Accelerating Artificial Intelligence with Programmable Silicon Photonics

by

Saumil Bandyopadhyay

S.B., Massachusetts Institute of Technology (2017)
M.Eng., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2023

© 2023 Saumil Bandyopadhyay. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Saumil Bandyopadhyay
Department of Electrical Engineering and Computer Science
May 19, 2023

Certified by: Dirk R. Englund
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Accelerating Artificial Intelligence with Programmable Silicon Photonics

by

Saumil Bandyopadhyay

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2023, in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Abstract

Advances in the fabrication of large-scale integrated silicon photonics have sparked interest in optical systems that process information at high speeds with ultra-low energy consumption. Photonic systems, which have historically been used for optical telecommunications, have recently been demonstrated to accelerate tasks in quantum simulation, artificial intelligence, and combinatorial optimization.

This thesis reports work towards the goal of realizing large-scale programmable photonic systems for information processing: 1) we develop deterministic error correction algorithms for programmable photonic systems, whose capabilities are believed to be limited by fabrication error, showing that these systems can be programmed to implement accurate linear matrix processing suitable for deep neural networks at scales of up to hundreds of channels; 2) we describe a new paradigm for coupling large numbers of optical channels to photonic circuits with exceptionally high alignment tolerance, enabling the use of high-volume, low-precision electronic pick-and-place equipment for photonic assembly; and 3) we design, fabricate, and demonstrate the first single-chip, end-to-end photonic processor for deep neural networks. This fully-integrated coherent optical neural network (FICONN), which monolithically integrates multiple optical processor units for matrix algebra and nonlinear activation functions into a single chip, implements single-shot coherent optical processing of a deep neural network with sub-nanosecond latency. On-chip, *in situ* training of a deep neural network is demonstrated on this system, obtaining high accuracies on a vowel classification task comparable to that of a digital system. Our results open the path towards integrated, large-scale photonic processors for low-latency inference and training of deep neural networks.

Thesis Supervisor: Dirk R. Englund
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

The work in this thesis would not have been possible without the support of an exceptional group of mentors, colleagues, family, and friends.

First, I would like to express my deep gratitude to my doctoral advisor, Professor Dirk Englund, who has mentored me through my time as an undergraduate, master's, and finally doctoral student. Dirk is an energetic and creative researcher who dedicates enormous amounts of time to working with and mentoring his students. A critically important part of my PhD has been my many discussions with Dirk on new research directions, which often sparked many of the ideas considered in this thesis and has greatly influenced my development as a scientist. Dirk provided me with the opportunity and flexibility to explore many new and exciting areas in the field of photonics during my time in graduate school, and I am very grateful for his mentorship and guidance over all of these years.

An equally important part of my doctoral work has been the time I've spent collaborating with the many members of the Quantum Photonics Group. Dr. Alexander Sludds has been a friend and collaborator in the lab for nearly all of my PhD, and I've enjoyed the many hours we've spent racing each other to fiber couple into chips and our long discussions on new ideas in the exciting field of silicon photonics. The work on mesh error correction in this thesis was conducted in close collaboration with Dr. Ryan Hamerly, with whom I've had the pleasure of working with not only on programmable optics, but on many other projects spanning machine learning, integrated photonics, and nonlinear optics. I've also benefitted from many discussions and collaborations with the talented group of researchers working on AI hardware in our lab–in particular, I'd like to thank Liane Bernstein and Dr. Sri Krishna Vadlamani for being great friends and colleagues from whom I've learned a great deal.

The experiments I present in this thesis were enabled by the experience, knowledge, and infrastructure built by past group members working in silicon photonics for both quantum information and artificial intelligence. I would like to thank Dr. Jacques Carolan, who mentored me when I started graduate school and taught me the intricacies of setting up and working with large silicon photonic chips, and Dr. Mihika Prabhu, from whom I learned a lot about photonic chip testing in the early years of my PhD.

An important part of scientific research is being able to communicate one's ideas effectively. I would like to thank Professor Jelena Notaros for serving on my thesis committee and for giving me the opportunity to serve as a teaching assistant for her pilot course on silicon photonics. It is a rare and incredibly exciting opportunity to help develop a new course, especially on a topic so closely linked to one's doctoral research.

I am also incredibly thankful to Dr. Michael Hochberg, who has been a mentor to me since before I started graduate school. Before I started my PhD, Michael offered me the opportunity to work with him and the Photonic Design Team at Elenion Technologies to learn silicon photonics firsthand from the experts in the field, and I returned to graduate school with a new appreciation for photonic system design. I am very grateful to Michael for his mentorship during my PhD and for the many illuminating discussions on my research, particularly on our collaboration on the end-to-end photonic deep neural network processor presented in this thesis. I would also like to thank the other members of the

team at Elenion, who provided my first introduction to the exciting field of silicon photonics and patiently answered my many questions, particularly Dr. Matthew Streshinsky, who has been another mentor and collaborator on the photonic DNN experiments.

Our group is extremely lucky to have excellent administrators who keep our lab running–in particular, I'd like to thank Janice Balzer, David Barnett, and Andrew Birkel for supporting the logistics of my doctoral research. I've also been fortunate to have made many friends in my time in the group–thank you to Dr. Mohamed ElKabbash, Prof. Stefan Krastanov, Dr. Sivan Trajtenberg-Mills, Hugo Larocque, Ian Christen, Sophia Duan, Prof. Carlos Errando Herranz, Dr. Eric Bersin, and Dr. Mikkel Heuck for being excellent friends and colleagues. I have learned a lot from all of you.

Finally, I owe a lot to my parents, Anuradha and Supriyo, who have provided endless support and inspiration and pushed me to achieve more than I ever thought possible without ever doubting that I could. They have been a bedrock of support during my PhD, which has been both the most challenging and the most rewarding of my academic endeavors, and have worked tirelessly to provide me with the best opportunities and ensure I will always be able to pursue my dreams. I would never have made it this far without their love and support. Thank you!

Contents

1 Introduction **19**

2 Programmable photonic processors **27**
2.1 Introduction 27
2.2 Waveguides 28
2.3 Fiber-to-chip interfaces 30
2.4 2×2 Couplers 32
2.5 Optical resonators 33
2.6 Phase shifters 35
2.7 Electrical-to-optical interfaces 36
2.7.1 Modulators 36
2.7.2 Photodiodes 37
2.8 Mach-Zehnder interferometers 37
2.9 Towards fully-reconfigurable photonics 38

3 Error correction for programmable photonics **41**
3.1 A manufacturing problem 41
3.2 Hardware Error Correction 44
3.3 What we've learned so far 49
3.4 Hardware Performance 50
3.5 Can we correct errors in neuromorphic hardware? 51
3.6 Hardware error correction for photonic signal processing 54
3.7 Modeling hardware errors from insertion loss 57
3.8 Scaling to larger circuits 63
3.9 What about other types of errors? 68
3.10 Improving the bandwidth of photonic signal processing 70
3.11 Can this scale? 71
3.12 Conclusion 73

4 Alignment-free photonic interconnects **75**
4.1 Introduction . . . 75
4.2 Photonic circuit boards . . . 77
4.3 Theory . . . 79
4.4 Mismatched couplers: what's the efficiency? . . . 81
4.5 Simulation . . . 85
4.6 Discussion . . . 89
4.7 System Integration and Outlook . . . 92
4.8 Conclusion . . . 92

5 Single chip photonic neural network processors **95**
5.1 Introduction . . . 95
5.2 Architecture . . . 96
5.3 A fully-integrated coherent optical neural network . . . 98
5.4 System Packaging, Characterization, and Control . . . 100
5.4.1 Evaluation board . . . 100
5.4.2 Control electronics . . . 101
5.4.3 Transmitter . . . 103
5.4.4 Coherent matrix multiplication unit . . . 105
5.5 Nonlinear optical function unit . . . 109
5.6 Optically accelerating training . . . 113
5.7 Why does training work? . . . 117
5.8 Discussion . . . 118
5.9 Scaling . . . 119
5.10 Conclusion . . . 120

6 The road ahead **121**
6.1 Introduction . . . 121
6.2 Scalable, error-corrected photonic meshes . . . 121
6.3 Large-scale, multi-chip photonic modules . . . 122
6.4 Energy-efficient, high-speed photonic nonlinearities . . . 122
6.5 Optically accelerated neural network training . . . 123

List of Figures

1-1 Historical trends in microprocessor performance over the last fifty years. Data sourced from ref. [2]. 20

1-2 Example architecture of a four-layer deep neural network. Each neuron is connected to a neuron in the following layer through a series of synaptic connections represented by a linear matrix. 21

1-3 Reported performances on the ImageNet task vs. parameter size (data sourced from ref. [12], which aggregates results reported in the literature). State-of-the-art performances on ImageNet require model sizes exceeding 10^9 parameters. 22

1-4 The size of state-of-the-art DNN models for image classification and natural language processing vs. the year they were first reported. 23

2-1 The silicon-on-insulator platform. Waveguides are defined in a 220 nm silicon layer and cladded by silicon dioxide. Insets show the fundamental (TE) modes of a ridge (left) and strip-loaded (right) waveguide. 28

2-2 Footprint comparison between a 5 μm radius bend (left), common for dense routing in silicon photonics, vs. a 50 μm radius bend (right) required for routing on lower index platforms such as lithium niobate and silicon nitride. 30

2-3 Photonic devices on chip are typically interfaced to optical fiber through edge coupling (top) or grating coupling (bottom). 31

2-4 2×2 couplers are realized in silicon photonics using either a directional coupler (left) or a multimode interferometer (right). 32

2-5 Top: an all-pass ring resonator on the silicon photonic platform. Bottom: Transmission of an all-pass ring resonator when the device is critically coupled, i.e. $t = a$. 34

2-6 A Mach-Zehnder interferometer in a silicon photonic circuit. Each MZI consists of two 50-50 splitters, implemented using a directional coupler, and two electrically-programmable thermal phase shifters. 38

2-7 Universal architectures for programmable photonic matrix processors. Rectangular (a) and triangular (b) configurations of MZIs can be used to implement arbitrary unitary operations on optical modes with the Clements [45] and Reck [46] decompositions, respectively. Each MZI, consisting of an internal phase shift θ and an external phase shift ϕ, can perform an arbitrary 2×2 operation on a pair of optical fields. 39

3-1 Simulated variation of the splitting behavior of a directional coupler across a 40x40 mm^2 reticle. An ideal splitter is 50-50; there are some local correlations, but across a reticle the splitting can vary quite significantly. Here, we assume the splitting varies with a standard deviation $\sigma = 2.1\%$ and a correlation length of 5 mm. We obtain these results using the procedure outlined in ref. [65] for simulating layout-dependent correlations in manufacturing. 42

3-2 While we typically model the MZI as an ideal device when developing algorithms for photonic computation, in practice manufacturing variations α, β in the directional couplers produce errors in the gate operation. The effect of these hardware errors is to left- and right-multiply each programmable 2×2 unitary $T_{ij}(\theta, \phi)$ implemented by an MZI by error matrices $\beta_{ij}, \alpha_{ij}(\phi)$. Applying the standard decomposition for ideal components to these imperfect optical gates will not produce the correct gate operation. 44

3-3 Fabrication-induced errors within each MZI can be corrected by applying local corrections $\theta \rightarrow \theta'$, $\phi \rightarrow \phi'$ to the device. We first correct θ to set the magnitudes of the elements of T_{ij} equal to T'_{ij}. Once the amplitude terms are set correctly, we apply phase corrections to the input and outputs of the device to correct phase errors between T_{ij} and T'_{ij}. 45

3-4 The corrections $\phi' - \phi$, $\theta - \theta'$, ψ_1, ψ_2 applied to an MZI with two 52-48 beamsplitters ($\alpha = \beta = 0.02$). The arrows on the plot indicate which vertical axis each curve corresponds to. 47

3-5 The procedure for programming a unitary with hardware errors on a 4×4 rectangular unitary circuit. We first program each MZI to the (θ, ϕ) setting obtained with the standard decomposition in [45]. Each MZI is then converted $T_{ij} \rightarrow T'_{ij}$ to the settings for an imperfect device one column at a time. At each step we propagate the output phase shifts ψ_1, ψ_2 forward in the circuit until the entire network is corrected. 48

3-6 Matrix error ϵ before and after correction for 100 random unitaries implemented on 100 random circuits with varying beamsplitter statistics. . . . 50

3-7 Matrix error ϵ before and after correction for $N = \{64, 128, 256\}$ with a beamsplitter variation $\sigma_{BS} = 2\%$. 51

3-8 Trajectory of light when input into the top-most port of a 64×64 circuit programmed to a Haar random unitary. Note that optical power is equally distributed to all outputs, requiring the main diagonal to be programmed close to the cross ($\theta = 0$) state. 52

3-9 a) The MNIST data set was pre-processed with a Fourier transform and truncated to a $\sqrt{N} \times \sqrt{N}$ center window for a N-mode unitary circuit [73]. b) The activation function architecture as described in [78]. A small fraction α of the input signal is tapped off to a photodiode driving a Mach-Zehnder modulator. c) The activation function $f(E)$ for the parameters used in the simulation. Since the hidden layers operate on electric field amplitudes, we plot the square root of the optical power in units $\sqrt{\text{mW}}$. Technically, $f(E)$ is non-monotonic for high optical powers, as the Mach-Zehnder interferometer will produce a $\cos(|E|^2)$ modulation. However, the input optical powers in our simulations are chosen to ensure the activation function operates only in the modReLU-like region. d) The input vectors into the neural network were normalized to unit length, which can be realized optically with a diagonal line of MZIs. 53

3-10 Architecture of the simulated two-layer optical neural network for the MNIST task. Matrix-vector products are calculated optically in the photonic circuit, and modReLU-like activation functions are implemented electro-optically [79, 78]. The output signal is photodetected and L_2 normalized to generate a quasi-probability distribution for the classification. 54

3-11 Median accuracy for 300 unitary circuits as a function of σ_{BS} with and without correction for a photonic image classifier for the MNIST task with $N = \{36, 64, 144, 256\}$ neurons. Error correction significantly improves the fabrication tolerance of the neural network to beyond current-day process tolerances, even for systems with hundreds of modes. As the inset shows, even circuits with 4% splitter error preserve the baseline performance within 1%. 55

3-12 A tunable dispersion compensator (TDC) can be implemented on a recirculating waveguide mesh with 15 tunable-coupling ring resonators coupled serially to one another. 56

3-13 Model for tunable coupling ring. The ring coupling is set by an MZI with errors α, β and internal phase θ, and the resonance is set with a phase setting ϕ. The coupler is assumed to be lossless and the feedback loop is assumed to have a round-trip transmission a. 56

3-14 After training the mesh parameters to implement a fixed linear group delay dispersion on an ideal model, small beamsplitter errors will introduce variations in the implemented group delay τ profile. Plotted are the group delay profiles for 500 randomly generated circuits before and after correction. Correcting the settings of each TBU restores the desired performance, eliminating the need to retrain on the hardware. Also displayed is the distribution of the group delay dispersion before and after correction. 57

3-15 a) Matrix error ϵ for 100 random unitaries implemented on 100 random circuits for $N = 32$ assuming different loss distributions. The typical and state-of-the-art distributions overlap very closely. b) Matrix error ϵ as a function of N for $\sigma_{\text{BS}} = 2\%$ and different loss distributions. 60

3-16 MNIST classification accuracy for a two layer optical neural network with a) 36; b) 64; c) 144; and d) 256 modes assuming variable optical loss. The results for a unitary circuit presented earlier are plotted for comparison. The typical (orange) and state-of-the-art (red) results overlap very closely with the results for unitary circuits. 61

3-17 Simulations of a tunable dispersion compensator (TDC) implemented on a recirculating waveguide mesh assuming state-of-the-art, typical, and conservative device losses. The top plots show the group delay profile implemented before and after correction, while the bottom histograms show the group delay dispersion. For all loss distributions hardware error correction obtains the desired group delay dispersion, albeit with some additional spread introduced by the loss within the devices. While the group delay profiles for circuits drawn from the conservative distribution appear to show little effect from error correction, it still recovers the required group delay dispersion with high accuracy. 62

3-18 Equations (3.33) and (3.49) for the uncorrected and corrected beamsplitter errors as a function of circuit size N. The scatter plot shows the median error for 12 simulations, showing excellent agreement with the derived expressions. 64

3-19 The probability density function of the internal phase shifter setting θ for $N = \{32, 64, 128\}$. As N increases, $\langle\theta\rangle$ is further biased towards 0. . . . 65

3-20 The probability an MZI must be programmed to a splitting $\theta < \xi$, $\theta > \pi - \xi$ for $N = \{32, 64, 128\}$. $P(\theta > \pi - \xi)$ is orders of magnitude smaller than $P(\theta < \xi)$; thus, we can neglect it when computing the expected corrected hardware error. 66

