Research Article Vol. 27, No. 21 / 14 October 2019 / Optics Express 29620

Optics EXPRESS

Designing integrated photonic devices using artificial neural networks

ALEC M. HAMMOND AND RYAN M. CAMACHO*

Department of Electrical & Computer Engineering, Brigham Young University, Provo, UT 84602, USA
**camacho@byu.edu*

Abstract: We develop and experimentally validate a practical artificial neural network (ANN) design framework for devices that can be used as building blocks in integrated photonic circuits. As case studies, we train ANNs to model both strip waveguides and chirped Bragg gratings using a small number of simple input and output parameters relevant to designers of integrated photonic circuits. Once trained, the ANNs decrease the computational cost relative to traditional design methodologies by more than 4 orders of magnitude. To illustrate the power of our new design paradigm, we develop and demonstrate both forward and inverse design tools enabled by the ANN. We use these tools to design and fabricate several integrated Bragg grating devices within a useful photonic circuit. The ANN's predictions match the experimental measurements well and do not require any post-fabrication training adjustments.

1. Introduction

Integrated photonics technologies have become powerful tools for processing classical and quantum information. In the era of big data, integrated photonics is increasingly used for optical interconnects [1], on-chip signalling [2], and on-chip photonic processing [3]. Processing quantum information using integrated photonics also shows great promise [4,5], with known advantages in applications ranging from long-distance quantum communications [6] to on-chip quantum simulation [7,8].

Designing integrated photonics components and circuits, however, remains a major bottleneck [9]. Current design flows are complicated by computational tractability and the need for researchers with extensive experience [10]. Unlike their electronic counterparts, photonic integrated circuits require computationally expensive simulation routines to accurately predict their optical response functions. 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 integrated photonics that leverages traditional artificial neural networks (ANN) in an intuitive way to address the specific needs of photonic circuit designers. While several researchers have reported on using ANN's to design individual nanophotonic devices, to our knowledge generating machine learning models for devices that can be directly inserted into integrated photonic circuits has not yet been pursued. The models and devices presented here are available immediately for direct integration with real-world integrated photonic circuits.

Our approach provides several benefits to photonic circuit designers, such as rapid prototyping, inverse design, and direct integration with existing design flows. We demonstrate practical confidence in our method's accuracy by fabricating and measuring devices that experimentally validate the ANN's predictions. As illustrative examples, we design, fabricate, and test chirped integrated Bragg gratings within a useful integrated photonic circuit. Their large parameter spaces and nonlinear responses are typical for devices that are computationally prohibitive using other techniques. The experimental results show remarkable agreement with the ANN's predictions, and to the authors' knowledge represent the first experimental validation ever of any photonic devices designed using ANN's.

#375599 https://doi.org/10.1364/OE.27.029620
Journal © 2019 Received 16 Aug 2019; revised 19 Sep 2019; accepted 19 Sep 2019; published 30 Sep 2019

In designing an ANN freamework, we have made several design choices not typical of previous work:

1. The ANN has a small number of inputs that are physically intuitive and under the direct control of the designer and eventual fabrication and testing engineers (e.g. device geometry, wavelength)
2. The ANN has outputs that are necessary to cascade devices and build photonic circuits, such as transmission and reflection amplitudes, phases and/or group delays commonly used to generate s-parameters.
3. The wavelength variable is introduced to the ANN as an input parameter rather than an output.
4. The wavelength variable is continuous (not discrete) so as to naturally interpolate among wavelength points not in the training set.

Our design framework builds on several theoretical results published in the last several months showing that it is possible to model nanophotonic structures using ANNs. Building on the early work Andrawis et al. [11] to model the effective index of plasmonic waveguides, Ferreira et al. [12] and Tahersima et al. [13] appear to be the first to show that ANNs can assist with the numerical optimization of guided wave structures such as integrated photonic couplers and splitters respectively. 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. In addition, each wavelength required its own output neuron, making it difficult to interpolate between non-sampled wavelength points without additional computational routines, a feature of great practical importance for designers.

As early as 2012 it was shown that ANN's could be successfully employed to model a single unit cell of a 2D periodic structures [14]. More recently, Inampudi et al. [15] and Z. Liu et al. [16] optimized the out-of-plane diffraction patterns from a periodic gratings. Similar to the first guided-wave structures referenced above, their input parameter space consisted of a 2D array of grid points. de Silva Ferreira et al. [17] instead used simple geometric parameters as input neurons to model a unit cell of a periodic structure. Their work appears to be the first ANN to allow for a continuous variable wavelength output, rather than requiring an output neuron for each wavelength point. Nonetheless, their ANN cannot be applied to more complicated non-periodic structures whose phase and amplitude response functions must be obtained for constructing building blocks in an integrated photonic circuit.

Other researchers have focused on overcoming known limitations of ANN's for photonics applications. D. Liu et al. [18] address the problem of degeneracies among solutions found by ANN's for inverse design problems using simple 1D and 3D structures, and propose a bidirectional forward and inverse modelling. Likewise, Ma et al [19] employ bidirectional neural networks to design chiral materials.

Two theoretical papers by Zhang et al [20], and Peurifoy et al. [21] use ANNs to calculate more complicated spectra using a small number of intuitive, smoothly varying geometric input parameters. Zhang et al.'s results are most similar to this work in that they design guided-wave structures using ANN's. Nonetheless, they do not experimentally validate their results, use a large number of discrete wavelength outputs instead of a continuous wavelength input, and only model transmission and not reflection or phase outputs required for a real-world photonic circuit building blocks.

Perhaps most relevant to this work, two recent papers by Gabr et al. [22] and Gostimirovic et al. [23] leverage ANNsto simulate specific guided wave components like waveguides, directional couplers, and polarization-insensitive subwavelength grating (SWG) couplers. We note that only the simple waveguide model in [22] uses wavelength as an input or output to the ANN. Models that train across a broad wavelength spectrum are essential to designing most useful experimental circuit elements and also add non-trivial complexities to the ANN training. Furthermore, our work also includes experimental results and data for the device behavior in complex multi-element photonic circuits, rather than just a theoretical analysis of single-element components. Our work builds upon our previous efforts [24] and enables several new applications, including parameter extraction [25].

