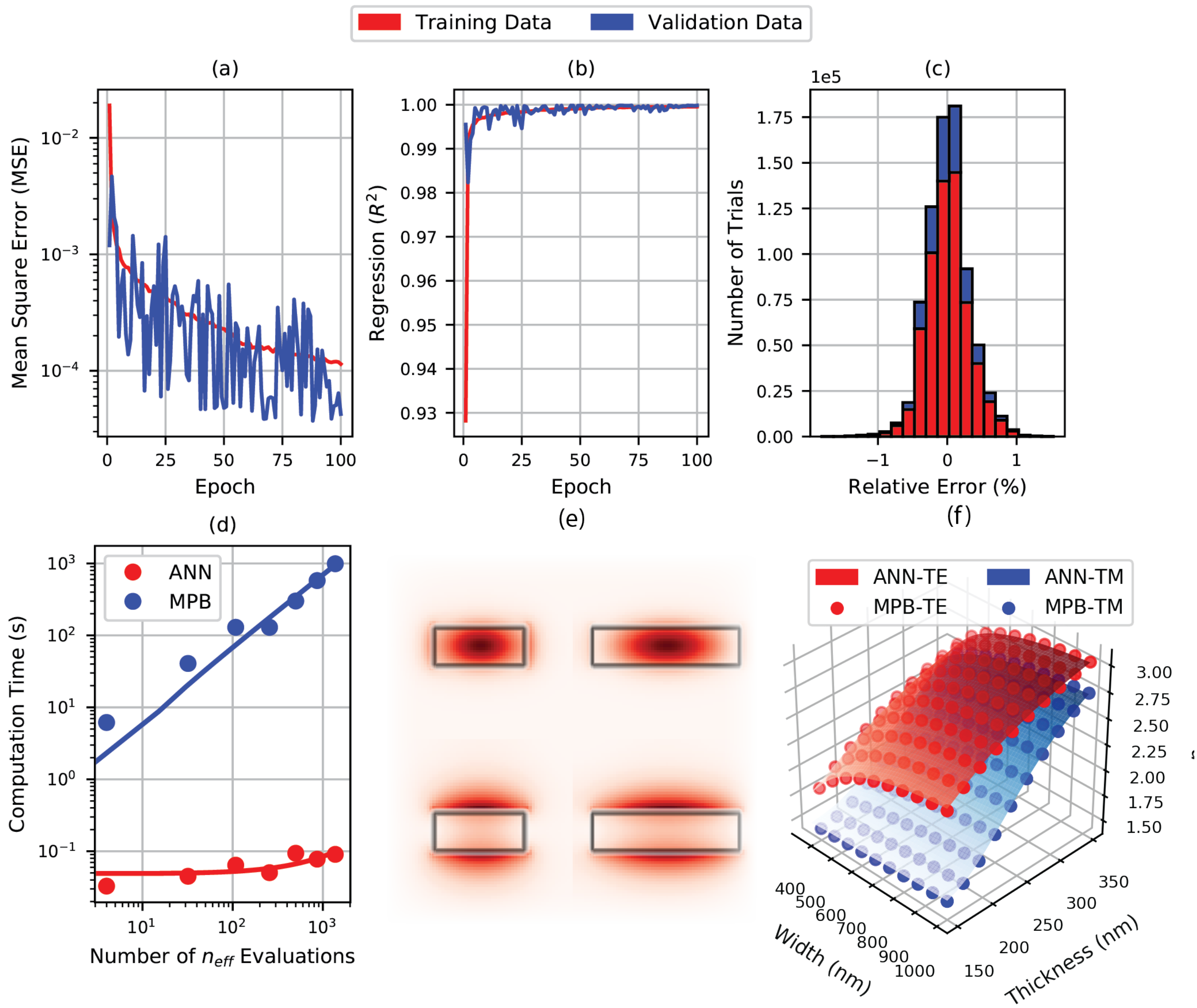

Figure 2: Waveguide neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the residual errors after training (c). Panel (d) compares the computational cost for the ANN and the eigenmode solver. Panel (e) exhibits the effective index profiles as a function of a waveguide geometry at 1550 nm for the first TE and TM modes.