3-21 $\langle\epsilon\rangle$, $\langle\epsilon_{\text{corrected}}\rangle$ as a function of circuit size N for $\sigma_{\text{BS}} = \{1.2, 2, 4\}\%$. . . . 67

3-22 The relative error contributions from beamsplitter error, thermal drift, and quantization error as a function of circuit size N. If the component errors are left uncorrected, then even small beamsplitter variations produce errors significantly larger than those produced by dynamic effects. Hardware error correction suppresses these component errors to a point where dynamic effects begin to play an important role, particularly if the DAC resolution is low. 68

3-23 Wavelength vs. cross coupling for the optimally tolerant directional coupler design reported in [60]. 70

3-24 Average circuit error as a function of wavelength for $N = \{64, 128, 256\}$ using the optimal directional coupler design in [60]. 71

3-25 Alternate MZI unit cells with superior error scaling. (a), which incorporates an extra phase shifter, can guarantee zero error for splitting errors as high as 70-30. (b) and (c), which incorporate an additional 50-50 splitter and waveguide crossing, respectively, realize "asymptotic fault tolerance," where the error diminishes with increasing circuit size N. 72

4-1 The SAPCB consists of a polymer-laminate film bonded onto an electrical PCB. PICs are flip-chip bonded to the polymer film, which includes a linear, closely-spaced array of single-mode waveguides that carry signals between chips. i): The SAPCB consists of efficient, board-level optical interconnects by making use of an alignment-free "hockey stick" coupler that intersects the polymer waveguides at an angle θ. This approach makes the efficiency of our architecture insensitive to in-plane displacements and permits coupling over a wide range of waveguide pitches. Additionally, intersecting the two waveguides at an angle eliminates the requirement to place PICs onto the SAPCB with sub-micron placement accuracy. ii) The alignment-free coupler also simplifies "pick-and-place" integration of microchiplets into PICs, which enables the introduction of gain, detectors, and single-photon sources into a single chip. iii): Electrical connections can be made in our architecture by punching holes through the polymer film, which permits bump bonding to pads on the electrical PCB. 78

4-2 The alignment-free coupler can be modeled as two waveguides weakly coupled vertically by an evanescent interaction strength κ to one another at an off-axis angle θ. At an arbitrary point z along the propagation, the coupling constant κ will decay exponentially by the vertical offset $z\tan\theta$ with a characteristic decay length γ, i.e. $\kappa(z) = \kappa_0 e^{-\gamma|z|\tan\theta}$. 79

4-3 The theoretical power transfer efficiency η of the alignment-free coupler vs. θ for varying values of κ/γ. At small values of θ, η will oscillate rapidly from minimum to maximum power transfer. The alignment-free coupler should not be used in this regime and it is omitted from the plot for clarity. 80

4-4 Angular dependence of coupling efficiency for varying levels of Δ when $\kappa = 0.05$; $\gamma = 1$. 82

4-5 Phase-matching bandwidth of angular (solid line) and directional (dashed line) coupler for varying levels of κ 83

4-6 a) Effective mode index mismatch $\Delta n = n_{\text{SAPCB}} - n_{\text{PIC}}$ as a function of the PIC (SiN) waveguide width and the SAPCB (polymer) waveguide width. The two waveguide geometries should be engineered such that their modes have equal propagation constants, i.e. $\Delta k = 2\pi\Delta n/\lambda = 0$. b-f) Power transfer efficiency as a function of the PIC waveguide width (b), PIC waveguide height (c), coupling gap (d), wavelength (e), and temperature (f) for the design with parameters in Table I. 87

4-7 The field profile of the alignment-free coupler with parameters in Table I. The insets below show the cross-sectional field profile at varying points along the propagation. 88

4-8 Power transfer efficiency η vs θ for designs with coupling gap $g = 1$ μm and $g = 0.5$ μm. The solid lines indicate FDTD simulation results, while the dotted lines are fit to equation (4.9). 88

4-9 a) Transmission vs. θ for an alignment-free coupler designed to interface a 640 $\times$ 300 nm InP gain microchiplet to a 500 $\times$ 220 nm silicon photonic waveguide. The strong mode confinement in both materials eliminates scattering loss at the intersection, permitting mode transfer with no insertion loss. As a result, the transmission characteristic reproduces nearly perfectly equation (4.9). b) The transmission efficiency as a function of wavelength. The coupler has a 1-dB bandwidth exceeding 230 nm. . . . 89

4-10 Lateral and angular alignment tolerance of the alignment-free coupler compared to inverse tapered edge couplers and tapered adiabatic couplers. The lines indicate the 1-dB coupling efficiency contour as a function of in-plane displacement $\delta r_{||}$ and angular displacement $\delta\theta$. The alignment-free coupler has a combined alignment tolerance $\Delta r_{||}\Delta\theta$ that exceeds current approaches. 91

4-11 a) The alignment-free coupler can interface photonic circuits with differing waveguide pitches and process stacks. As the waveguides interact at an angle, precise matching of the waveguide pitch is not necessary. By varying the coupling angle θ, one can easily optimize the transmission for any coupling gap g. b) The requirement for phase-matching permits simplified routing with minimal crosstalk. By tapering the waveguide to ensure $\Delta k \gg 0$, waveguides can be routed over one another with negligible crosstalk. 93

5-1 A coherent optical deep neural network processor processes the entire model in optics, including linear algebra and nonlinear functions. Each layer directly feeds the optical outputs into the next, enabling processing of an entire DNN with ultra-low latency. 97

5-2 Architecture of the fully-integrated coherent optical neural network (FICONN). Inference is conducted entirely in the optical domain, without readout or amplification between layers. Light is fiber coupled into a single input on the chip and fanned out to the six channels of the transmitter **(i)**. Each channel encodes the amplitude and phase of one element of the input $\mathbf{x}_{(j)}$ into the optical field $\mathbf{a}_{(j)}^{(1)}$ with a Mach-Zehnder modulator and an external phase shifter. The coherent matrix multiplication unit **(ii)**, consisting of a Mach-Zehnder interferometer mesh, implements linear transformations. Programmable nonlinear optical function units **(iii)** realize activation functions $\mathbf{a}_{(j)}^{(n+1)} = f(\mathbf{b}_{(j)}^{(n)})$ by tapping off part of the signal to a photodiode, which drives a cavity off-resonance by injecting carriers into the waveguide. An integrated coherent receiver **(iv)** reads out the DNN output by homodyning the output field with a local oscillator. . . . 98

5-3 The fabricated photonic integrated circuit. This circuit, which consumes a footprint of 6 $\times$ 5.7 mm^2, implements an end-to-end photonic DNN processor and was fabricated in a commercial CMOS foundry. 99

5-4 The printed circuit board interfacing on-chip electronics to the drivers. . 100

5-5 Fully-assembled evaluation board for the PIC with wirebonding, PCB, and fiber attach on mechanical chassis. 101

5-6 Schematic of a single channel of the transmitter board, which implements the Howland current pump architecture. 102
5-7 Test setup in the lab, including evaluation board, custom transmitter and receiver boards, and driver electronics. 103
5-8 Typical fitting procedure for an MZI on the PIC. 104
5-9 a) To determine the elements of the thermal crosstalk matrix M, we drive an aggressor channel j while characterizing the static phase p_0 of channel i. As an example, here we characterize M_{12} by plotting the static phase of channel 1 as a function of the phase setting of channel 2. We fit a linear function to this data to find a crosstalk coefficient of $M_{12} = -0.00735$. b) We benchmark the effectiveness of thermal crosstalk correction by repeatedly trying to program a channel to $\theta_1 = \pi/2$, while setting all other channels to random values. We then determine the actual phase implemented by measuring the output transmission T and computing $2 \arccos \sqrt{T}$. As an example, here we show the results for channel 2, where over 500 random experiments thermal crosstalk correction greatly improves the repeatability of programming a channel to a desired phase. 106
5-10 Calibration procedure for internal phase shifters in the CMXU. The devices along the main diagonal and antidiagonal are calibrated first. Once these devices are characterized, the remainder of the phase shifters can be calibrated by programming devices along the main diagonal. 107
5-11 "Meta-MZI" for calibrating external phase shifters. Two phase shifters in columns $i-1, i+1$ are set to implement a 50-50 beamsplitter. The output transmission of this meta-interferometer, which functions exactly like a discrete MZI, is dependent on the phase difference between the external phase shifters $\Delta\phi = \theta_{2,b} - \theta_{2,a}$. 108
5-12 Measured fidelity of 500 arbitrary unitary matrices implemented on a single layer using a "direct" approach (orange) and an approach that takes into account hardware errors and thermal crosstalk (blue). 109
5-13 Circuit diagram of resonant EO nonlinearity. The photocurrent I_p directly drives a *pn*-doped resonant modulator. No amplifier stage is required between the two and the devices are directly connected on chip. By adjusting the bias voltage V_B, the nonlinearity can be operated in forward or reverse bias. 110
5-14 Left: Detuning of the cavity resonance at various incident optical powers when operated in carrier injection mode ($V_B > 0$). Right: Cavity detuning in carrier depletion mode ($V_B < 0$). Our system realizes close to a linewidth detuning without the use of any amplifier, improving energy consumption and latency of the nonlinearity. A full linewidth detuning can be realized by further engineering the cavity finesse. 111
5-15 a) Phase shift $\Delta\phi$ in cavity vs. incident photocurrent. b) Round-trip amplitude loss a as a function of incident photocurrent. As photocurrent increases more carriers are injected into the waveguide, increasing the loss of the optical signal inside the resonator. 112

5-16 Activation functions measured on chip. Programmable function shapes can be realized by adjusting the cavity detuning $\Delta\lambda$ and fraction of light β tapped off to the photodiode. 113
5-17 a) A multivariate cost function $\mathcal{L}(\mathbf{\Theta})$ can be minimized by computing the directional derivative of the function along a random direction (black). This directs the optimization along the component of the gradient (red) parallel to the search direction. Over multiple iterations, the steps taken along random directions average to follow the direction of steepest descent to the minimum. b) *In situ* training procedure. At every iteration, the directional derivative of the cost function $\mathcal{L}(\mathbf{\Theta})$ is computed in hardware along a randomly chosen direction $\mathbf{\Delta}$ in the search space. $\mathbf{\Delta}$ is chosen from a Bernoulli distribution to be $\pm\delta$. The weights $\mathbf{\Theta}$ are then updated by the measured derivative following a learning rate η chosen as a hyperparameter of the optimization. 114
5-18 *In situ* training of a photonic DNN for vowel classification. We obtain 92.5% accuracy on a test set, which is comparable to the performance (92.5%) obtained on a digital model with the same number of weights. Despite not having direct access to gradients, our approach produces a training curve similar to those produced by standard gradient descent algorithms. 115
5-19 Performance of the digital model on the vowel classification task. The model overfits the training set, achieving 100% accuracy, but performance on the test set is comparable to the accuracy achieved by our system (92.5% on the digital model vs. 92.5% on the FICONN). 116

List of Tables

1.1 Computation and communication energies for modern digital processors. Values sourced from ref. [3]. 20

4.1 Simulation parameters for the alignment-free photonic coupler. 86

1 Introduction

The end of Dennard scaling

For most of the history of the semiconductor industry, exponential growth in transistor counts has directly correlated to exponential growth in computing power. This can be explained through Dennard's law [1], which observed that with each successive technology node:

- Each dimension d of the transistor reduces by $\sqrt{2}$, reducing the total area $A = d^2$ by a factor of 2.
- As a result, the capacitance C of the device, which is $\epsilon A/d$, reduces by a factor of $\sqrt{2}$.
- A constant electric field E across the device necessitates reducing the voltage $V = E/d$ by a factor of $\sqrt{2}$.
- As circuit delay is proportional to $1/d$, the clock frequency f can increase by a factor of $\sqrt{2}$.
- Thus, the total power dissipation CV^2f reduces by a factor of $1/2$.

In other words, even though transistor density has doubled, the total power dissipation will remain constant. This advantageous scaling fueled rapid increases in computing power for over thirty years.

However, as of 2005, this scaling has greatly slowed down. Figure 1-1 shows historical trends in transistor count, clock frequency, single-thread performance and power dissipation over the last fifty years [2]. Note that while transistor count continues to increase, clock frequency, single-thread performance and power consumption have flatlined. Thermal limits have bottlenecked the continual scaling of microprocessors; as transistor sizes decrease, leakage currents due to tunneling and other quantum-mechanical effects greatly increase power dissipation across the chip. As a result, operating voltages have not continued to scale with transistor size, slowing improvements in energy efficiency with each progressive technology node.

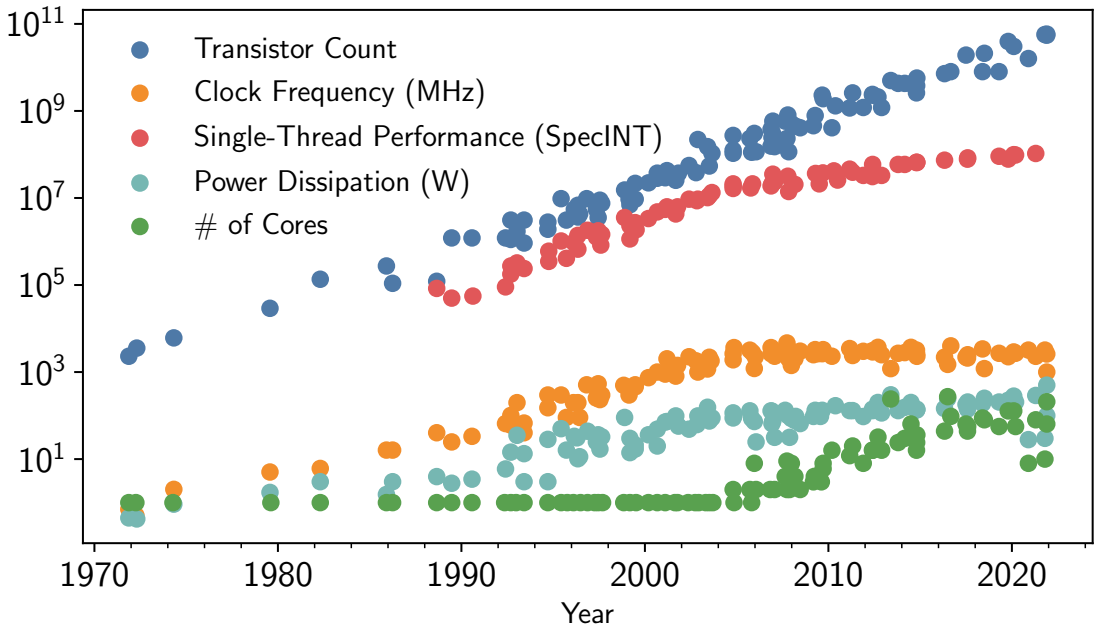

Figure 1-1: Historical trends in microprocessor performance over the last fifty years. Data sourced from ref. [2].

Table 1.1: Computation and communication energies for modern digital processors. Values sourced from ref. [3].

Operation	**Energy/bit**
Switching CMOS gate	50 aJ-3 fJ
Energy in DRAM cell	10 fJ
Floating point operation	100 fJ
Communicating across chip	600 fJ
Data link multiplexing and timing circuits	2 pJ
Reading DRAM	5 pJ
Communicating off chip	1-20 pJ

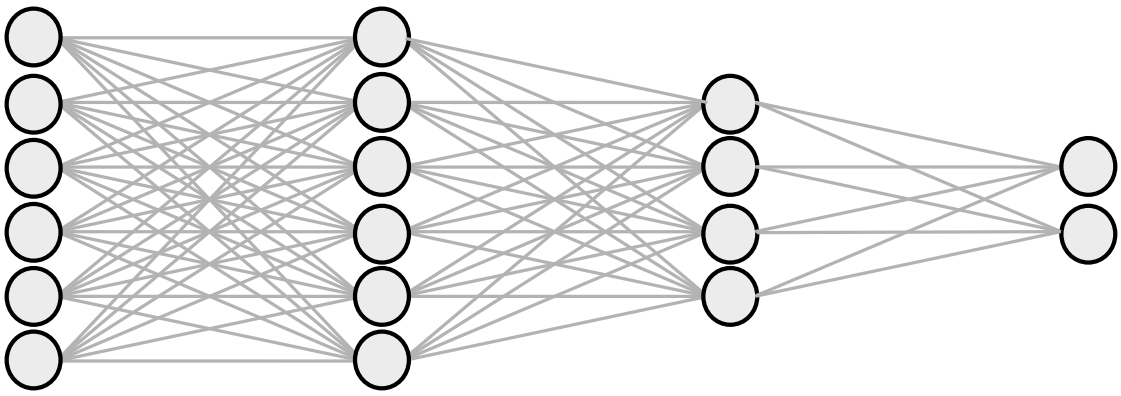

Figure 1-2: Example architecture of a four-layer deep neural network. Each neuron is connected to a neuron in the following layer through a series of synaptic connections represented by a linear matrix.