To illustrate the power of our new design paradigm, we demonstrate both forward and inverse design tools that use a chirped Bragg grating ANN as a computational backend. To demonstrate ease of use, the forward design tool is interactive, and was used to design our fabricated circuits. The inverse design tool is then used to quickly construct a temporal pulse compressing chirped Bragg grating within specified design constraints — a task typically too computationally expensive for traditional methods.

2. Results

2.1. Overview

To motivate our new 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. These advantages include 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 [26], compensate for semiconductor laser dispersion [27,28], and enable nonlinear temporal pulse compression [29,30].

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.

2.2. 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. 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. The ANN's smooth output also produces smooth analytic derivatives, which are essential for calculating group index profiles and for gradient-based optimization routines. The detailed model is described in Section 4.2.

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

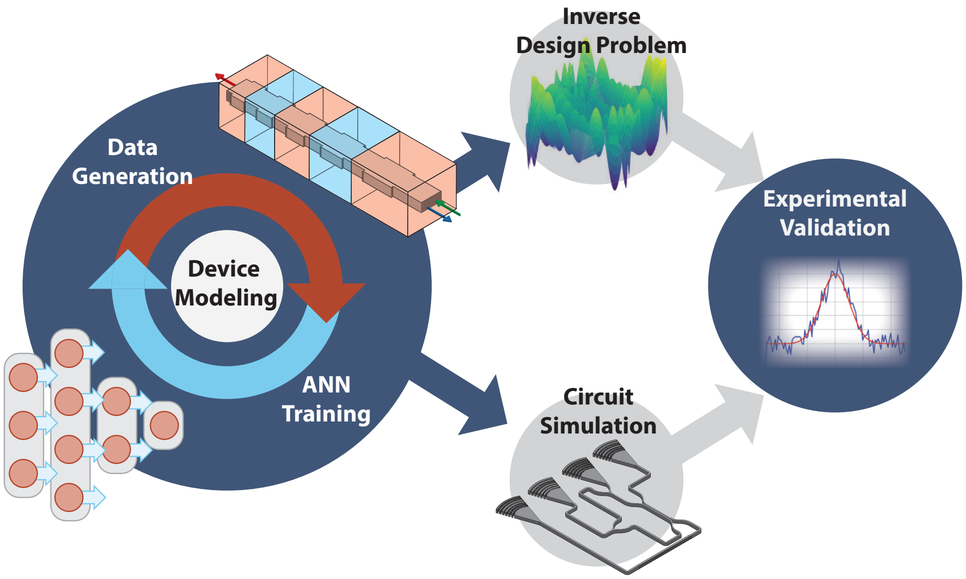

Fig. 1. The process overview describing the new design methodology. First, datasets are generated using traditional numerical methods (described in Methods). From this dataset, a neural network is trained to characterize the device under consideration. Figures 2 & 3 illustrate this process for a strip waveguide and a chirped grating respectively. 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 and inverse design solutions, are available. The designs are then fabricated to validate the model's results. From here, the model can be shared and extended.

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 Fig. 2(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 are 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, MIT Photonic Bands (MPB) [31]. This significant speedup enables many simulation techniques, like the layered dielectric media transfer matrix method (LDMTMM) [32] or the eigenmode expansion method (EMM) [33], 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.

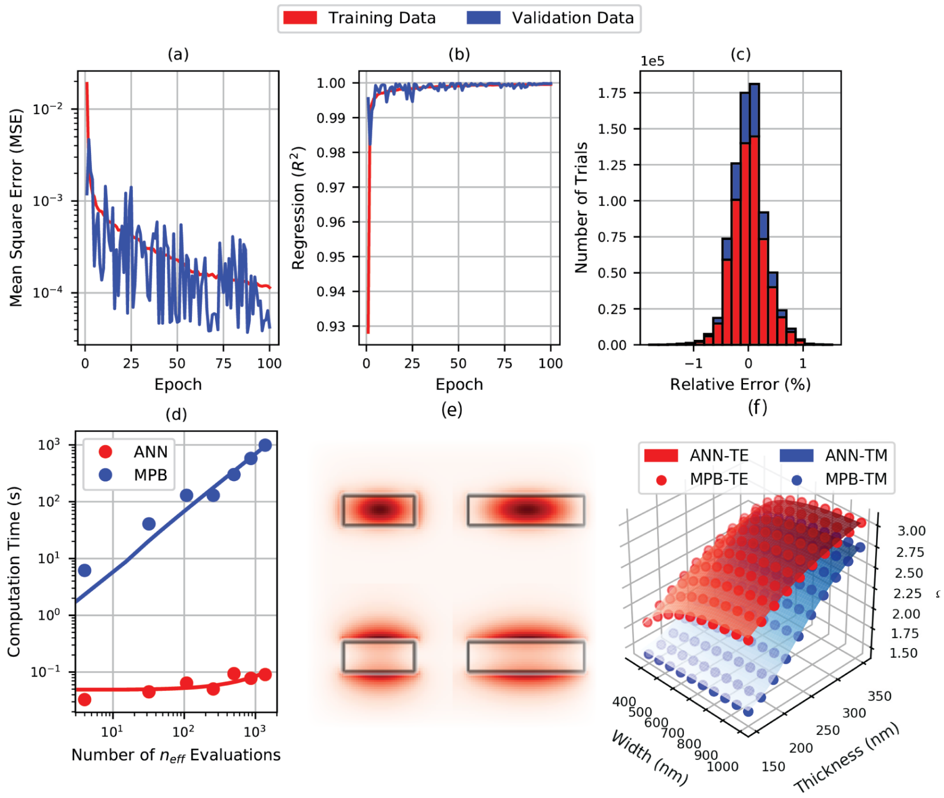

Fig. 2. Waveguide artificial neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the residual errors after training (c). Panel (d) compares the computational cost for the ANN and the eigenmode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

2.3. Bragg grating neural network

Unlike the waveguide model, 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 Fig. 3(d). We note that generating spectra for each device was approximately 2 orders of magnitude faster using the waveguide ANN reported above rather than traditional methods. To smooth apodization dependent ringing, we pre-processed the training data. More information regarding this step is provided in Appendix A. We parameterized the gratings by length of the first grating period (a_0), length of the last grating period (a_1), number of grating periods NG, and grating corrugation width difference($\Delta w = w_1 - w_0$). We designed the network to receive these four parameters along with a single wavelength point as inputs. The network has two outputs: reflected optical power and group delay.