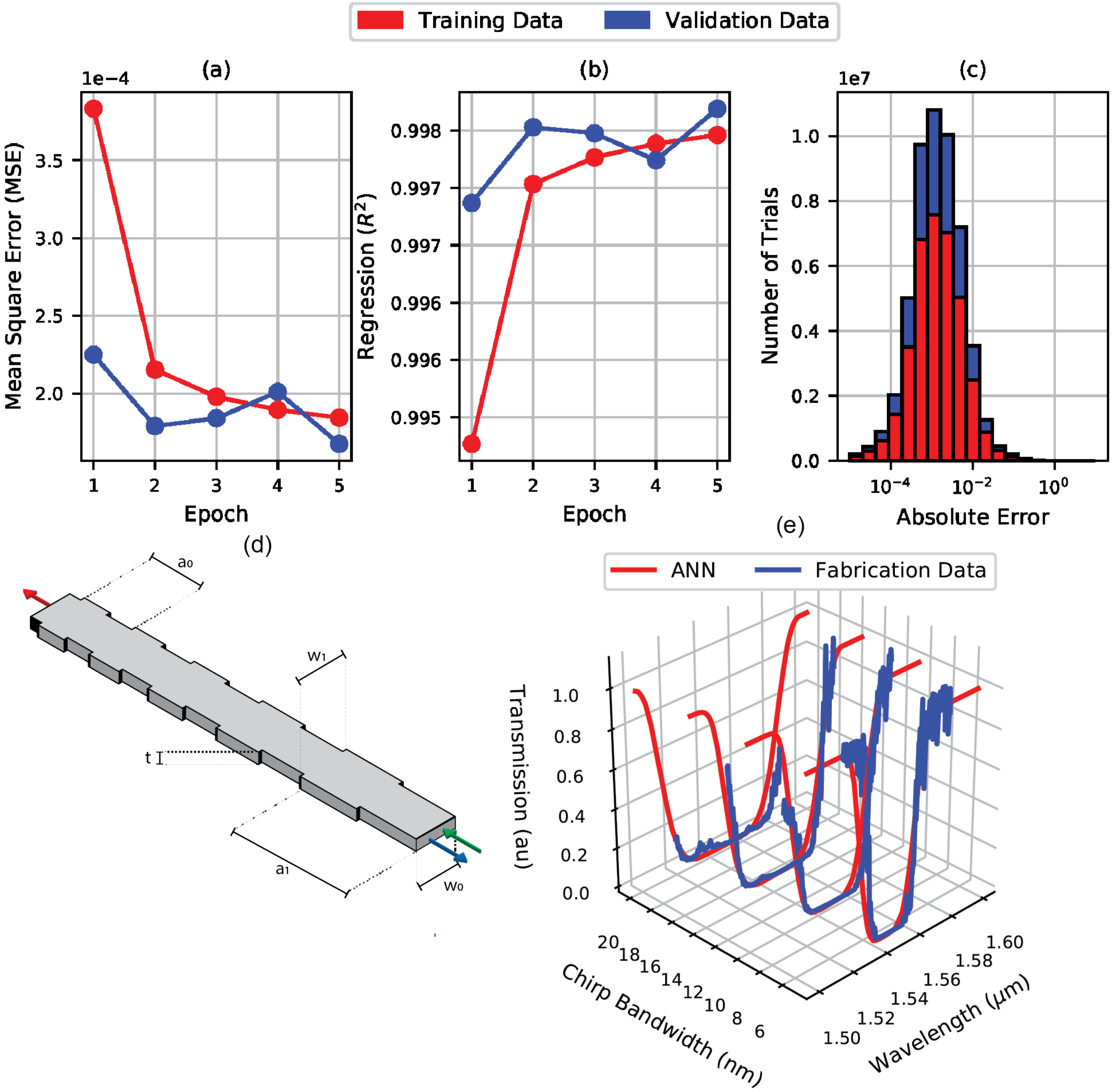

Figure 3: Bragg neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the absolute error after training (c). (d) illustrates the the different adjustable grating parameters and (e) demonstrates fabricated results compared to their corresponding ANN prediction.

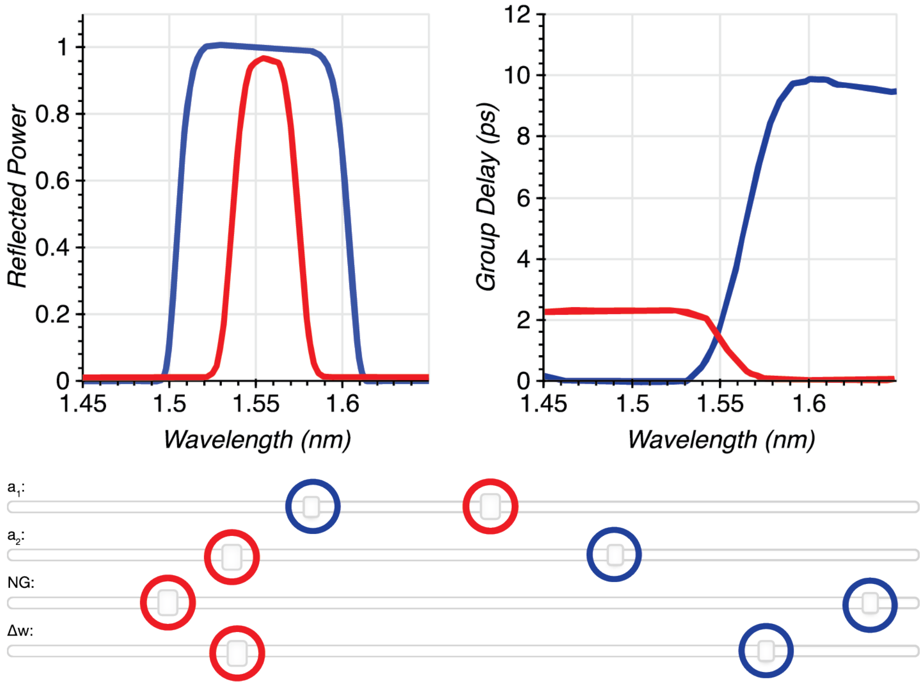

Figure 4: Graphical user interface used to explore the design space of a chirped grating. The slider bars on the left control physical parameters like grating length (NG), grating corrugation (dw), and the grating chirp (a1 and a2). Any time the user adjusts these parameters, the program calls the ANN and reproduces the expected reflection and group delay profiles for that particular grating. Due to the ANN's speed, the program is extremely responsive.