Moreover, as shown in Table 1.1, the vast majority of energy consumed in modern microprocessors is consumed not for logic, which is on the order of single fJ per CMOS gate, but communications, which can consume anywhere from hundreds of fJ to tens of pJ per bit [3]. As the physical dimensions of the metal wires in a CMOS chip shrink, the interconnect capacitance stays roughly constant or can increase [3]. Since communicating across a transmission line requires charging the line capacitance up to the signaling voltage V, interconnect capacitance $C_{\text{interconnect}}$ does not scale with geometry, and operating voltage V has stopped decreasing with progressive transistor nodes, the energy consumption $C_{\text{interconnect}}V^2/2$ of communication on-chip remains bottlenecked to about ~ 1 pJ/bit. The upshot of this is that while power dissipation for computation continues to drop, the end-to-end system power dissipation is decreasing far more gradually[1].

Deep neural networks and application-specific computing

At the same time as growth in computing power is slowing down, the demands of computation continue to grow. Deep neural networks have revolutionized computing, realizing state-of-the-art performances in tasks ranging from image classification [5, 6], natural language processing [7], signal processing, game playing [8, 9], chip design [10], and engineering.

A simplified architecture of a deep neural network (DNN) is shown in Figure 1-2. Input neurons are connected to output neurons through a set of synaptic connections, which can be represented as a linear matrix transformation. After each layer, a nonlinear activation function, emulating the firing threshold of a biological neuron, is applied to each output neuron. These results are then fed into the following layer and the process continues to realize a *deep* network. This combination of linear and nonlinear mappings between

[1] In order to continue scaling computational power, the industry has moved to integrating multiple logical cores on chip, as shown in Figure 1-1. However, as there still remains a fixed thermal design power (TDP) that can be dissipated on chip, not all cores can be used at the same time and vast sections of the circuitry may be off at a given time. This design strategy has been termed "dark silicon" [4].

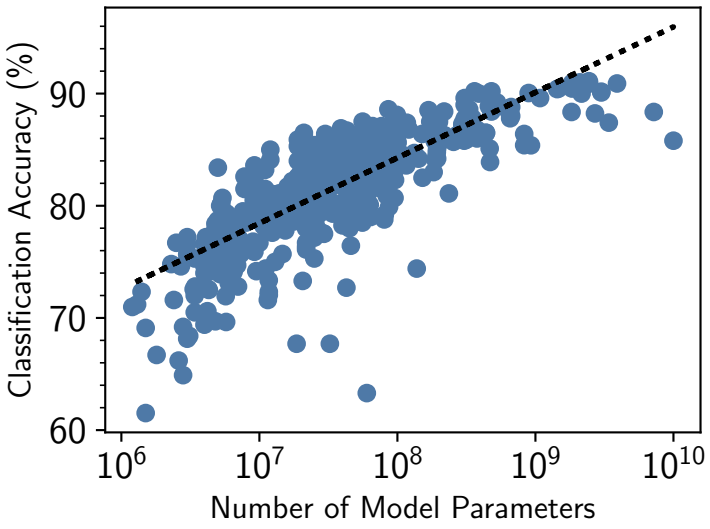

Figure 1-3: Reported performances on the ImageNet task vs. parameter size (data sourced from ref. [12], which aggregates results reported in the literature). State-of-the-art performances on ImageNet require model sizes exceeding 10^9 parameters.

neurons renders DNNs extremely powerful computationally; theoretically, a DNN with an infinite number of layers, i.e. infinitely deep, or with an infinite number of neurons, i.e. infinitely wide, can be mathematically proven to be a universal function approximator [11].

While the idea of deep neural networks has been around for decades, it is only recently that they have become the state-of-the-art in computing. This is directly tied to the exponential increase in the size of these models, and the growth in computing power that has enabled this trend. Figure 1-3 shows performance on the ImageNet classification task as a function of model parameters across 590 papers reported in the literature [12]. A clear trend can be observed where state-of-the-art performance is correlated with parameter number and classification accuracies exceeding 90% require model sizes on the order of 10^9 weights[2].

The size of DNN models has also grown exponentially with time. Figure 1-4 shows the size of state-of-the-art models vs. year they were first reported. At the time this thesis was written, the newest language processing models such as GPT-3 and Megatron-Turing NLG required over 100 billion parameters[3].

The growing demands on computation has led both researchers and the industry to start developing systems for application-specific computing. A typical microprocessor, such as a CPU, is general-purpose. However, from the perspective of processing a deep neural network, the CPU includes a great deal of bells and whistles that are not only completely unnecessary, but drive up the power consumption and bottleneck the computation. This was recognized in the early 2000s, when the first graphical processing units (GPUs) were being re-purposed specifically to accelerate linear algebra processing for DNNs. The

[2]In other words, the model requires about 1 GB of storage assuming each weight is an 8-bit integer.
[3]Or 100 GB.

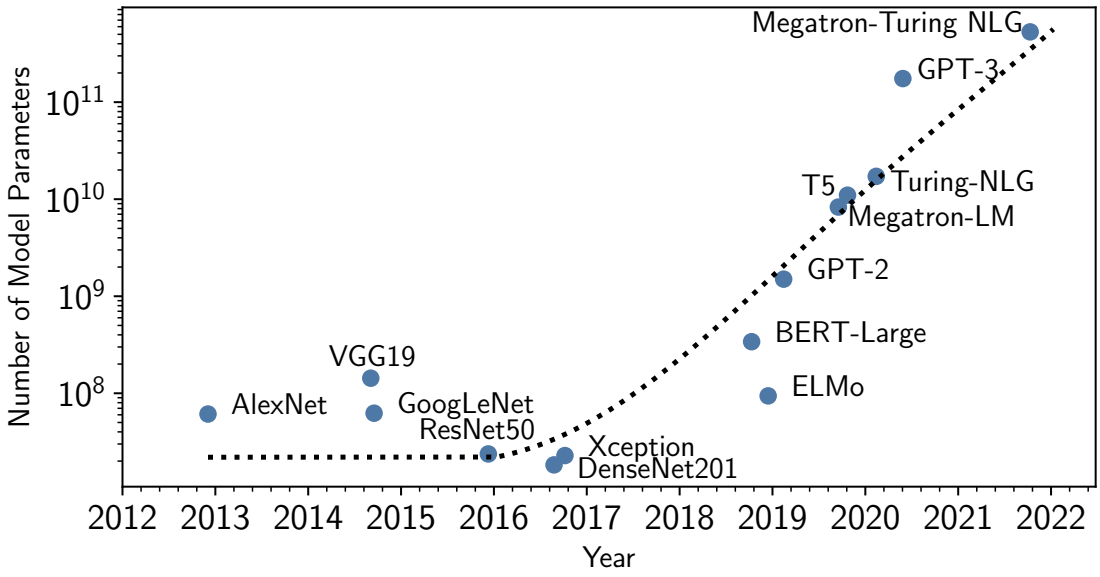

Figure 1-4: The size of state-of-the-art DNN models for image classification and natural language processing vs. the year they were first reported.

transition in the industry to application-specific processors accelerated the advances made in the early 2010s in machine learning.

However, a GPU is still not truly application-specific. They are optimized for graphics processing on computers; while many of the features of these systems make them ideal for parallel matrix processing, a truly optimized application-specific integrated circuit for DNNs could realize further speedups. In 2017, Google announced the development of the tensor processing unit, or TPU, which was specifically designed to accelerate matrix processing for artificial intelligence [13]. Impressively, this system realized an order-of-magnitude performance gain over state-of-the-art GPUs, representing a landmark result for the field of application-specific computing.

While the tensor processing unit showed the potential of a fully-optimized digital computer for matrix processors, it had certain limitations endemic to digital systems. The designers found that although their compute unit was highly efficient, the overall system performance was bottlenecked by access to and from memory—in other words, the interconnect problem discussed earlier. This ultimately limited the energy efficiency of the system to be ≈ 1 pJ/OP[4].

Optics for computing?

The bottlenecks inherent to digital systems has motivated interest in alternative architectures for computing. One of the most exciting areas of research in recent years has been optical systems for computation. While modern electronic processors for DNNs are bottlenecked by the power dissipation of interconnects, optical systems are particularly

[4]Although recent research results [14] in digital systems has improved this to 100 fJ/OP.

well-suited for addressing the problem of data movement. In electronic interconnects, energy consumption scales with link length; as a result, fetching data over long distances from memory is energetically quite expensive. Optical interconnects, however, exhibit a length-independent scaling in energy consumption originating only from the expense of electrical-to-optical conversion and back.

Optical systems are also exceptional at particular forms of computation—in particular, linear operations. While linear matrix operations can consume significant amounts of energy in digital processors, a carefully-designed optical architecture can perform matrix processing passively and with speed-of-light limited latencies. Other important linear functions, such as fanout, where we copy a single signal and distribute it to N nodes, also consume immense amounts of energy in digital systems and are trivial to realize in optics[5].

Linear processing is the key computation for deep neural networks and consumes the vast majority of the power dissipation. Given that linear matrix operations can be performed passively in optics, the idea of application-specific optical systems for DNNs suddenly starts to make a lot of sense. When I started graduate school, a set of landmark results in optical processing of DNNs, including one from our group [15, 16, 17, 18], had begun to be reported. In a series of proof-of-principle demonstrations, these groups leveraged key advantages of optics to realize computation of DNNs, namely:

- Unlike electronics, optics can realize **massively parallel data processing** — multiplexing across spatial, polarization, and wavelength modes, enabling potentially high throughput [17, 18]. Fiber optical communications leverages this today to achieve datarates exceeding terabits per second.
- **Passive linear matrix operations**–the bulk of DNN computation–performed "for free" [15, 17].
- Optical matrix processing that is time-of-flight limited in latency, i.e. **clockless** processing of data [15, 18, 17].
- Matrix processing of **coherent** optical data [15].

While optics is fantastic for linear processing, there remain many challenges for realzing an end-to-end optical DNN processor. For example:

- Optical systems require careful, sub-micron alignment and extraordinary **stability**. While many benchtop optical setups in the lab are on the order of tens to hundreds of components, realizing accurate optical computation at useful scales will require the ability to stabilize millions of optical components.
- Moreover, we need to be able to interface **large numbers of channels** in and out of these optical systems to scale to useful model sizes.
- The computation is analog. Analog computation is notorious for being low-precision, as errors in these systems cascade. **Novel error correction algorithms** will be needed to scale these systems.

[5]For example with a diffractive optical element.

- The computational power of DNNs originates from the combination of both linear and nonlinear transformations in the function. While performing linear operations in DNNs is well understood, realizing **efficient nonlinear operations in optics** remains quite difficult.
- **Training** deep neural networks in digital systems is well understood. However, it is an open question as to what are the most efficient algorithms for training computational models on photonic hardware.

Scaling up optical systems for artificial intelligence will eventually require surmounting all of these challenges.

What this thesis is about

This thesis describes the work I did in my PhD to tackle some of these emerging challenges at the intersection of **optics** and **computing**. In particular, I have focused on programmable silicon photonic systems, which enable dense integration of photonic computing units, rapid reconfigurability, and complex interferometric processing of optical signals for deep neural networks.

- In **Chapter 2**, I introduce the key components of an integrated, programmable photonic processor for deep neural networks, which include waveguides for routing high-bandwidth optical signals, linear optical components for interferometric processing, and high-speed modulators and photodetectors for efficient electrical-to-optical interfaces.
- **Chapter 3** addresses the obstacle of increasing component error in programmable photonic processors. We report the development of the first deterministic, gate-by-gate error correction algorithm for programmable photonics, whose scaling was previously believed to be limited by fabrication error. We show that this algorithm enables scaling of photonic processors for DNNs to commercially relevant sizes.
- A key obstacle to realizing end-to-end photonic processors for deep neural networks, which will require hundreds of optical channels, is the high cost and low yield of photonic packaging. We report in **Chapter 4** the development of a novel paradigm for photonic interconnects that is highly tolerant to alignment error, realizing tolerances that are significantly higher than conventional interconnects and enabling the use of low-precision, high-volume assembly tools for photonic packaging.
- In **Chapter 5** we report the experimental realization of the first chip-scale system for end-to-end, photonic processing of deep neural networks. Our system, which monolithically integrates optical processing units for matrix algebra and nonlinear functions into a single silicon photonic chip, performs both inference and training of deep neural networks entirely in the optical domain.
- Finally, in **Chapter 6**, we summarize the main results of this thesis and discuss future research directions towards the ultimate goal of realizing end-to-end photonic systems for computing.

2 Programmable photonic processors

2.1 Introduction

An outstanding goal of the optics research community has been dense integration of optical components on-chip for enabling complex systems, similar to advances in very-large scale integration (VLSI) in the 20th century that enabled the fabrication of processors with millions of transistors on a single die. A number of promising platforms for achieving this integration have been studied by the academic community, including lithium niobate [19], gallium arsenide [20], indium phosphide [21], aluminum nitride [22], and many others.

Silicon photonics has emerged over the last decade as the state-of-the-art platform for large-scale integrated optical circuits. Part of the reason for this is economics: silicon is inexpensive, billions of dollars have been invested worldwide in the CMOS platform, providing for a highly mature infrastructure for fabrication, and optics fabricated on silicon could potentially be co-integrated with electronics within the same process.

In addition to its advantageous economics, silicon possesses a high-quality native oxide suitable for cladding, has excellent thermal and electrical properties, and is transparent at telecommunications wavelengths used in fiber optics. Moreover, the silicon-on-insulator (SOI) platform that uses silicon dioxide as the cladding has exceptionally high index contrast ($n_{\text{Si}} = 3.5$ and $n_{\text{oxide}} = 1.44$) relative to competing platforms, allowing for waveguide dimensions on the order of hundreds of nanometers and bend radii as small as 5 μm [23], permitting dense component integration. This has enabled the realization of complex optical processors in the silicon platform, with applications ranging from quantum information processing [24, 25, 26] to telecommunications [27] to signal processing [28] to artificial intelligence [15].

In this section, we introduce the key elements of the silicon photonic platform. We then discuss their application to programmable photonic circuits [29], an emerging class of integrated optical systems that are promising for reconfigurable signal processing in the optical domain.

2.2 Waveguides

The most fundamental component in an integrated photonics platform is the waveguide, which consists of a high refractive-index "core" surrounded by a lower index "cladding." The waveguide confines light in two spatial dimensions (x and y), while permitting propagation in the third (z); thus, they serve as optical "wires" in a circuit, transporting optical signals from component to component for further processing.

Optical propagation in a waveguide can be described by a set of "modes", which refer to electromagnetic fields within the structure that do not vary in spatial profile as they propagate. Due to the translational invariance of the waveguide structure along the propagation (z) direction, the field distribution between two points $E(x, y, z)$ and $E(x', y', z')$ in a waveguide can be related by a phase factor $E(x', y', z') = E(x, y, z)e^{i\beta(z'-z)}$, where β denotes the propagation constant of the mode [30]. One can calculate the electromagnetic modes of a structure with Maxwell's equations:

$$\nabla \cdot (\epsilon(\mathbf{r})\mathbf{E}(\mathbf{r}, t)) = 0 \tag{2.1}$$

$$\nabla \cdot \mathbf{H}(\mathbf{r}, t) = 0 \tag{2.2}$$

$$\nabla \times \mathbf{E}(\mathbf{r}, t) = -\mu_0 \frac{\partial \mathbf{H}(\mathbf{r}, t)}{\partial t} \tag{2.3}$$

$$\nabla \times \mathbf{H}(\mathbf{r}, t) = \epsilon_0 \epsilon(\mathbf{r}) \frac{\partial \mathbf{E}(\mathbf{r}, t)}{\partial t} \tag{2.4}$$

Here, we modify the equations to assume that ϵ, the dielectric constant of the medium, can be a function of position. The general propagating solution to these equations are plane waves of the form $\mathbf{H}(\mathbf{r}, t) = \mathbf{v}H(x, y)\exp(i(kz - \omega t))$, where k is the wavenumber,

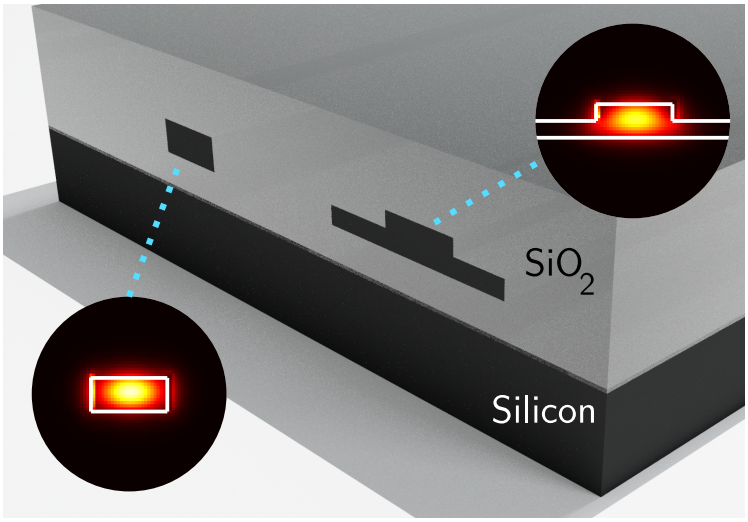

Figure 2-1: The silicon-on-insulator platform. Waveguides are defined in a 220 nm silicon layer and cladded by silicon dioxide. Insets show the fundamental (TE) modes of a ridge (left) and strip-loaded (right) waveguide.