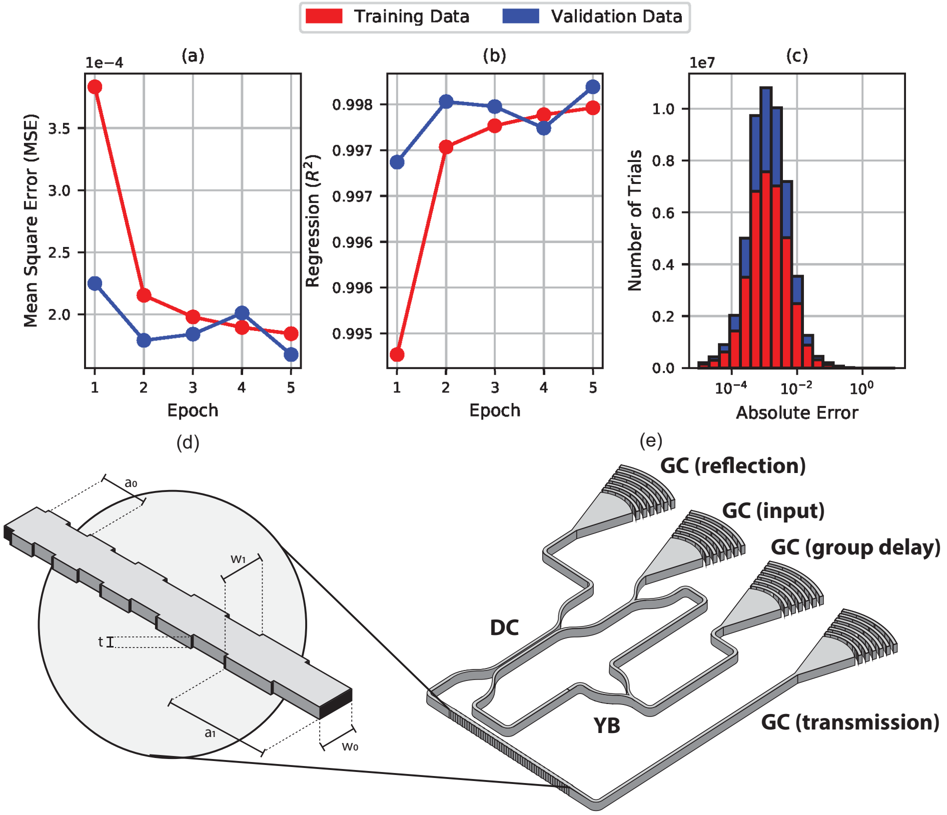

Fig. 3. Bragg grating artificial neural network training results demonstrated by the training convergence with reference to the mean square error (a), the coefficient of determination (b), and the absolute error after training (c). (d) illustrates the different adjustable grating parameters and (e) illustrates the interrogation circuit used to extract the reflection, transmission, and group delay profiles simultaneously from the chirped Bragg grating. A grating coupler (GC) feeds light into various Y-branches (YB) and directional couplers (DC) such that the transmission and reflection spectra can both be extracted from the chirped Bragg grating (BG). Half of the reflected signal is sent through a Mach-Zehner Interferometer (MZI). The output of which is used to extract the group delay.

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 epoch (after several hundred batch iterations) and we stopped training after just five epochs to prevent overfitting. The final MSE for the training and validation sets was 1.845×10^{-4} and 1.677×10^{-4} 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. Figures 3(a)–3(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 calculating 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.

To validate the Bragg ANN, we fabricated and measured several silicon photonic Bragg gratings with different chirping patterns and compared their transmission, reflection, and group delay spectra to the neural network's predictions. The gratings were arranged in one of two configurations: (1) a simple circuit that only measured the Bragg grating's transmission spectra and (2) a more complicated interrogation circuit capable of measuring the reflection, transmission, and group delay profiles from the same device simultaneously. Figure 3 illustrates the interrogation circuit used to measure all three responses simultaneously. In both configurations, grating couplers were used to route light on and off the chip. While the simpler circuit required less de-embedding, the full interrogator circuit allowed for a more comprehensive device characterization.

The transmission-only gratings were designed with various grating period bandwidths from 5 nm to 20 nm, each with 600 periods and a corrugation width of 50 nm. The initial design parameters produced ANN predictions that match the measured data extremely well. Small discrepancies in the grating responses are largely attributed to the grating's apodization profile and detector noise. Figure 4 illustrates the comparison between the ANN's predictions and the measured data.

We designed the remaining gratings using a much smaller chirp bandwidth of 3 nm with 750 grating periods and a 30 nm corrugation width. We mirrored the orientation of half the gratings in order to measure both positive and negative sloped group delay profiles. Once measured, we normalized the data by de-embedding the responses from the various Y-branches, directional couplers, and grating couplers that complicate the measurement data. The process is explained in Appendix B. Even with the rather complex transfer function, the transmission, reflection, and group delay profiles match the ANN's corresponding predictions well except for occasional resonant features caused by fabrication defects. These defects are expected since the narrow bandwidth devices have a grating pitch with a fine discretization that approaches the e-beam raster grid resolution. Small changes in grating pitch that don't align with the raster grid occasionally produce weak Fabry-Perot resonance conditions visible in the data. These raster-induced defects also account for a small lateral shift (˜1 nm) in the responses. Even with these fabrication challenges, it is notable that the ANN successfully predicts the transmission, reflection, and group delay profiles simultaneously. In fact, the ability to do so in noisy fabrication environments is one of the key advantages of the ANN and may allow for efficient parameter extraction where other methods fail.

2.4. Forward design

The neural network's speed and flexibility enable forward design exploration. For example, Fig. 5 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 even novice designers the ability to rapidly gain device intuition without necessarily understanding the underlying numerical techniques.