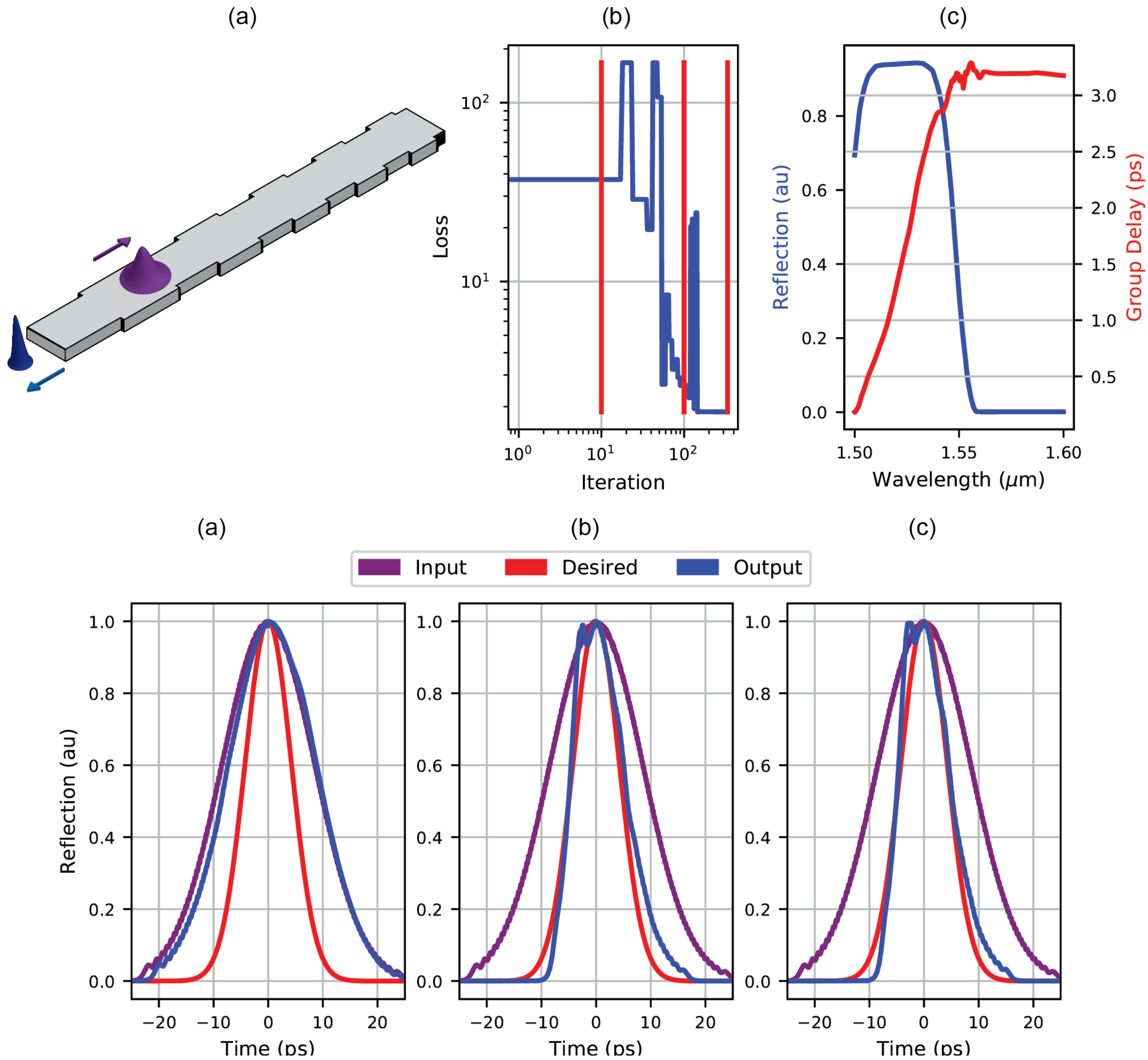

Figure 5: The design of a monolithic temporal pulse compressor using a silicon photonic chirped Bragg grating (a). A truncated Newton algorithm was tasked with constructing a grating that compressed an arbitrary chirped pulse by a factor of 2. After 400 grating simulations, the optimizer sufficiently minimized a cost function (b) that compared the new pulse's width to the old pulse. The resulting grating is demonstrated in (c) and the input, output, and desired pulses for iterations 10, 100, and 400 are demonstrated in (d-f) respectively.