ω is the frequency, and $\mathbf{v}$ is the direction of the magnetic field $\mathbf{H}$ [31]. The use of complex exponentials simplifies the underlying mathematics; the physical magnetic field can be obtained by taking the real part of the expression, and the electric field can be found by substituting the solution into equation (4). More complicated modes can be described by expanding them as a weighted sum of plane waves

$$\mathbf{H}(\mathbf{r}, t) = \sum_k \mathbf{v} H_k(x, y) \exp(i(kz - \omega(k)t)) \tag{2.5}$$

We can find a mode satisfying equations (1) and (2) by ensuring that $\mathbf{v}$ is orthogonal to $\mathbf{r}$, making the modes calculated tranverse to the direction of propagation. We satisfy equations (3) and (4) by taking the curl of both sides of equation (4):

$$\nabla \times \left(\frac{1}{\epsilon(\mathbf{r})} \nabla \times \mathbf{H}(\mathbf{r}, t) \right) = \nabla \times \epsilon_0 \frac{\partial \mathbf{E}(\mathbf{r}, t)}{\partial t} \tag{2.6}$$

$$= \frac{\partial}{\partial t} \left[\nabla \times \epsilon_0 \mathbf{E}(\mathbf{r}, t)) \right] \tag{2.7}$$

$$= \frac{\partial}{\partial t} \left[-\epsilon_0 \mu_0 \frac{\partial \mathbf{H}(\mathbf{r}, t)}{\partial t} \right] \tag{2.8}$$

$$= -\frac{1}{c^2} \frac{\partial^2 \mathbf{H}(\mathbf{r}, t)}{\partial t^2} \tag{2.9}$$

Now substituting our plane wave solution $\mathbf{H}(\mathbf{r}, t) = \mathbf{v} H(x, y) \exp(i(kz - \omega t))$, we find

$$\nabla \times \left(\frac{1}{\epsilon(\mathbf{r})} \nabla \times \mathbf{H}(\mathbf{r}, t) \right) = \left(\frac{\omega}{c} \right)^2 \mathbf{H}(\mathbf{r}, t) \tag{2.10}$$

where we have replaced $\epsilon_0 \mu_0$ with $1/c^2$.

We can rewrite the left side as a Hermitian operator $\hat{\Theta} = \nabla \times \left(\frac{1}{\epsilon(\mathbf{r})} \nabla \times \right)$, such that the above equation is $\hat{\Theta}\mathbf{H} = \left(\frac{\omega}{c} \right)^2 \mathbf{H}$ [31]. This equation has now taken the form of an eigenvalue problem whose solutions will be the modes of the structure. The eigenvalues correspond to the propagation constant k, which can be related to the vacuum frequency ω_0 through $\omega_0 = ck/n_{\text{eff}}$. n_{eff}, referred to as the effective index, is an important parameter of waveguide modes that describes the phase velocity of light in the structure.

For a waveguide, one can find the propagating modes by defining a two-dimensional cross section with separate dielectric constants $\epsilon(\mathbf{r})$ defined for the core and cladding regions and then solving the above continuous eigenvalue problem. Solving the problem analytically, however, is quite challenging; in practice, the problem is solved computationally by discretizing the region into a space of sub-wavelength sized cells and then solving the problem numerically with approaches such as finite-element methods (FEM), the finite difference time domain (FDTD) method, the boundary integral and resonant mode expansion (BI-RME) method, or the Lanczos method [32]. For a waveguide, pictured in Figure 2-1, one will find modes corresponding to polarizations along the two transverse axes of the structure; by convention, the **x**-polarized mode is labeled "TE" and the **y**-polarized mode is labeled "TM." In most (but not all) situations, the waveguide will be designed

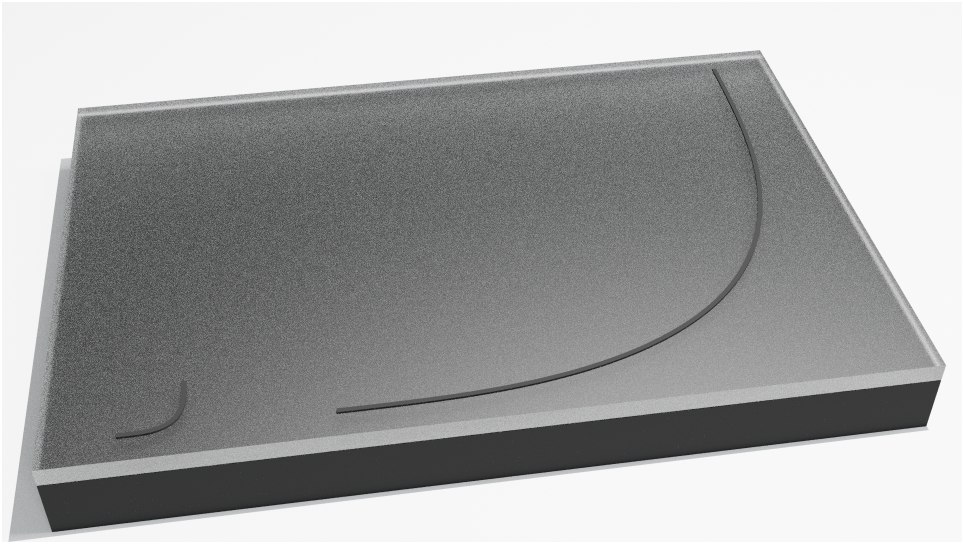

Figure 2-2: Footprint comparison between a 5 μm radius bend (left), common for dense routing in silicon photonics, vs. a 50 μm radius bend (right) required for routing on lower index platforms such as lithium niobate and silicon nitride.

such that there is only one solution to the eigenvalue problem above; such "single-mode" waveguides allow the device designer to presume that any light coupled into the structure will be in the waveguide's fundamental mode. Thus, the device designer will know at any point along the waveguide exactly what the transverse field profile will be.

A key advantage of the silicon photonic platform is the high index contrast between core and cladding, which results in strong modal confinement that enables dense routing with bend radii of < 5 μm [23]. This is in contrast to lower-index platforms such as silicon nitride and lithium niobate, which require bend radii on the order of 50 μm. To give a sense of the scale, in Figure 2-2 we show the footprint occupied by a 5 μm bend compared to one that is 50 μm.

2.3 Fiber-to-chip interfaces

Eventually, light on chip will need to interface to the outside world. This requires efficient interfaces between waveguides on chip and optical fiber. Figure 2-3 shows two common approaches to achieving this interface:

- An **edge coupler** is comprised of a waveguide that adiabatically tapers to expand the size of the optical mode. Once the mode has been expanded, it is launched into free space and end-fire coupled into the optical fiber.

  A key challenge for edge coupling is the difficulty of converting the waveguide mode, which has a spot size of < 1 μm diameter, to the 10 μm mode-field diameter that is used in standard single mode fiber. Several approaches have been pursued to tackle this problem, including metamaterial designs [33] and "trident" couplers that construct a supermode using multiple waveguides [34].

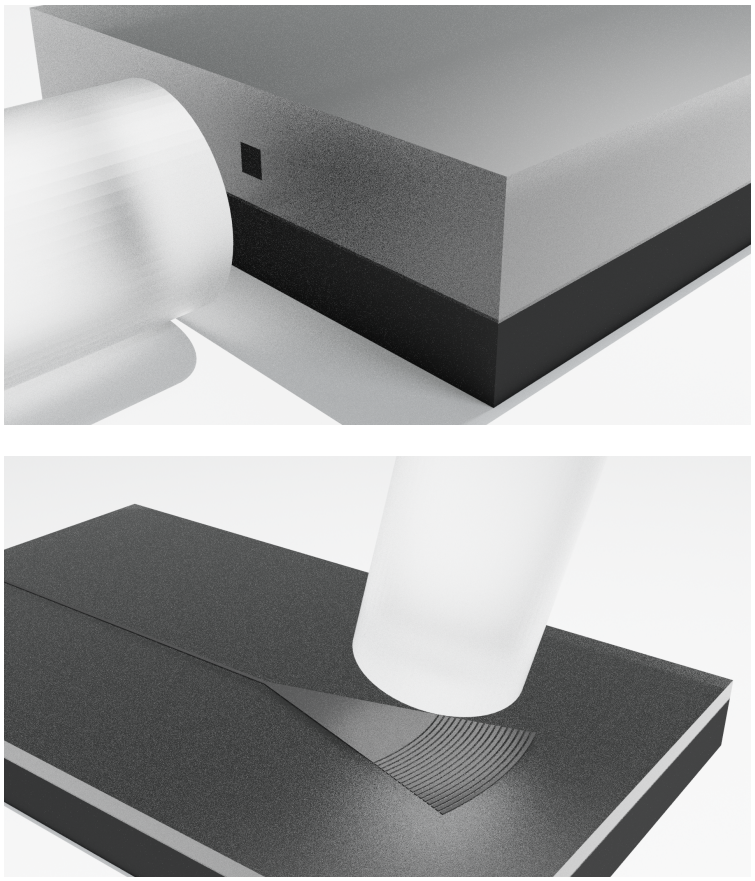

Figure 2-3: Photonic devices on chip are typically interfaced to optical fiber through edge coupling (top) or grating coupling (bottom).

- **Grating couplers** realize a periodic waveguide structure that can diffract incident light from a waveguide mode into free space. Here, the idea is to realize a waveguide period that matches the Bragg condition [30]:

$$n_{\text{eff}} - n_c \sin\theta = \frac{\lambda}{\Lambda} \tag{2.11}$$

  where n_{eff} is the effective index of the waveguide mode, n_c is the cladding index, θ is the diffraction angle from the vertical, λ is the wavelength, and Λ is the grating period.

  As grating couplers diffract light vertically, they can be placed anywhere on a photonic circuit. This makes circuit design simpler than using edge couplers, which must be placed at the chip edge.

  The main disadvantage of grating coupling is the higher insertion loss, due to low diffraction efficiency, and the limited bandwidth–as n_{eff} will vary with λ, the Bragg condition will only be realized for a limited range of wavelengths.

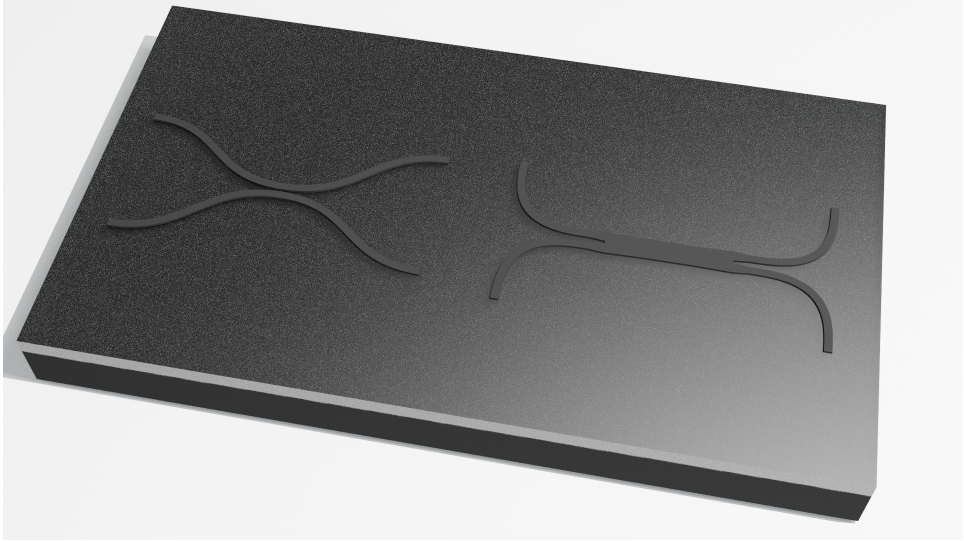

Figure 2-4: 2×2 couplers are realized in silicon photonics using either a directional coupler (left) or a multimode interferometer (right).

A key metric for both types of devices is the mode overlap, which determines the coupling efficiency η to fiber [35]:

$$\eta = \frac{|\int E_1^* E_2 \, \mathrm{d}A|^2}{\int |E_1|^2 \, \mathrm{d}A \int |E_2|^2 \, \mathrm{d}A} \tag{2.12}$$

Misalignment in the fiber reduces the mode overlap and therefore the coupling efficiency η.

2.4 2×2 Couplers

Programmable photonic systems require the ability to coherently split and re-combine light. A fundamental component of these systems is the 50-50 beamsplitter, which applies to two input optical modes a_0, a_1 the coherent matrix operation:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \tag{2.13}$$

For an optical field input into a single port, i.e. input vector $[1, 0]$, this device applies an amplitude transmission of $1/\sqrt{2}$ to each output port, corresponding to 50-50 splitting in power. These devices can be used to implement reconfigurable beamsplitters on chip, as we describe later in this chapter.

In photonic platforms, 2×2 couplers are realized using either directional couplers or multimode interferometers. In a **directional coupler**, two waveguides are perturbatively coupled through the interaction of the evanescent field of each waveguide mode. This interaction results in two hybridized modes that are even (in phase) and odd (in anti-phase) [36]. Assuming a constant coupling κ, the coupled mode equations describing the

system are:

$$i\frac{\mathrm{d}}{\mathrm{d}z}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \beta & \kappa \\ \kappa & \beta \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \tag{2.14}$$

When light is input into a single port, i.e. $a_0 = 1; a_1 = 0$, these equations can be solved to find that:

$$a_0(z) = \cos(\kappa z) \tag{2.15}$$

$$a_1(z) = \sin(\kappa z) \tag{2.16}$$

For two identical waveguides, κ can be computed as:

$$\kappa = \frac{\pi(n_{\text{even}} - n_{\text{odd}})}{\lambda} \tag{2.17}$$

where n_{even}, n_{odd} are the effective indices of the even and odd supermodes of the structure.

Directional couplers are extremely low loss (< 0.1 dB) and can reach nearly ideal 50-50 splitting behavior. For this reason, they are the splitter of choice in most programmable photonic circuits. However, they are strongly wavelength dependent, as the effective index n_{eff}, and therefore Δn, will vary with λ. This makes it difficult to use programmable circuits built with directional couplers over a wide wavelength range.

Alternatively, **multimode interference** (MMI) couplers can be used to realize 50-50 splitting. In these devices, two waveguides interface to a multimode region in which interference takes place. At specific self-imaging lengths, this device can realize a 2×2 splitting operation identical to a directional coupler.

The self-imaging length depends on the difference in effective indices between the TE0 and TE1 modes of the waveguide. This usually depends less on wavelength, making MMI couplers ideal for broadband behavior. The tradeoff is that coupling in and out of the multimode region efficiently is challenging, and thus these devices typically have higher insertion loss than directional couplers.

2.5 Optical resonators

In silicon photonics, optical microcavities are frequently implemented using **ring resonators**, which comprise a loop of waveguide that is evanescently coupled to an external bus waveguide. Resonances occur whenever the mode constructively interferes after a round-trip in the loop, i.e. when:

$$m\lambda = n_{\text{eff}}L \tag{2.18}$$

where m is an integer, n_{eff} is the effective index, and L is the round-trip length around the ring. The spacing between resonances in wavelength, i.e. the free spectral range, can be computed as:

$$\text{FSR} = \frac{\lambda^2}{n_g L} \tag{2.19}$$

where $n_g = n_{\text{eff}} - \lambda(\mathrm{d}n_{\text{eff}}/\mathrm{d}\lambda)$ is the group index of the waveguide.