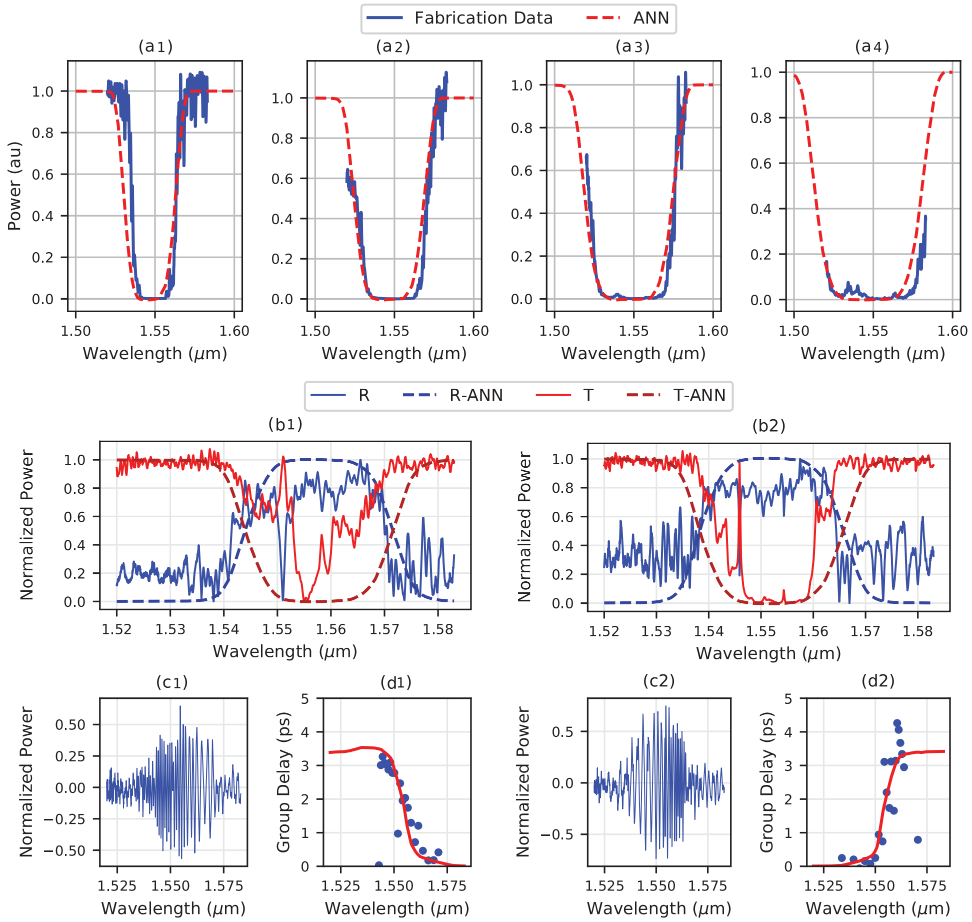

Fig. 4. Fabrication data compared to corresponding ANN predictions. (a1)-(a4) Measured transmission responses for gratings with a period chirp of 5 nm (a1), 10 nm (a2), 15 nm (a3), and 20 nm (a4). (b1)-(b2) Transmission and reflection responses for two different Bragg gratings. Both gratings share the same design parameters, and have an identical but opposite linear chirp. The result of the mirrored chirping is seen in both the normalized MZI interference patterns (c1) and (c2) and the extracted group delay responses (d1), and (c2).

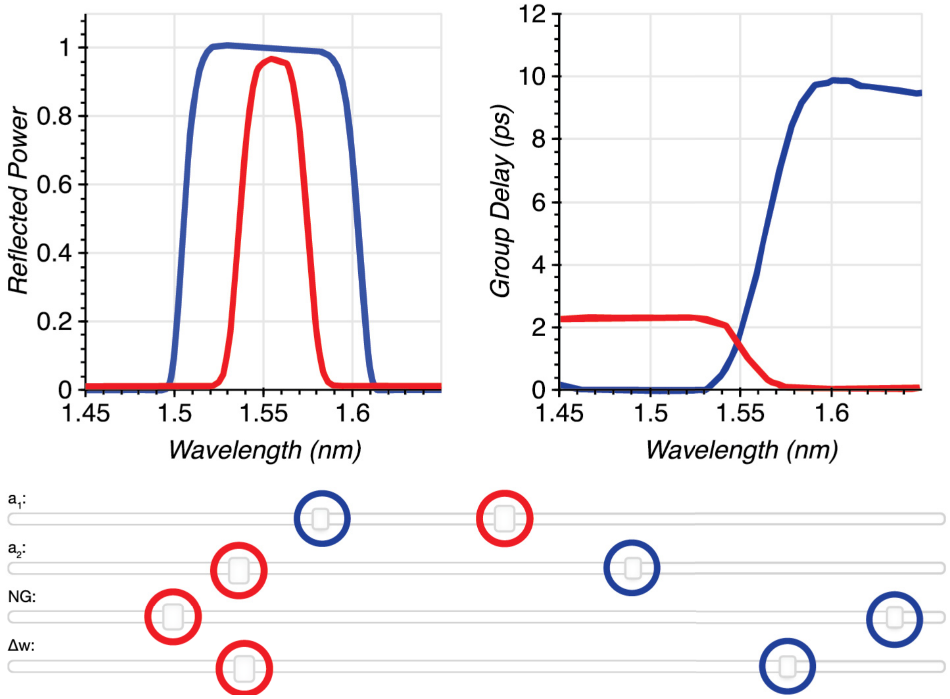

Fig. 5. Graphical user interface used to explore the design space of a chirped Bragg grating. The slider bars on the left control physical parameters like grating length (NG), grating corrugation (dw), and the grating chirp (a1) and (a2). Any time the user adjusts these parameters, the program calls 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.

2.5. 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 [29]. In this particular case study, we assumed an arbitrary source generates a 20 ps wide chirped pulse with a 4 nm bandwidth. Figure 6 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, owing to the accelerated cost function. The agnostic nature of the neural network interface works well with a variety of 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.

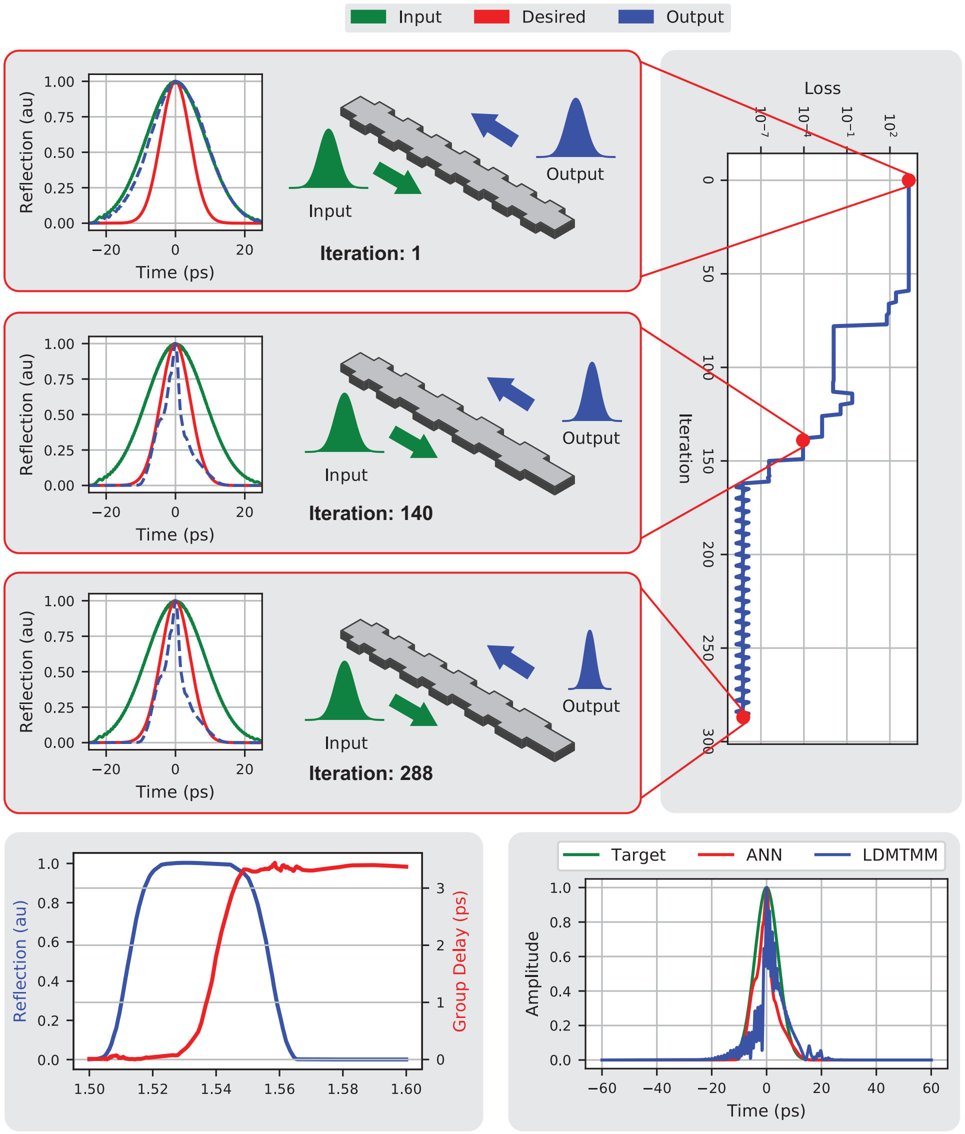

Fig. 6. ANN-assisted design of a monolithic temporal pulse compressor using a silicon photonic chirped Bragg grating. A truncated Newton algorithm was tasked with constructing a grating that compressed an arbitrary chirped pulse by a factor of 2. After 340 grating simulations, the optimizer sufficiently minimized a cost function (right) that compared the new pulse's width to the old pulse. The resulting grating is demonstrated in the bottom left panel and the input, output, and desired pulses for iterations 1, 140, and 288 are demonstrated on the left. The final compressed pulse predicted by the ANN (red) is compared to the target pulse (green) and the result simulated using the LDMTMM method (blue) in the lower right panel. The results show that the ANN accurately predicts the pulse width.

3. Discussion

Our method demonstrates a new, viable platform for silicon photonic circuit design. With a single global parameter fit, we successfully modeled silicon photonic waveguides and silicon photonic chirped Bragg gratings with arbitrary bandwidths, chirping patterns, lengths, and corrugation widths and allowed arbitrary wavelength sampling. Future work could explore new network architectures (e.g. different activation functions, layer connections, etc.) and training algorithms. Several other devices, like ring resonators [34], 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 as done in all previous work known to the authors. 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 and experimentalists. 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 personal computers. With the growing availability of vast cloud-based computational resources, however, several million training simulations can now be run in hours or days. [35]. 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 [36]. 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.

As with all deep learning applications, the network's utility is limited by biases introduced in the training set, the network architecture, or even the training process itself [37]. 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) [38]. 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

4.1. Training data generation and preprocessing

We generated our waveguide neural network's training set on a high performance computing cluster (HPC) using MIT Photonic Bands (MPB) [31], 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) [32]. 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 fit the results to a generalized skewed Gaussian (see Appendix B) to reduce ringing and to generalize the grating's response to arbitrary apodization profiles. We found that without 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 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.

4.2. Neural network design and training

Both neural networks were trained on the same HPC cluster using Keras [39] and Tensorflow [40]. 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. No dropout was 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.

4.3. Simulation benchmarks

We performed all benchmarks using a quad-core Intel(R) i5-2400 CPU clocked at 3.10 GHz 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.

4.4. 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.

4.5. 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.

A. Training data processing

Artificial neural networks model the relationship between their inputs and outputs by cascading various computational units, known as neurons [41]. 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 [42]. If the network architecture is sufficiently large, any arbitrary relationship manifested in the dataset can be modeled [43]. Capturing complex behavior, like parameterized sinusoidal ringing, requires sufficient sampling.

Bragg grating device responses are heavily influenced by their apodization [44]. For example, devices without apodization have much more ringing than devices with a raised cosine apodization. The effect is similar to finite impulse response (FIR) filtering windows [45]. Figure 7 illustrates various apodization windows and the resultant ringing. While the layered dielectric medium transfer matrix method (LDMTMM) can effectively simulate such ringing, fabrication inconsistencies make predicting the actual ringing close to impossible.

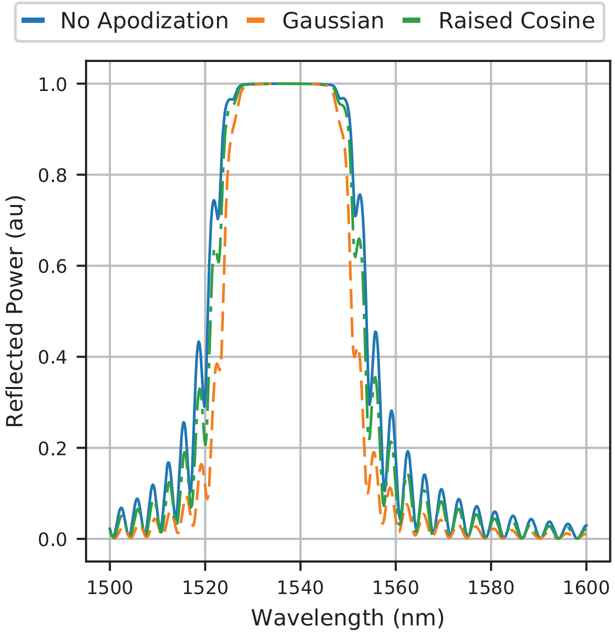

Fig. 7. Reflection profile for an integrated chirped Bragg grating with no apodization (blue), a Gaussian apodization (orange), and a raised cosine apodization. Different apodization functions reduce the response's ringing.