In an all-pass ring resonator, as shown in Figure 2-5, input light is coupled from the bus to the resonator with amplitude transmission $i\kappa$. An optical field coupled into the ring

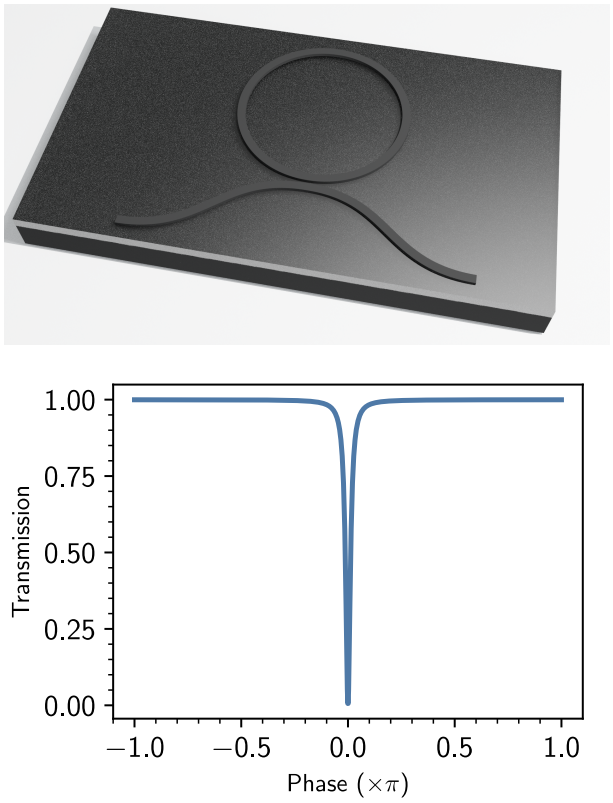

Figure 2-5: Top: an all-pass ring resonator on the silicon photonic platform. Bottom: Transmission of an all-pass ring resonator when the device is critically coupled, i.e. $t = a$.

will make a round-trip around the cavity, accumulate $\phi = 2\pi n_{\text{eff}} L/\lambda$ in phase, attenuate by an amplitude transmission a corresponding to the round-trip loss in the cavity, and then couple back to the bus with probability $i\kappa$. Assuming a lossless coupler with a pass-through transmission t, i.e. $|t|^2 + |\kappa|^2 = 1$, we can compute the output field as:

$$E_{\text{out}} = tE_{\text{in}} + \left((i\kappa)^2 ae^{i\phi} + (i\kappa)^2 t(ae^{i\phi})^2 + (i\kappa)^2 t^2 (ae^{i\phi})^3 + ...\right) E_{\text{in}} \tag{2.20}$$

$$= tE_{\text{in}} - \kappa^2 ae^{i\phi} \sum_{n=0}^{\infty} (tae^{i\phi})^n E_{\text{in}} \tag{2.21}$$

$$= tE_{\text{in}} - \frac{\kappa^2 ae^{i\phi}}{1 - tae^{i\phi}} E_{\text{in}} \tag{2.22}$$

$$= \frac{(1 - tae^{i\phi})t - \kappa^2 ae^{i\phi}}{1 - tae^{i\phi}} E_{\text{in}} \tag{2.23}$$

$$= \frac{t - t^2 ae^{i\phi} - \kappa^2 ae^{i\phi}}{1 - tae^{i\phi}} E_{\text{in}} \tag{2.24}$$

$$= \frac{t - ae^{i\phi}}{1 - tae^{i\phi}} E_{\text{in}} \tag{2.25}$$

The amplitude transfer function is therefore:

$$\frac{E_{\text{out}}}{E_{\text{in}}} = \frac{t - ae^{i\phi}}{1 - tae^{i\phi}} \tag{2.26}$$

This expression is plotted in Figure 2-5. At resonance, i.e. $\phi = 2\pi m$, the ring drops a substantial amount of power to the loss mode. At critical coupling, i.e. when $t = a$, all of the power is lost and the through-port transmission will drop to zero.

At resonance, the feedback introduced by constructive resonance greatly increases the effective path-length, as light travels around the ring many times. Resonators are therefore useful for enhancing weak optical effects, such as nonlinear interactions or high-speed, carrier-based modulation (which typically is rather inefficient and requires long device lengths). The degree of enhancement can be related to the *finesse* F[37], which is roughly the number of times light travels around the ring:

$$F = \frac{\pi\sqrt{ta}}{1 - ta} \tag{2.27}$$

A related quantity is the quality factor (or Q-factor), which also quantifies the width of the resonance:

$$Q = \frac{\Delta\lambda}{\lambda} = \frac{\pi\sqrt{ta}}{1 - ta}\left(\frac{n_g L}{\lambda_{\text{res}}}\right) \tag{2.28}$$

This is related to the photon lifetime τ by $Q = \omega_0\tau$. The photon lifetime quantifies how long, in time, optical power "lives" within the cavity. When designing modulators, high Q can reduce the modulation voltage as the finesse is increased. However, it also diminishes the bandwidth $f_{\text{BW}} = 1/(2\pi\tau)$, as each cycle requires the optical field within the cavity to be fully charged and discharged.

2.6 Phase shifters

Reconfigurability in a photonic circuit requires the ability to actively tune photonic elements. An important degree of freedom in optics is the phase; thus, programmable control over the phase of an optical field is of paramount importance in these systems.

Phase shifters in silicon photonics make use of the **thermo-optic effect**. At room temperature, the thermo-optic coefficient of silicon $dn/dT = 1.8 \times 10^{-4}$ K^{-1}. Increasing the local temperature T of a waveguide of length L will induce a phase shift:

$$\Delta\phi = \frac{2\pi}{\lambda}\left(\frac{\mathrm{d}n}{\mathrm{d}T}\right)(\Delta T)L \tag{2.29}$$

Thermal tuning is typically realized using resistive heating, where a current is driven into an on-chip resistor to generate Joule heating through I^2R dissipation. This process can be remarkably efficient, particularly when the resistor is integrated into the waveguide; for instance, in ref. [38], the silicon waveguide itself is weakly p-doped and connected to metal contacts. As the waveguide itself is a resistor, driving a current into the device produces local heating that efficiently overlaps with the optical mode. Efficient tuning over more than 2π in phase can therefore be realized over relatively short device lengths of ~ 200 μm.

The main drawbacks of thermal tuning are: 1) its relatively high power dissipation (usually $\sim$25 mW); and 2) its low response speed, as tuning is realized by introducing and

depleting heat from the structure. Lower tuning powers on the order of ~ 1 mW have been realized by undercutting the waveguide [39], thereby improving the thermal isolation of the waveguide and minimizing heat loss to parasitic sinks such as the silicon substrate. However, as the structure traps heat more efficiently, the bandwidth is reduced even further. This makes thermal tuners ideal for devices on-chip that are tuned infrequently, but unsuitable for high-speed input and output of optical data.

2.7 Electrical-to-optical interfaces

End-to-end, photonic signal processing systems will require the ability to communicate with electronic microprocessors. Therefore, realizing photonic processors for computing will require high-speed interfaces between electrical and optical signals.

2.7.1 Modulators

Electrical-to-optical conversion requires **modulators** that "imprint" an electrical signal onto either the amplitude or phase of an optical field. In many platforms, such as lithium niobate, this modulation is performed using the Pockels effect, which produces an electric-field induced change in the refractive index n. This effect, which originates from the second-order nonlinear susceptibility $\chi^{(2)}$ and is extremely fast, enables direct transduction of voltage signals onto the optical field at datarates exceeding tens of Gbit/s.

Unfortunately, silicon is centrosymmetric, prohibiting second-order nonlinear interactions. Initially, the lack of a viable high-speed modulation mechanism greatly impeded progress in silicon as a photonic problem. In 1987, however, Soref and Bennett showed that injection or depletion of carriers in silicon will modulate the local refractive index [40].

This phenomenon, dubbed the plasma dispersion effect, is the basis of high-speed (GHz+) modulators implemented today in the silicon photonics platform. These devices are realized by transforming a silicon waveguide into a diode: by p- and n-doping regions of a waveguide, one can fabricate an effective p-i-n diode in a waveguide that can produce large changes in carrier concentration with relatively low applied biases at high speeds [30]. The mechanism, however, is not ideal; an injection of carriers will also incur optical loss due to free carrier absorption, and thus phase modulation will also induce amplitude chirp [40].

These modulators can be operated in **injection** where the diode is forward biased. This can produce remarkably strong modulation, as the carrier concentration in the waveguide can increase exponentially. The main drawback of this strategy is the strong amplitude modulation coupled with the phase, due to free carrier absorption. Moreover, injection modulators are typically limited to < 1 GHz bandwidth, as the reset time is limited by carrier recombination lifetimes in silicon, which are about 1 ns [41].

Alternatively, these modulators can be operated in **depletion** mode, where the diode is driven in reverse bias. Here, increasing the reverse bias widens the depletion region of the junction, removing carriers from the waveguide and inducing an index change. This is the preferred mode for high-speed modulators used in telecommunications, as the

bandwidth is limited by the *RC*-time constant of the junction, which can exceed 10 GHz. While there is still some residual amplitude modulation, it is far less than in injection mode as the waveguide is largely depleted of carriers. The tradeoff is that as there are few carriers to begin with in the depletion region, the modulation efficiency is relatively weak, although it is still comparable to Pockels modulators. These modulators have been successfully used to demonstrate high-speed, traveling-wave modulators suitable for use in optical transceivers [42].

2.7.2 Photodiodes

Efficient optical-to-electrical conversion is performed with high-speed photodiodes that absorb the optical field to generate a photocurrent. Fortunately, most CMOS foundries incorporate germanium into the process flow, enabling the implementation of hybrid SiGe photodiodes on chip.

Typically, these devices are realized by epitaxially growing germanium onto a silicon waveguide to realize a *p-i-n* photodiode. The optical field is absorbed within the germanium, generating a photocurrent that is collected by the metal contacts. As the photodetector is waveguide integrated, the quantum efficiency of these devices is usually quite high (over 80%). The small footprint of these detectors further minimizes *RC*-delay, with some devices reporting transit-time limited bandwidths on the order of tens of GHz.

Consequently, photodetectors today in silicon photonics perform exceptionally well and can realize nearly unity photon-to-electron conversion. Research in these devices today mostly focuses on improving the bandwidth to support higher datarate communications[1], lowering dark current[2], and improving power handling[3].

2.8 Mach-Zehnder interferometers

Having discussed the key elements of the silicon photonic platform, we now proceed to motivating the development of programmable photonic systems. The fundamental unit cell of these systems is the Mach-Zehnder interferometer, which comprises a pair of 2×2 couplers and a voltage-controllable phase shifter, such as a thermal tuner.

Figure 2-6 shows a Mach-Zehnder interferometer on the silicon platform. It consists of two 50-50 splitters, realized with directional couplers, and two electrically-contacted phase shifters–one on the input (conventionally referred to as ϕ) and one between the two couplers (conventionally referred to as θ).

This device is a reconfigurable beamsplitter that performs the 2×2 operation:

$$T_{ij}(\theta, \phi) = \frac{1}{2}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}\begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}\begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix}$$

[1] In 2021, IHP reported the realization of photodiodes in their silicon photonic platform with *265 GHz* bandwidth [43].

[2] The dark current of SiGe photodiodes is notoriously poor, largely originating from defects at the epitaxial interface.

[3] At high optical powers, charge screening in the germanium can deteriorate responsivity. Evanescently coupling the optical mode to a germanium slab has been shown to greatly reduce this problem [44].

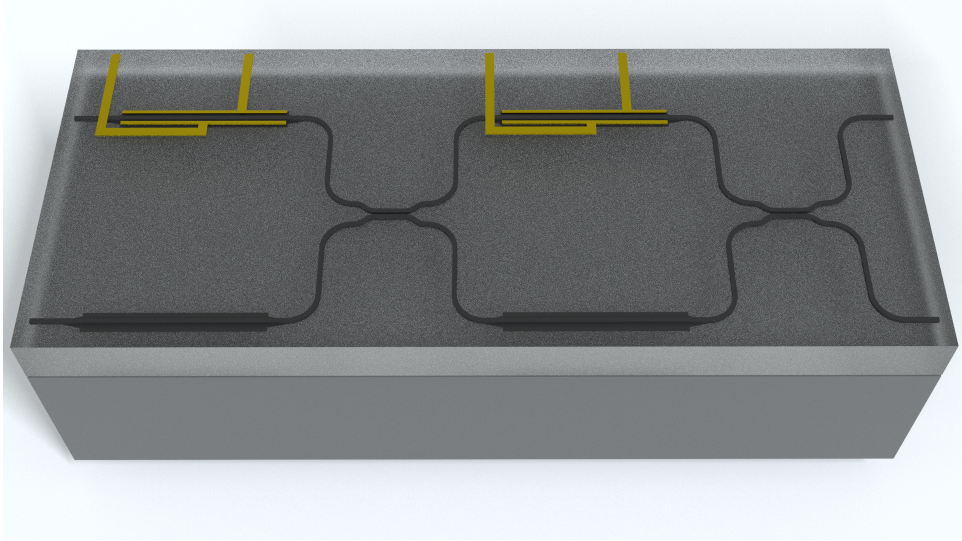

Figure 2-6: A Mach-Zehnder interferometer in a silicon photonic circuit. Each MZI consists of two 50-50 splitters, implemented using a directional coupler, and two electrically-programmable thermal phase shifters.

$$= ie^{i\theta/2} \begin{bmatrix} e^{i\phi}\sin(\theta/2) & \cos(\theta/2) \\ e^{i\phi}\cos(\theta/2) & -\sin(\theta/2) \end{bmatrix}$$

where θ, ϕ are single-mode phase shifts on the top arm. When inputting optical power into a single input, the output power in each port is:

$$P_{\text{cross}} = \cos^2(\theta/2) \tag{2.30}$$

$$P_{\text{bar}} = \sin^2(\theta/2) \tag{2.31}$$

Thus, an MZI can realize arbitrary splitting of an optical input by programming θ. However, more generally, this device can implement coherent matrix processing on optical fields; specifically, by tuning θ and ϕ electrically, one can realize any arbitrary 2×2 unitary operation on a pair of optical modes.

2.9 Towards fully-reconfigurable photonics

The vast majority of photonic circuits today are fabricated for a specific application–often for optical communications. Recently, however, interest has grown in a special class of photonic circuits that are flexibly programmable post-fabrication to implement different functions. This flexible control is realized through electrically tunable beamsplitters, i.e. a Mach-Zehnder interferometer unit cell [29]. In these circuits, each MZI functions effectively as an analog 2×2 gate on optical fields, and they can be tiled into larger networks to realize complex operations on a set of optical modes $a_0, a_1, ..., a_N$.

Figure 2-7 shows two implementations of universal, feedforward programmable photonic meshes. Feedforward circuits, where light propagates only in a single direction in

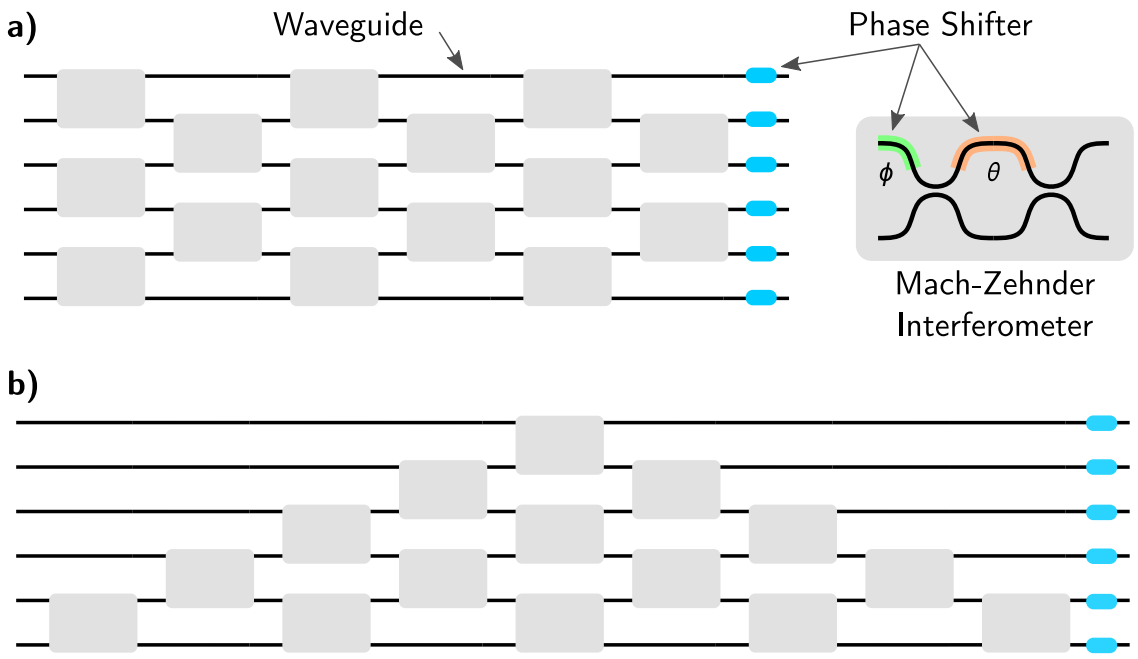

Figure 2-7: Universal architectures for programmable photonic matrix processors. Rectangular (a) and triangular (b) configurations of MZIs can be used to implement arbitrary unitary operations on optical modes with the Clements [45] and Reck [46] decompositions, respectively. Each MZI, consisting of an internal phase shift θ and an external phase shift ϕ, can perform an arbitrary 2×2 operation on a pair of optical fields.

the mesh, can implement arbitrary matrix-vector operations directly on the optical field. This is essential for quantum information processing in the photon basis [47, 25, 26], real-time mode demultiplexing (for instance from a multimode fiber) [48], and reconfigurable network switches.