Consequently, designers focus more effort on shaping the Bragg grating's reflection, transmission, and group delay responses than mitigating ringing. This mentality encourages a design tool that temporarily ignores any ringing and simulates the basic grating response profiles.

To accomplish this, we filtered all apodization dependent ringing from the Bragg ANN's training set by fitting the reflection spectra and group delay responses. Through trial and error, we found that a generalized, skewed Gaussian of the form

$$f(\lambda, \lambda_0, \sigma, \beta,\ a,\ p,\ c) = \frac{a\sigma}{\gamma} e^{\frac{-\beta|\lambda-\lambda_0|^p}{\gamma}} + c \tag{1}$$

where

$$\gamma = \frac{2\sigma}{1 + e^{-\beta(\lambda-\lambda_0)}} \tag{2}$$

most comprehensively models both the reflection and group delay profiles for a wide array of tuning parameters. Each parameter strategically tunes a particular feature commonly found in both responses. For example, a simply scales the maximum reflectivity or maximum group delay, σ shapes the reflection bandwidth or group delay spread, β induces skewness to one side or the other, p flattens the function's main lobe (i.e. for saturated reflection profiles), and c provides the

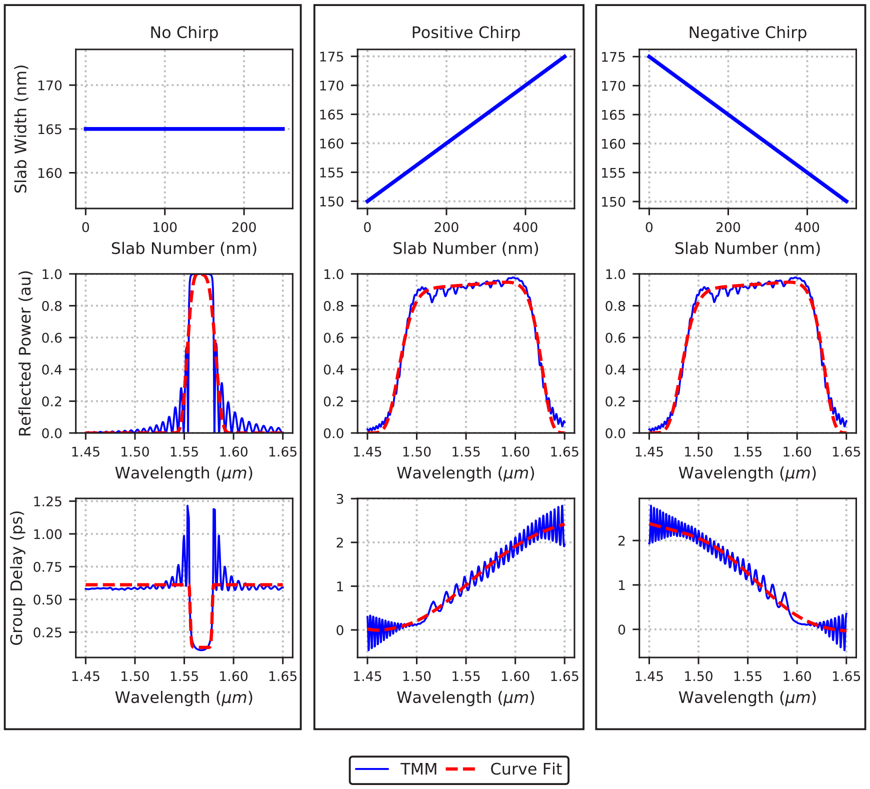

Fig. 8. Demonstration of the fitting algorithm for a Bragg grating with no chirp (column one), a postive chirp (column two) and a negative chirp (column three). The chirp patterns themselves are illustrated in the first row, the reflection profiles are depicted in the second row, and the respective group delay responses are found in the third row. The modified Gaussian function accounts for the wider bandwidths, skewness, and overall shape of the responses will "filter" out the apodization dependent ringing.

necessary offsets for the group delay profiles. Figure 8 illustrates various LDMTMM simulations and the corresponding fit.

To find the optimal coefficients, we first ran a differential evolution algorithm that minimized the mean squared error between the LDMTMM simulation and the functional fit [46]. This global optimization routine provided a suitable starting point for a Levenberg-Marquardt algorithm to locate the local minima [47]. Using a two-staged optimization approach increased the robustness of the training processing and enabled batched parallelization on the high performance cluster (HPC).

B. Measurement data normalization

The circuit used to simultaneously extract the transmission, reflection, and group delay profiles of the integrated Bragg gratings incorporated several additional devices that skewed the Bragg grating's transfer function. The grating couplers, Y-branches, and directional couplers used to route the signal all have non uniform frequency responses, and must be normalized out in order to accurately measure the Bragg grating's response.

Many designers fabricate de-embedded devices where each individual transfer function can be extracted and subsequently eliminated from the larger circuit [9]. Fabrication variability, however, prevents consistency from one device to another. The circuit's grating couplers, for example, sometimes showed 3 dB of variation across the band from one de-embedded structure to another. Consequently, this method cannot be reliably used to calibrate the Bragg grating responses.

Instead, we fit the signal outside of the Bragg grating's reflection/transmission band using a fifth order polynomial that much more accurately describes the other devices' spectral influences. Figure 9 illustrates this procedure. Since the Bragg grating's group delay and magnitude responses are band-limited, we know what the rest of the spectrum should look like.

To extract the group delay from the interference pattern, we first normalize out the low frequency carrier by fitting the entire signal to a fifth degree polynomial. Then, we estimate the free spectral range (FSR) by tracking each oscillation peak. Using the transfer function of the MZI [9] along with the FSR, the resulting group index and group delay is estimated. Figure 10 illustrates this process.

Optics EXPRESS

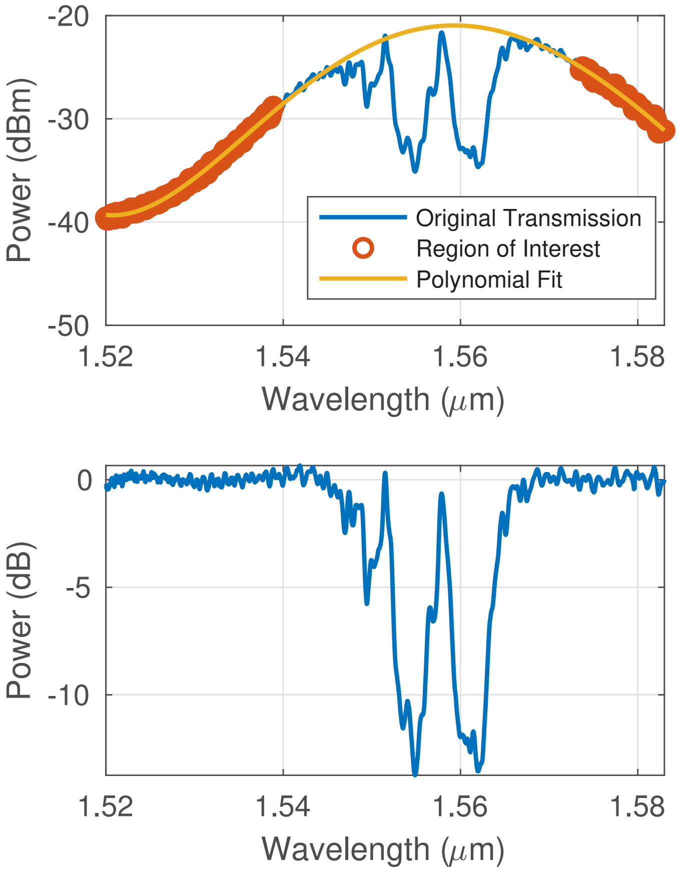

Fig. 9. The normalization process for the integrated Bragg gratings. Data points outside of the stop band are fitted to a fifth degree polynomial to capture the response of the grating couplers and other devices within the circuit (top). The polynomial is then normalized from the data (bottom).

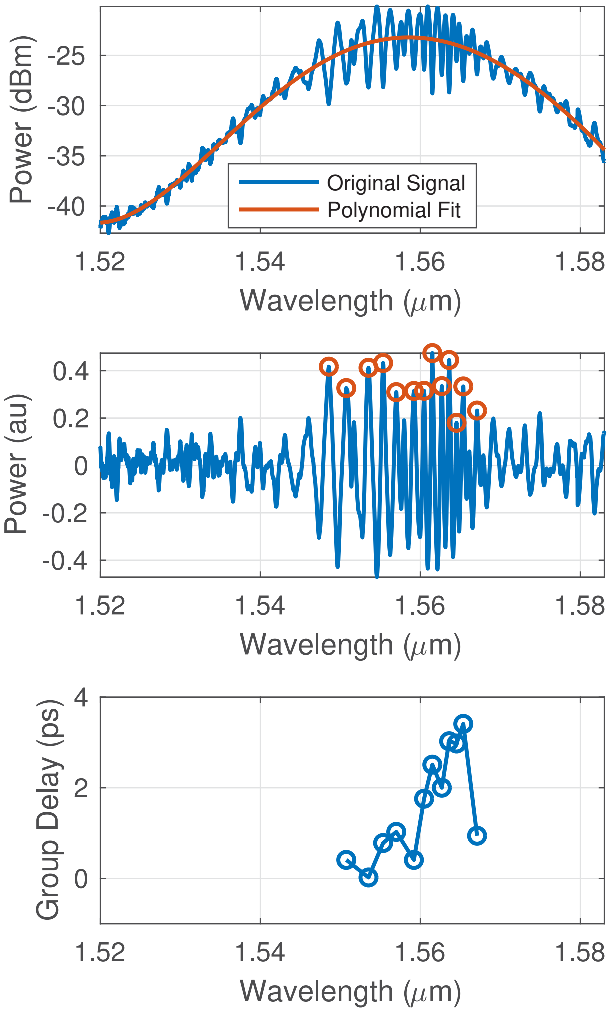

Fig. 10. The group delay extraction process. First, the low frequency carrier is removed by using a fifth order polynomial fit (top). Next, the oscillation peaks are tracked and the FSR is estimated (middle). Finally, the group delay is calculated using the transfer function of the MZI (bottom).

Acknowledgments

The authors thank Lukas Chrostowski for useful discussions relating to the Bragg structures and for facilitating the SiEPIC fabricating process, as well as David Buck for supplying additional fabrication data.