Reck and Zeilinger in 1994 [46] showed that any arbitrary unitary operation can be realized on a set of optical fields through the triangular arrangement of reconfigurable beamsplitters shown in Figure 2-7a. This kicked off a decades-long effort within the optical community to realize such a system. However, stabilizing such a system on an optical benchtop proved to be extremely challenging, and it was not until 2014 that Carolan et al. [47] realized the first fully reconfigurable mesh on a photonic integrated circuit.

This landmark demonstration sparked a resurgence of interest in the notion of fully programmable optics on chip. Within the next few years, there were two more landmark results that established the importance of programmable integrated optics:

- In 2016, Clements et al. [45] showed that reconfigurable unitary operations could also be realized in a rectangular network of MZIs as shown in Figure 2-7a. A major practical impediment to the Reck architecture is that the triangular configuration is not compact (a significant drawback when chip area is expensive) and that different paths through the circuit travel through variable numbers of MZIs, and therefore incur variable loss (deteriorating the fidelity of the unitary realized). The rectangular

configuration reported by Clements demonstrated a far more scalable and practical approach to realizing these systems.

- In 2017, Shen and Harris et al. [15] demonstrated the first coherent photonic integrated circuit for matrix-vector products in DNNs. This system, which computed the linear weighting layers for a deep neural network through passive interference in optics, demonstrated the potential for photonics to accelerate next-generation systems for artificial intelligence.

Progress in the field has rapidly advanced since then. Experimental demonstrations of these circuits have already shown working systems that operate on a few modes, which have been used to accelerate tasks in quantum simulation [24, 49, 26, 25], mode unscrambling [48, 50] and combinatorial optimization [51].

Scaling up the next generation of these systems, however, will require addressing new challenges in error correction, packaging, algorithms, calibration, and control. In this thesis, I discuss my work to address some of these questions.

3

Error correction for programmable photonics

This chapter is adapted from work[1] *published in ref. [58].*

3.1 A manufacturing problem

In the previous chapter, we outlined why there is such massive interest in programmable integrated photonic systems. The field programmable gate array, or FPGA, enabled the advent of post-fabrication, reconfigurable digital electronics. Seminal papers by Reck [46], Clements [45], and Miller [59], which showed how to realize arbitrary linear operations in optics, introduced the possibility of realizing a similarly reconfigurable "optical FPGA" on chip.

There is, however, a critical difference between a programmable photonic processor and an FPGA: photonic systems are analog. Naturally, any practical use of such a system will require scaling them to hundreds or thousands of modes. State-of-the-art digital ASICs for machine learning will compute at 8 bits of accuracy; emulating similar accuracies in photonics, however, will require precise fabrication of tens of thousands of optical interferometers. This is incredibly challenging to do at scale, especially with volume manufacturing techniques such as photolithography. Realistically, due to variations in wafer thickness, etch depth, and myriad other process parameters, the performance of a photonic component will differ across a chip within some range of variability. In electronic circuits, this variability is not a problem as these circuits are digital and we can make use of digital error correction; in a photonic system, however, static component errors induced by process variation will introduce errors that rapidly add up. The decomposition [46, 45] algorithms for these circuits assume that all of the components are ideal; thus, any component errors result in a programming of the wrong operation, rendering the system useless.

[1] This work was conducted in close collaboration with Dr. Ryan Hamerly, a visiting scientist from NTT Research. The ideas developed in this paper, published in *Optica*, sparked a highly productive research effort in our group to develop error correction algorithms for programmable optics, leading to two patent applications [52, 53], many conference talks, and further publications in *Physical Review Applied* [54, 55], *Nanophotonics* [56], and *Nature Communications* [57].

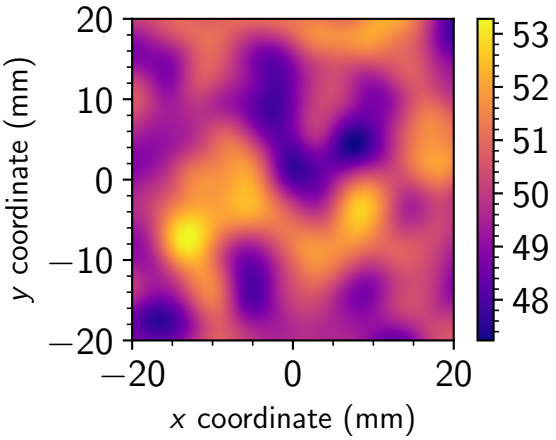

Figure 3-1: Simulated variation of the splitting behavior of a directional coupler across a 40x40 mm^2 reticle. An ideal splitter is 50-50; there are some local correlations, but across a reticle the splitting can vary quite significantly. Here, we assume the splitting varies with a standard deviation $\sigma = 2.1\%$ and a correlation length of 5 mm. We obtain these results using the procedure outlined in ref. [65] for simulating layout-dependent correlations in manufacturing.

Component imprecision therefore has serious implications for the future of these systems. The community was quickly discovering this early in my PhD; for example, beamsplitter variation as small as 2%, which is a typical wafer-level variance [60], was found in simulation to degrade accuracy by nearly 50% for a relatively simple image classifier implemented on a photonic circuit [61]. This is not a problem isolated to neuromorphic systems; other photonic signal processing architectures are similarly susceptible. For example, recirculating lattices of MZIs are being pursued for implementing RF signal processing in the optical domain [62, 63, 28]. These devices were similarly found to be extremely sensitive to manufacturing error [64].

Thus, scaling these systems up to commercially relevant sizes will become an increasingly intractable control problem. When we started working on this, a few proposals existed for confronting this issue with numerical optimization. Some early work focused on using gradient-free, global optimizers [66, 67, 68, 69], including some work from our own group [70]. These optimizers, however, converge slowly, and their runtimes rapidly scale with the number of parameters. Gradient descent can be employed to efficiently direct this process, as proposed in [71]; unfortunately, there is no native way to compute these gradients in hardware and thus the method is mainly limited to optimizing simulated devices.

The most promising optimization approaches employ progressive algorithms making use of local feedback [72, 73]. Here, the idea is that by using a well-chosen set of optical input fields, a photonic processor can be programmed, or "self-configured," to implement an arbitrary operation by locally optimzing one MZI at a time. This is a neat and elegant solution to programming a photonic circuit with high accuracy, as it reduces an $2N^2$-dimensional optimization problem to a sequence of $2N^2$ single variable optimizations. The problem, however, is that this method requires the ability to measure the optical

power *at every point in the chip*; i.e., every MZI on chip will require a tap photodiode with accompanying readout and feedback electronics.

For a mesh network, this resource overhead scales as $O(N^2)$, greatly increasing the number of electrical lines and overall power consumption of the system. As a result, there have been relatively few experimental demonstrations of such a system [50]. Moreover, such a procedure will limit the reconfiguration speed of the circuit.

During the early stage of my PhD, we began to think about ways we could correct component errors in these devices deterministically[2]. In many optical architectures for information processing, error correction can become intractable as device errors will add in quadrature and quickly compound. However, programmable circuits of Mach-Zehnder interferometers turn out to be quite special among photonic computing architectures–in particular, they possess certain unique attributes that present an opportunity to mitigate device errors:

1. It is already known how to obtain a desired unitary operation in these systems deterministically assuming ideal components. If a unitary operation is realizable by an imperfect photonic circuit, it should not require optimization to deduce the required settings; rather, *a small perturbation in the device behavior due to component deviation should translate directly to a small perturbation in the interferometer's phase settings to recover the original unitary.*

2. A programmable photonic circuit is composed of a discrete set of 2x2 MZIs, *which function effectively as optical "gates."* The self-configuration scheme [74, 59] discussed earlier cleverly leverages this fact to reduce the task of programming to optimizing one MZI at a time. An error correction algorithm should also take advantage of this feature of the architecture–as the scale of these systems grow, we should still be able to apply corrections gate-by-gate, even if the total number of hardware parameters grow nonlinearly.

3. The computation is *coherent*. Most photonic computing architectures are incoherent, and thus errors add in quadrature. Coherent architectures for computing, however, introduce the opportunity to *undo errors coherently as they arise in the circuit.*

In this chapter, we develop a local approach that corrects hardware errors one at a time within each optical gate composing the circuit. Our approach, which was the first deterministic error correction algorithm for photonic processors, outperformed previous approaches in several key respects:

1. It is flexible, requiring only a one time device calibration to directly compute the hardware settings for any given unitary;

2. For sufficiently low hardware errors the computed settings yield the exact unitary desired;

[2]I would like to thank Professor David A. B. Miller and Dr. Sunil Pai, both from Stanford University, for many valuable discussions on correcting errors in these systems.

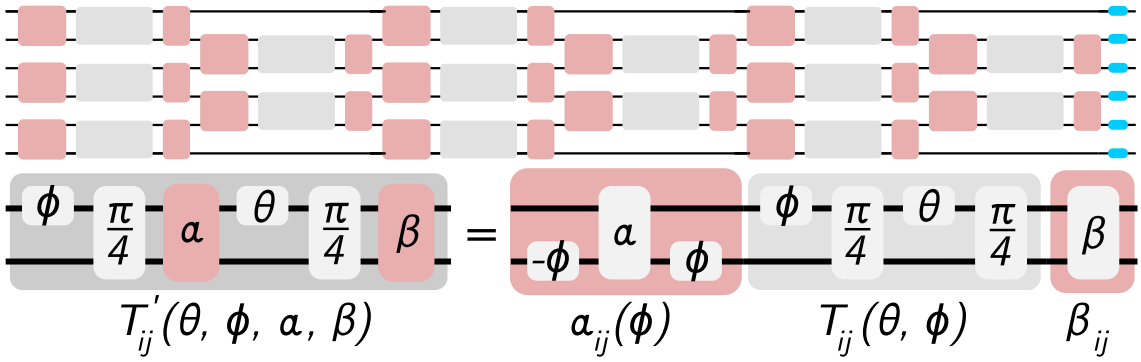

Figure 3-2: While we typically model the MZI as an ideal device when developing algorithms for photonic computation, in practice manufacturing variations α, β in the directional couplers produce errors in the gate operation. The effect of these hardware errors is to left- and right-multiply each programmable 2×2 unitary $T_{ij}(\theta, \phi)$ implemented by an MZI by error matrices $\beta_{ij}, \alpha_{ij}(\phi)$. Applying the standard decomposition for ideal components to these imperfect optical gates will not produce the correct gate operation.

3. Our approach requires minimal overhead and does not make use of additional interferometers or internal detectors within every device.

Moreover, unlike other algorithms, it is the only one that does not assume any particular structure to the circuit and can be generalized to any programmable architecture making use of interferometers. Thus, it is relevant to many architectures being considered for optical information processing, including feedforward circuits for neuromorphic computing and recirculating waveguide meshes for RF photonic signal processing.

3.2 Hardware Error Correction

The central idea here is that if we know all of the relevant component parameters, we can coherently undo their effect on the circuit "locally," i.e. one MZI at a time. Local error correction thereofre requires characterization of each phase shifter and passive splitter in the photonic circuit. The calibration is performed once with the results stored in a lookup table; any arbitrary function can then be programmed by computing the settings for an ideal set of MZIs and converting them, one by one, to the corresponding settings for an imperfect device. In the Supplementary Information of ref. [58], we developed a protocol to calibrate these errors using detectors only at the circuit outputs. Assuming these parameters are known, we can then deterministically correct circuit errors.

As discussed in the previous chapter, the fundamental optical unit cell of a programmable photonic circuit is a 2×2 Mach-Zehnder interferometer (MZI) composed of an external phase shifter on one input, two 50-50 beamsplitters, and an internal phase shifter on one of the modes between the splitters. This device is an electrically programmable beamsplitter capable of performing a 2×2 unitary operation $T_{ij}(\theta, \phi)$ on optical modes i, j parameterized by the external phase shift ϕ and the internal phase shift θ.

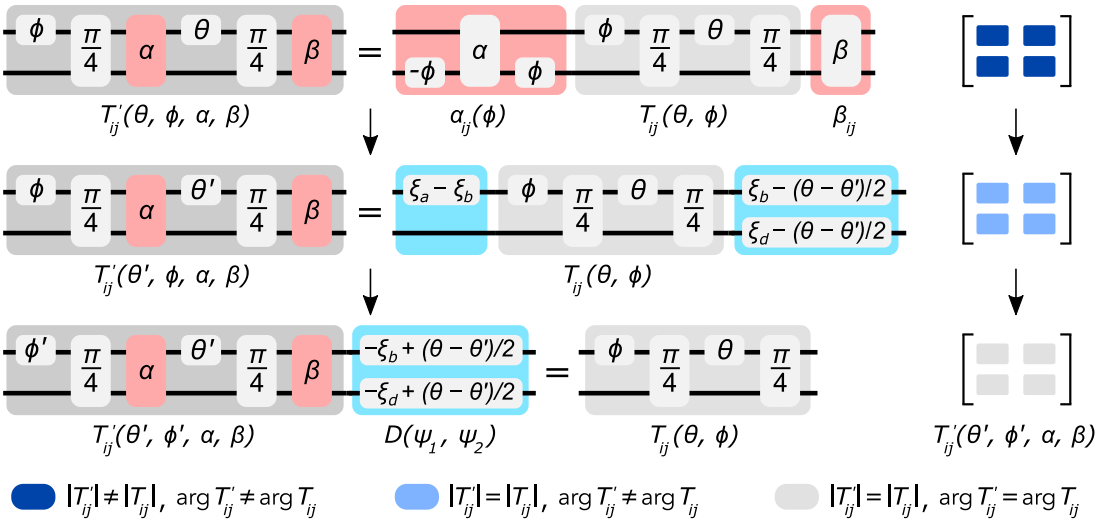

Figure 3-3: Fabrication-induced errors within each MZI can be corrected by applying local corrections $\theta \to \theta'$, $\phi \to \phi'$ to the device. We first correct θ to set the magnitudes of the elements of T_{ij} equal to T'_{ij}. Once the amplitude terms are set correctly, we apply phase corrections to the input and outputs of the device to correct phase errors between T_{ij} and T'_{ij}.

On an integrated photonics platform, the 50-50 splitters can be realized by a directional coupler or multimode interferometer (MMI); the operation of these splitters can be described by a 2×2 matrix:

$$\begin{bmatrix} \cos(\pi/4 + \alpha) & i\sin(\pi/4 + \alpha) \\ i\sin(\pi/4 + \alpha) & \cos(\pi/4 + \alpha) \end{bmatrix} \tag{3.1}$$

where α describes the deviation from an ideal 50-50 splitting behavior. For an ideal splitter $\alpha = 0$, this matrix reduces to:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \tag{3.2}$$

The overall operation $T_{ij}(\theta, \phi)$ performed by a single ideal MZI is therefore:

$$\begin{aligned} T_{ij}(\theta, \phi) &= \frac{1}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \\ &= ie^{i\theta/2} \begin{bmatrix} e^{i\phi}\sin(\theta/2) & \cos(\theta/2) \\ e^{i\phi}\cos(\theta/2) & -\sin(\theta/2) \end{bmatrix} \end{aligned}$$

where θ, ϕ are single-mode phase shifts on the top arm.