References

1. M. J. Heck, H.-W. Chen, A. W. Fang, B. R. Koch, D. Liang, H. Park, M. N. Sysak, and J. E. Bowers, "Hybrid silicon photonics for optical interconnects," IEEE J. Sel. Top. Quantum Electron. **17**(2), 333–346 (2011).
2. T. Barwicz, H. Byun, F. Gan, C. W. Holzwarth, M. A. Popovic, P. T. Rakich, M. R. Watts, E. P. Ippen, F. X. Kärtner, H. I. Smith, J. S. Orcutt, R. J. Ram, V. Stojanovic, O. O. Olubuyide, J. L. Hoyt, S. Spector, M. Geis, M. Grein, T. Lyszczarz, and J. U. Yoon, "Silicon photonics for compact, energy-efficient interconnects (invited)," J. Opt. Netw. **6**(1), 63–73 (2007).
3. J. Wang, "Chip-scale optical interconnects and optical data processing using silicon photonic devices," Photon. Netw. Commun. **31**(2), 353–372 (2016).
4. A. Orieux and E. Diamanti, "Recent advances on integrated quantum communications," J. Opt. **18**(8), 083002 (2016).
5. F. Flamini, N. Spagnolo, and F. Sciarrino, "Photonic quantum information processing: a review," Rep. Prog. Phys. **82**(1), 016001 (2019).
6. D. Bunandar, A. Lentine, C. Lee, H. Cai, C. M. Long, N. Boynton, N. Martinez, C. DeRose, C. Chen, M. Grein, D. Trotter, A. Starbuck, A. Pomerene, S. Hamilton, F. N. C. Wong, R. Camacho, P. Davids, J. Urayama, and D. Englund, "Metropolitan quantum key distribution with silicon photonics," Phys. Rev. X **8**(2), 021009 (2018).
7. N. C. Harris, G. R. Steinbrecher, M. Prabhu, Y. Lahini, J. Mower, D. Bunandar, C. Chen, F. N. C. Wong, T. Baehr-Jones, M. Hochberg, S. Lloyd, and D. Englund, "Quantum transport simulations in a programmable nanophotonic processor; EP," Nat. Photonics **11**(7), 447–452 (2017).
8. X. Qiang, X. Zhou, J. Wang, C. M. Wilkes, T. Loke, S. O'Gara, L. Kling, G. D. Marshall, R. Santagati, T. C. Ralph, J. B. Wang, J. L. O'Brien, M. G. Thompson, and J. C. F. Matthews, "Large-scale silicon quantum photonics implementing arbitrary two-qubit processing," Nat. Photonics **12**(9), 534–539 (2018).
9. L. Chrostowski and M. Hochberg, *Silicon Photonics Design: From Devices to Systems* (Cambridge University, 2015).
10. W. Bogaerts and L. Chrostowski, "Silicon Photonics Circuit Design: Methods, Tools and Challenges," Laser Photonics Rev. **12**(4), 1700237 (2018).
11. R. R. Andrawis, M. A. Swillam, M. A. El-Gamal, and E. A. Soliman, "Artificial neural network modeling of plasmonic transmission lines," Appl. Opt. **55**(10), 2780–2790 (2016).
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," Appl. Soft Comput. **65**, 1–11 (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] (2018).
14. G. N. Malheiros-Silveira and H. E. Hernandez-Figueroa, "Prediction of Dispersion Relation and PBGs in 2-D PCs by Using Artificial Neural Networks," IEEE Photonics Technol. Lett. **24**(20), 1799–1801 (2012).
15. S. Inampudi and H. Mosallaei, "Neural network based design of metagratings," Appl. Phys. Lett. **112**(24), 241102 (2018).
16. Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative Model for the Inverse Design of Metasurfaces," Nano Lett. **18**(10), 6570–6576 (2018).
17. 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," J. Lightwave Technol. **36**(18), 4066–4073 (2018).
18. D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures," ACS Photonics **5**(4), 1365–1369 (2018).
19. W. Ma, F. Cheng, and Y. Liu, "Deep-Learning-Enabled On-Demand Design of Chiral Metamaterials," ACS Nano **12**(6), 6326–6334 (2018).
20. T. Zhang, J. Wang, Q. Liu, J. Zhou, J. Dai, X. Han, Y. Zhou, and K. Xu, "Efficient spectrum prediction and inverse design for plasmonic waveguide systems based on artificial neural networks," Photonics Res. **7**(3), 368–380 (2019).
21. 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," Sci. Adv. **4**(6), eaar4206 (2018).
22. A. M. Gabr, C. Featherston, C. Zhang, C. Bonfil, Q.-J. Zhang, and T. J. Smy, "Design and optimization of optical passive elements using artificial neural networks," J. Opt. Soc. Am. B **36**(4), 999–1007 (2019).
23. D. Gostimirovic and N. Y. Winnie, "An open-source artificial neural network model for polarization-insensitive silicon-on-insulator subwavelength grating couplers," IEEE J. Sel. Top. Quantum Electron. **25**(3), 1–5 (2019).
24. A. M. Hammond and R. M. Camacho, "Designing silicon photonic devices using artificial neural networks," arXiv preprint arXiv:1812.03816 (2018).

25. A. M. Hammond, E. Potokar, and R. M. Camacho, "Accelerating silicon photonic parameter extraction using artificial neural networks," OSA Continuum **2**(6), 1964–1973 (2019).
26. M. Rochette, M. Guy, S. LaRochelle, J. Lauzon, and F. Trepanier, "Gain equalization of EDFA's with Bragg gratings," IEEE Photonics Technol. Lett. **11**(5), 536–538 (1999).
27. D. T. H. Tan, K. Ikeda, R. E. Saperstein, B. Slutsky, and Y. Fainman, "Chip-scale dispersion engineering using chirped vertical gratings," Opt. Lett. **33**(24), 3013–3015 (2008).
28. M. J. Strain and M. Sorel, "Design and Fabrication of Integrated Chirped Bragg Gratings for On-Chip Dispersion Control," IEEE J. Quantum Electron. **46**(5), 774–782 (2010).
29. D. T. H. Tan, P. C. Sun, and Y. Fainman, "Monolithic nonlinear pulse compressor on a silicon chip," Nat. Commun. **1**(1), 116 (2010).
30. N. M. L. B. J. Eggleton and G. Lenz, "Optical Pulse Compression Schemes That Use Nonlinear Bragg Gratings," Fiber Integr. Opt. **19**(4), 383–421 (2000).
31. S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis," Opt. Express **8**(3), 173–190 (2001).
32. R. Helan, "Comparison of methods for fiber bragg gratings simulation," in *2006 29th International Spring Seminar on Electronics Technology*, (IEEE, 2006), pp. 161–166.
33. 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*, vol. 4987 Y. S. Sidorin and A. Tervonen, eds. (Spie-Int Soc Optical Engineering, Bellingham, 2003), pp. 69–82.
34. 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 Rev. **6**(1), 47–73 (2012).
35. C. Vecchiola, S. Pandey, and R. Buyya, *High-Performance Cloud Computing: A View of Scientific Applications* (IEEE, New York, 2009).
36. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010).
37. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," J. Mach. Learn. Res. **15**, 1929–1958 (2014).
38. Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," arXiv:1506.02142 [cs, stat] (2015).
39. F. Chollet, "Keras," https://github.com/fchollet/keras (2015).
40. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems".
41. J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks **61**, 85–117 (2015).
42. Y. le Cun, "A theoretical framework for back-propagation," *Proc. 1988 Connect. Model. Summer Sch. CMU*, Pittsburg, PA pp. 21–28 (1988).
43. K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," Neural Networks **4**(2), 251–257 (1991).
44. K. O. Hill and G. Meltz, "Fiber Bragg grating technology fundamentals and overview," J. Lightwave Technol. **15**(8), 1263–1276 (1997).
45. F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," Proc. IEEE **66**(1), 51–83 (1978).
46. R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," J. Glob. Optim. **11**(4), 341–359 (1997).
47. D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963).