Higher dimensional matrix operations can be implemented with this unit cell by applying the Clements [45] and Reck [46] decompositions (Fig. 2-7). These algorithms decompose an arbitrary N-dimensional unitary U into a product of $N(N-1)/2$ two-dimensional unitaries computed by interference between nearest-neighbor optical modes,

followed by phase shifts on the output modes corresponding to a diagonal matrix D:

$$U = D \prod T_{ij}(\theta, \phi) \tag{3.3}$$

So far, we have spoken in abstract terms of a set of "perfect" photonic components. Now, however, let's examine the impact of fabrication error on these systems. If the MZI has imperfect splitters with errors α, β, the operation of the MZI must now be parameterized with four variables $T'_{ij}(\theta, \phi, \alpha, \beta)$ (Fig. 3-2):

$$ie^{i\theta/2} \begin{bmatrix} e^{i\phi}(\cos(\alpha-\beta)\sin(\theta/2) + & \cos(\alpha+\beta)\cos(\theta/2) + \\ \quad i\sin(\alpha+\beta)\cos(\theta/2)) & \quad i\sin(\alpha-\beta)\sin(\theta/2) \\ e^{i\phi}(\cos(\alpha+\beta)\cos(\theta/2) - & -\cos(\alpha-\beta)\sin(\theta/2) + \\ \quad i\sin(\alpha-\beta)\sin(\theta/2)) & \quad i\sin(\alpha+\beta)\cos(\theta/2) \end{bmatrix} \tag{3.4}$$

$$= \begin{bmatrix} \cos\beta & i\sin\beta \\ i\sin\beta & \cos\beta \end{bmatrix} \hat{T}(\theta, \phi) \begin{bmatrix} \cos\alpha & ie^{-i\phi}\sin\alpha \\ ie^{i\phi}\sin\alpha & \cos\alpha \end{bmatrix} \tag{3.5}$$

In the limit $\alpha, \beta \to 0$, the second term of each entry in the matrix $T'_{ij}(\theta, \phi, \alpha, \beta)$ drops out and we recover the expected transformation for an ideal device. It should be no surprise that implementing the usual decomposition on these imperfect devices will not yield the desired unitary:

$$D \prod T'_{ij}(\theta, \phi, \alpha, \beta) \neq D \prod T_{ij}(\theta, \phi) \tag{3.6}$$

Notice, however, that the impact of errors α and β is to perform coherent rotations of the optical state relative to the desired programming. If we wanted to correct these errors, one could in principle program $\hat{T}(\theta, \phi)$ taking these coherent errors into account. To program into an imperfect circuit a desired unitary $U = \prod T_{ij}(\theta, \phi)$, we apply local corrections $\theta \to \theta', \phi \to \phi'$ to each device such that $T'_{ij}(\theta', \phi', \alpha, \beta) = T_{ij}(\theta, \phi)$.

Our approach is illustrated in Figure 3-3. We begin by finding θ' such that the magnitudes of the entries of $T'_{ij}(\theta', \phi', \alpha, \beta)$ equal those of $T_{ij}(\theta, \phi)$. The correction $\theta \to \theta'$ can be derived by requiring that the magnitude of the upper left entry of $T'_{ij}(\theta', \phi', \alpha, \beta)$ equal that of $T_{ij}(\theta, \phi)$. For a 2×2 unitary matrix U, the unitarity condition $UU^\dagger = I$ implies that setting the magnitudes of one term in both matrices to be equal is sufficient to set the magnitudes of all terms in the matrices to be equal. This condition produces an expression relating θ' to θ:

$$\cos^2(\alpha-\beta)\sin^2(\theta'/2) + \sin^2(\alpha+\beta)\cos^2(\theta'/2) = \sin^2(\theta/2) \tag{3.7}$$

Solving for θ', we find that:

$$\sin^2(\theta'/2) = \frac{\sin^2(\theta/2) - \sin^2(\alpha+\beta)}{\cos^2(\alpha-\beta) - \sin^2(\alpha+\beta)} \tag{3.8}$$

$$\theta' = 2\arcsin\sqrt{\frac{\sin^2(\theta/2) - \sin^2(\alpha+\beta)}{\cos^2(\alpha-\beta) - \sin^2(\alpha+\beta)}} \tag{3.9}$$

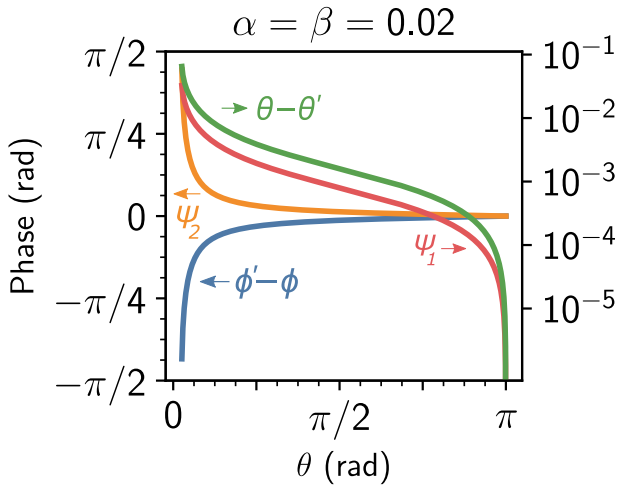

Figure 3-4: The corrections $\phi'-\phi$, $\theta-\theta'$, ψ_1, ψ_2 applied to an MZI with two 52-48 beamsplitters ($\alpha = \beta = 0.02$). The arrows on the plot indicate which vertical axis each curve corresponds to.

Since α, β are small, the denominator of the expression for θ' will always be positive. This expression therefore has a solution only when the numerator is positive, i.e. $\sin^2(\theta/2) > \sin^2(\alpha+\beta)$, and when the argument in the arcsin function is less than 1, i.e. $\sin^2\theta/2 - \sin^2(\alpha+\beta) < \cos^2(\alpha-\beta) - \sin^2(\alpha+\beta)$. These conditions yield the range over which θ is physically realizable:

$$2|\alpha+\beta| < \theta < \pi - 2|\alpha-\beta| \tag{3.10}$$

If the matrix decomposition requires θ outside this range, we can minimize the error by setting $\theta' = 0$ (if $\theta < 2|\alpha+\beta|$) or $\theta' = \pi$ (if $\theta > \pi - 2|\alpha-\beta|$).

Assuming we can physically implement the required value of θ', the magnitudes of the elements of $T'_{ij}(\theta', \phi', \alpha, \beta)$ and $T_{ij}(\theta, \phi)$ are now the same, but each element of T'_{ij} will have an undesired extraneous phase $\xi_a, \xi_b, \xi_c, \xi_d$ relative to the corresponding term in T_{ij} that must be corrected. We can therefore rewrite $T'_{ij}(\theta', \phi', \alpha, \beta)$ as

$$T'_{ij} = ie^{i\theta'/2}\begin{bmatrix} e^{i\phi'}e^{i\xi_a}\sin(\theta/2) & e^{i\xi_b}\cos(\theta/2) \\ e^{i\phi'}e^{i\xi_c}\cos(\theta/2) & -e^{i\xi_d}\sin(\theta/2) \end{bmatrix} \tag{3.11}$$

$$= ie^{i\theta'/2}\begin{bmatrix} e^{i\xi_b} & 0 \\ 0 & e^{i\xi_d} \end{bmatrix}\begin{bmatrix} e^{i(\phi'+\xi_a-\xi_b)}\sin(\theta/2) & \cos(\theta/2) \\ e^{i(\phi'+\xi_a-\xi_b)}\cos(\theta/2) & -\sin(\theta/2) \end{bmatrix} \tag{3.12}$$

where the simplification in the second line originates from unitarity requiring that $\xi_a + \xi_d = \xi_b + \xi_c$. We correct the phase errors in T'_{ij} by setting $\phi' = \phi + \xi_b - \xi_a$ and by applying additional phases $\psi_1 = -\xi_b + (\theta - \theta')/2$, $\psi_2 = -\xi_d + (\theta - \theta')/2$ to the top and bottom output modes, respectively. Applying these corrections will set $T'_{ij}(\theta', \phi', \alpha, \beta)$ exactly equal to $T_{ij}(\theta, \phi)$.

Expressions for the phase errors ξ_a, ξ_b, ξ_d can be constructed by setting the complex arguments of the elements of T_{ij} equal to those of $T'_{ij}(\theta', \phi', \alpha, \beta)$. From this, we find

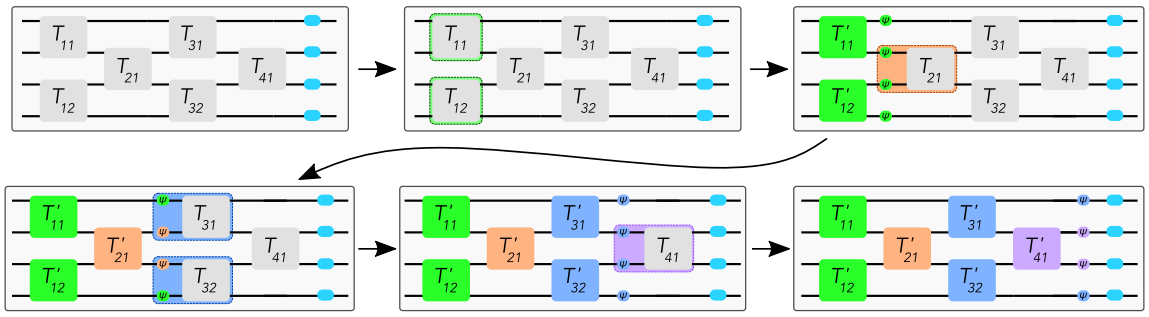

Figure 3-5: The procedure for programming a unitary with hardware errors on a 4×4 rectangular unitary circuit. We first program each MZI to the (θ, ϕ) setting obtained with the standard decomposition in [45]. Each MZI is then converted $T_{ij} \rightarrow T'_{ij}$ to the settings for an imperfect device one column at a time. At each step we propagate the output phase shifts ψ_1, ψ_2 forward in the circuit until the entire network is corrected.

that:

$$\phi' = \phi + \arctan\left[\frac{\sin(\alpha-\beta)}{\cos(\alpha+\beta)}\tan(\theta'/2)\right] \tag{3.13}$$

$$- \arctan\left[\frac{\sin(\alpha+\beta)}{\cos(\alpha-\beta)}\cot(\theta'/2)\right] \tag{3.14}$$

$$\psi_1 = -\arctan\left[\frac{\sin(\alpha-\beta)}{\cos(\alpha+\beta)}\tan(\theta'/2)\right] + (\theta-\theta')/2 \tag{3.15}$$

$$\psi_2 = \arctan\left[\frac{\sin(\alpha+\beta)}{\cos(\alpha-\beta)}\cot(\theta'/2)\right] + (\theta-\theta')/2 \tag{3.16}$$

The errors $\theta-\theta', \phi'-\phi, \psi_1, \psi_2$ as a function of θ for an example MZI with two 52-48 ($\alpha = \beta = 0.02$) splitters are shown in Figure 3-4. While the corrections to θ and ψ_1 are small ($\sim$ 0.1 rad), the errors for ϕ and ψ_2 are quite substantial. In particular, for low device reflectivities ($\theta \approx 0$), the phase corrections required can exceed 1 rad.

Generally, we cannot apply the auxiliary phases ψ_1, ψ_2 locally to the device being corrected, since the output modes do not have phase shifters. In most cases, one of the two can be incorporated into the external phase shifter setting of an MZI in the subsequent column. The other phase can be applied by observing that:

$$T_{ij}(\theta,\phi)\begin{bmatrix} e^{i\psi_1} & 0 \\ 0 & e^{i\psi_2} \end{bmatrix} = \begin{bmatrix} e^{i\psi_2} & 0 \\ 0 & e^{i\psi_2} \end{bmatrix} T_{ij}(\theta, \phi+\psi_1-\psi_2) \tag{3.17}$$

Using this fact, we can propagate the auxiliary phases forward, through all of the columns of the network, out to the phase shifters D located on the output modes of the circuit. This procedure, illustrated in Figure 3-5, produces a modified output phase screen D' such that:

$$U = D\prod T_{ij}(\theta,\phi) = D'\prod T'_{ij}(\theta',\phi',\alpha,\beta) \tag{3.18}$$

Depending on the component imperfections and the required value of θ, we may also be able to program θ' such that $|T'_{ij}(\theta', \phi', \alpha, \beta)| = |T_{ij}(\theta, \phi)|$ if the condition in equation (3.10) is satisfied. If every MZI in the circuit satisfies the condition in equation (3.10), we can recover the exact unitary desired. However, if some MZIs in the circuit cannot realize the required splitting, that exact unitary is not physically realizable by the device. In this case, correcting the phases ϕ', ψ_1, ψ_2 and setting θ' as close to the required value as possible minimizes the gate error $\| T_{ij} - T'_{ij} \|$.

We can summarize the algorithm for programming of a matrix U as follows:

1. Calibrate all phase shifters and splitter errors α, β and store in lookup table.

2. Calculate the required values for θ, ϕ assuming ideal components, using the procedure described by Reck [46] or Clements [45].

3. For each device, set $\theta \to \theta'$ using the expression in equation (3.9). If $\theta < 2|\alpha + \beta|$, set $\theta' = 0$; if $\theta > \pi - 2|\alpha - \beta|$, set $\theta' = \pi$.

4. Apply phase corrections ϕ', ψ_1, ψ_2 as given in equations 3.14-3.16. Propagate ψ_1, ψ_2 forward to the output phase screen D with the expression in equation (3.17).

3.3 What we've learned so far

In the last section, we developed an algorithm that enables high-accuracy (potentially zero error) matrix computation on photonic hardware. The derivation presented appears fairly straightforward; however, it masks what is actually a rather surprising finding.

Recall that this is a system of noisy photonic components we are attempting to perform computation on. In a run-of-the-mill photonic processor, these errors add up in quadrature and the result will be useless. One would expect that this architecture would be no different; however, because the entire computation (and the main sources of error) are coherent, we can coherently undo errors as they arise! Moreover, because the circuit structure is composed of discrete, optical "gates," i.e. 2×2 MZIs, errors can always be corrected "locally", making the problem of error correction far more manageable as circuit sizes scale up.

We have illustrated this procedure for the example of feedforward unitary circuits, but the same principles apply for other architectures. Each optical gate within any programmable circuit can be corrected to the required 2×2 unitary operation T_{ij} with the aforementioned procedure. The expressions provided assume a specific form for the MZI (Fig. 2-7), but they can be easily modified to apply to other designs. For example, recirculating architectures often use the dual-drive tunable basic unit (TBU), which has two individually controllable phase shifters on the arms of the MZI [75]; the expressions above can be applied to such an architecture with some simple modifications.

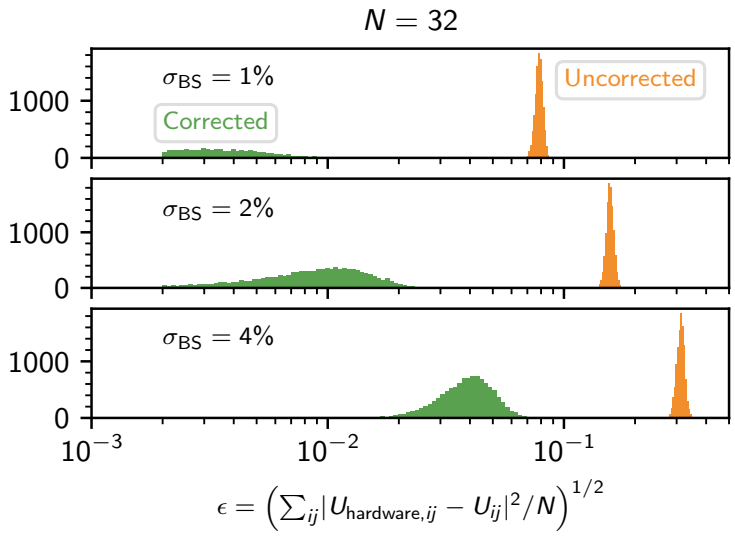

Figure 3-6: Matrix error ϵ before and after correction for 100 random unitaries implemented on 100 random circuits with varying beamsplitter statistics.

3.4 Hardware Performance

The performance of this algorithm was benchmarked using a custom simulation package for programmable photonic circuits with fabrication imperfections. This package is written in Python, mainly using `NumPy` [76], but performance-critical sections are compiled to C using Cython. This package started as a relatively simple script I wrote at the start of graduate school to help me gain intuition with these circuits. Over the course of the project, however, it rapidly expanded in complexity to account for myriad practical issues with these chips, including beamsplitter errors, device losses, variable photodiode responsivities, thermal crosstalk, cross-wafer correlations originating from errors in lithography, and quantization error from finite DAC resolution.

All of the results in this work were produced using Monte Carlo simulations of photonic circuits with random component errors. While our original work neglected the role of loss, in the next section I describe our findings when we account for non-uniform loss through the chip.

Figure 3-6 shows the matrix error $\epsilon = (\sum_{ij} |U_{\text{hardware},ij} - U_{ij}|^2 / N)^{1/2}$ for 100 Haar random unitaries implemented on 100 randomly generated $N = 32$-mode unitary circuits with mean beamsplitter transmission $\eta = (50 \pm \sigma_{BS})\%$. This metric, known as the Frobenius norm, can be interpreted as an average relative error per entry of the matrix U. This is particularly relevant for a system implementing a neural network, as it would correspond to the average relative error per weight.

The beamsplitter errors in these simulations are independently sampled from a Gaussian distribution, although we found that for large N, the distribution shape will not greatly affect the results. Remarkably, error correction reduces ϵ significantly, sometimes by more than an order of magnitude. This improvement is larger for circuits with small splitting errors, as they are more likely to satisfy equation (3.10) and program the required θ' for

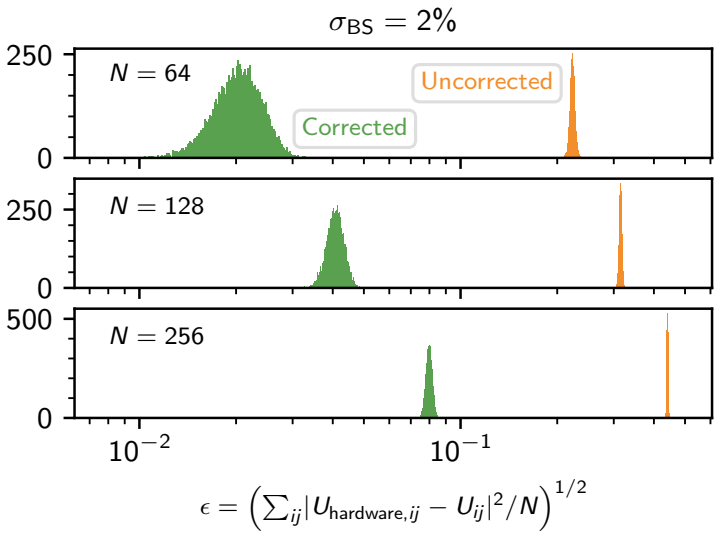

Figure 3-7: Matrix error ϵ before and after correction for $N = \{64, 128, 256\}$ with a beamsplitter variation $\sigma_{BS} = 2\%$.

all devices within the circuit. However, even for circuits with large σ_{BS}, where many MZIs may not be programmable to the required θ, the improvement in ϵ is substantial as all errors in ϕ, ψ_1, ψ_2 can always be corrected.

In Figure 3-7, we show ϵ with and without error correction for circuit sizes $N = \{64, 128, 256\}$. For these simulations, we chose a beamsplitter variation of $\sigma_{BS} = 2\%$, which is a typical wafer-level variance [60]. While the improvement in ϵ diminishes for larger N, we still find substantial improvement gained in our approach for up to 256 modes.

Why does the improvement in ϵ diminish with larger N? It turns out this is related to the statistics of unitary circuits; for large unitary circuits most MZIs need to be programmed to reflectivities close to $\theta \approx 0$ [77]. Physically, this can be understood by considering that for an "average" unitary, inputting light into a single mode will equally distribute it to all N outputs. For light input into mode 1, the only way to reach output N is to cross the main diagonal, and so these MZIs need to be programmed close to the cross ($\theta = 0$) state. As the circuit grows larger, more and more devices in proximity to the main diagonal will need to be programmed to $\theta \approx 0$. Since the minimum θ realizable for a device is $|\alpha + \beta|$, as N increases a larger fraction of devices cannot be programmed to the required splitting. Even in this case, however, there is always some improvement in ϵ, as any phase errors introduced by the components can be corrected.

3.5 Can we correct errors in neuromorphic hardware?

While we have established that our protocol reduces the Frobenius norm of the error, this tells us very little about how it will work for a realistic application. For neuromorphic systems, the Frobenius norm corresponds to an average error per model parameter. However, it is not clear how errors in any given model parameter, or weight, will impact the

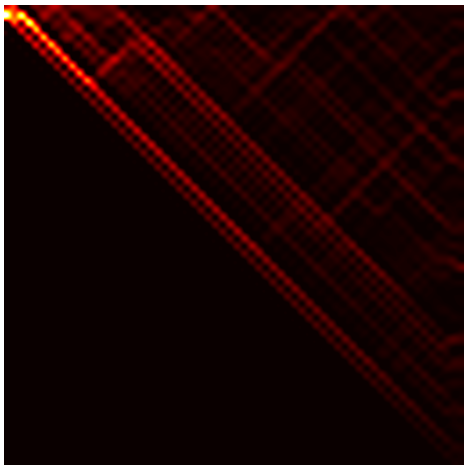

Figure 3-8: Trajectory of light when input into the top-most port of a 64×64 circuit programmed to a Haar random unitary. Note that optical power is equally distributed to all outputs, requiring the main diagonal to be programmed close to the cross ($\theta = 0$) state.

performance of a realistic photonic processors for neural networks.

In order to better understand this, we applied this approach to simulations of a two-layer photonic neural network performing image classification. The architecture of the neural network is similar to that studied in [15, 73, 61], where forward inference is optically computed through passive interference within a unitary photonic circuit coupled with an electrical or electro-optic nonlinearity [78].

The initial parameters for the optical neural networks were trained using the `Neurophox` package. Images of handwritten digits from the MNIST task are pre-processed with a Fourier transform and truncated to a $\sqrt{N} \times \sqrt{N}$ center window for a dimension N unitary circuit. We assume a fixed amount of optical power is available to the circuit; each input vector corresponding to an image is normalized to unit length, so that all images are encoded into the neural network with the same amount of optical power. This normalization can be realized optically with a diagonal line of MZIs, as depicted in Figure 3-9d.

The activation function is realized electro-optically with a tap photodiode coupled to a Mach-Zehnder modulator [78] (Fig. 3-9b). The activation function taps off 10% of the input power to the photodiode, while the remainder is directed to the modulator. The photocurrent drives the modulator through a transimpedance amplifier (TIA), resulting in a nonlinear modulation of the electric field.

The nonlinearity implements the activation function [78]:

$$f(E) = (\sqrt{1-\alpha})e^{-i(g|E|^2/2+\phi/2-\pi/2)}\cos(g|E|^2/2+\phi/2)E \tag{3.19}$$

where $\alpha = 0.1$ is the fractional power tapped off to the photodiode and $g = \pi/20$ is the modulator phase induced when 1 mW is incident upon the nonlinearity (prior to the tap). For typical electro-optic modulator drive voltages of < 8 V [80, 81] and a photodiode responsivity of 1 A/W [82], the required TIA gain for these parameters is roughly 36 dBΩ. The modulator is biased so that no transmission occurs when $E = 0$; as shown in

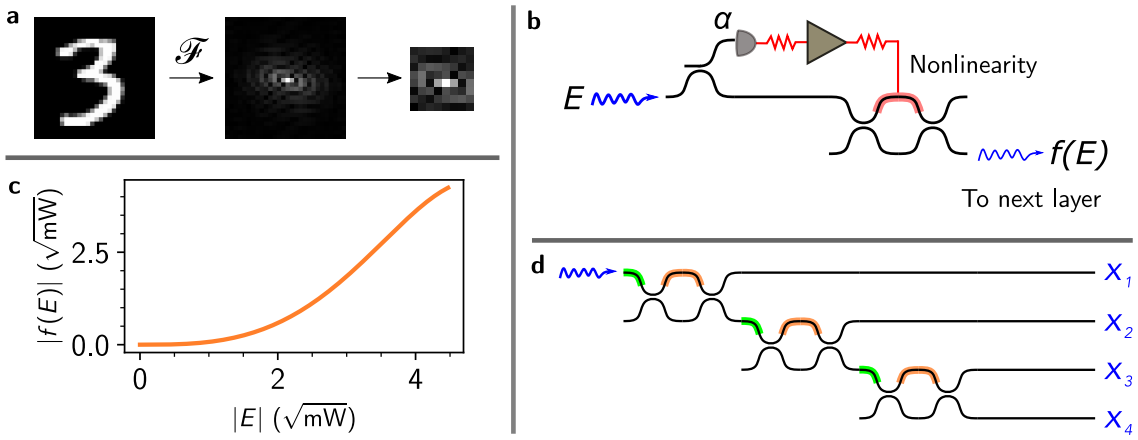

Figure 3-9: a) The MNIST data set was pre-processed with a Fourier transform and truncated to a $\sqrt{N} \times \sqrt{N}$ center window for a N-mode unitary circuit [73]. b) The activation function architecture as described in [78]. A small fraction α of the input signal is tapped off to a photodiode driving a Mach-Zehnder modulator. c) The activation function $f(E)$ for the parameters used in the simulation. Since the hidden layers operate on electric field amplitudes, we plot the square root of the optical power in units $\sqrt{\text{mW}}$. Technically, $f(E)$ is non-monotonic for high optical powers, as the Mach-Zehnder interferometer will produce a $\cos(|E|^2)$ modulation. However, the input optical powers in our simulations are chosen to ensure the activation function operates only in the modReLU-like region. d) The input vectors into the neural network were normalized to unit length, which can be realized optically with a diagonal line of MZIs.

Figure 3-9c, for optical powers < 20 mW $f(E)$ approximates a modReLU function [79].

As the network size N increases, the average power within a waveguide drops as $1/N$; for this reason, we assumed the total optical power input into the circuit increased commensurately to ensure the activation function could still be triggered. The $N = \{36, 64\}$ networks were trained with 20 mW of optical power, the $N = 144$ network was trained with 40 mW, and the $N = 256$ network was trained with 60 mW of optical power. All of the neural networks were trained to minimize the mean squared error between the L_2 normalized output power and the one hot encoding of the correct image.

Using the `Neurophox` package, we trained two-layer neural networks with up to 256 neurons per layer to recognize low-frequency Fourier features of handwritten digits from the MNIST task. Figure 3-11 shows the median classification accuracy for 300 randomly generated circuits as a function of the beamsplitter statistics $\eta = (50 \pm \sigma_{\text{BS}})\%$. The smaller circuits ($N = 36, 64$) exhibit roughly $95 - 96\%$ accuracy after training, while the larger circuits ($N = 144, 256$) exhibit a slightly higher model accuracy of $\sim 97\%$. The larger circuits, however, are less resilient to errors; without error correction classification accuracy drops to below 90% for all circuit sizes at a splitter variation as low as $\sim 3\%$.

Hardware error correction extends this cutoff to more than 6%, which is well beyond modern-day process tolerances [60]. Moreover, without correction classification accuracy drops significantly at even typical wafer-level variances (2%). However, with error correction there is almost no drop in accuracy at these variances and less than 1% accuracy loss for beamsplitter variations as high as 4%. We expect this margin for fabrication error

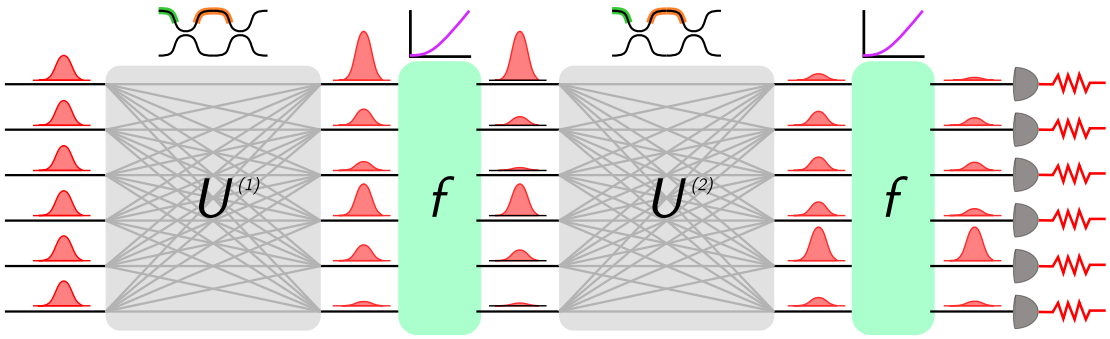

Figure 3-10: Architecture of the simulated two-layer optical neural network for the MNIST task. Matrix-vector products are calculated optically in the photonic circuit, and modReLU-like activation functions are implemented electro-optically [79, 78]. The output signal is photodetected and L_2 normalized to generate a quasi-probability distribution for the classification.

will prove important as optical neural networks scale up. These results suggest that error correction in programmable photonics can enable high-accuracy neural networks of up to hundreds of modes within current-day process tolerances.

3.6 Hardware error correction for photonic signal processing

While our analysis has focused on feedforward programmable photonic meshes, our results can also be applied to recirculating architectures useful in RF and optical signal processing. These recirculating meshes, which are usually configured in hexagonal or triangular lattices, enable implementation of finite impulse response (FIR) and infinite impulse response (IIR) filters by configuring waveguides into asymmetric MZIs and ring resonators, respectively [28, 63, 62]. Unlike the feedforward architectures, the programming of these structures usually cannot be determined analytically and must be found through optimization [67, 68, 69]. Since optimization can be time-consuming for complex systems, error correction can enable optimizing these circuit parameters on idealized models and then porting them over to hardware without retraining. As an example, we simulated the performance of an IIR filter functioning as a tunable dispersion compensator (TDC) on a hexagonal waveguide lattice [63]. TDC modules are of interest for numerous applications, including compensating chromatic dispersion in optical communication links [83] and enabling high-dimensional quantum key distribution (QKD) with temporal modes [84].

We implemented the TDC using an architecture similar to the tunable-coupling ring array described in [85]. Programmable dispersion is achieved by individually tuning the coupling and resonance of each ring in a chain of 15 resonators coupled serially to one another. Each ring is implemented with a single MZI (often referred to as the tunable basic unit, or TBU) in a hexagonal mesh acting as the coupler, while five other TBUs are programmed to the bar state to implement feedback. For simplicity we do not simulate routing within the hexagonal mesh, but instead simulate the transfer function of each

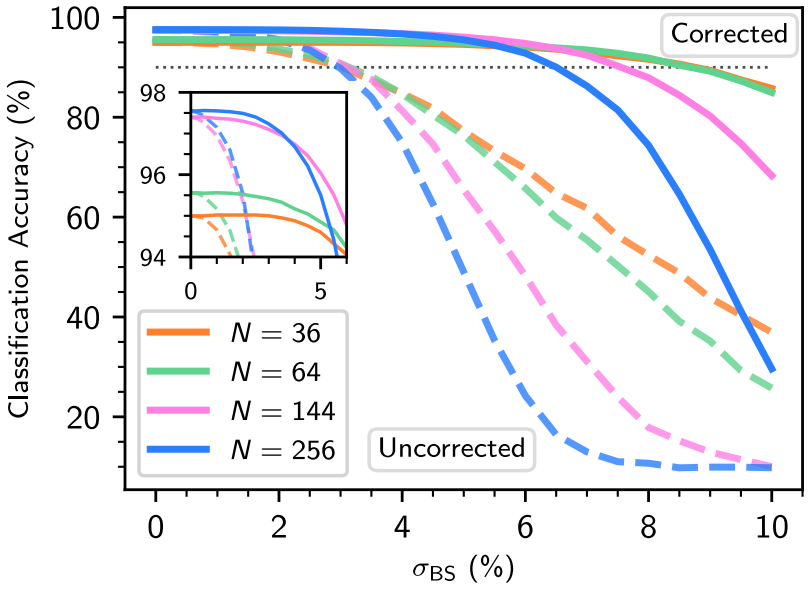

Figure 3-11: Median accuracy for 300 unitary circuits as a function of σ_{BS} with and without correction for a photonic image classifier for the MNIST task with $N = \{36, 64, 144, 256\}$ neurons. Error correction significantly improves the fabrication tolerance of the neural network to beyond current-day process tolerances, even for systems with hundreds of modes. As the inset shows, even circuits with 4% splitter error preserve the baseline performance within 1%.

individual filter implemented using TBUs with fabrication imperfections.

The transfer function $T_i(\omega)$ of a single tunable coupling ring can be derived with Mason's gain formula [86, 87]:

$$T_i(\omega) = \frac{a_{\text{loop}} a_{\text{top}} a_{\text{bot}} (\tau_1\tau_2 + \kappa_1\kappa_2) e^{i(k(2z_1+z_2)+\theta+\phi)} - a_{\text{bot}}\tau_1\tau_2 e^{ikz_1} + a_{\text{top}}\kappa_1\kappa_2 e^{i(kz_1+\theta)}}{a_{\text{loop}} a_{\text{top}}\tau_1\tau_2 e^{i(k(z_1+z_2)+\theta+\phi)} - a_{\text{loop}} a_{\text{bot}}\kappa_1\kappa_2 e^{i(k(z_1+z_2)+\phi)} - 1} \tag{3.20}$$

where $k = n(\omega)\omega/c$, $\tau_1 = a_{\text{splitter},1}\cos(\pi/4+\alpha)$, $\tau_2 = a_{\text{splitter},2}\cos(\pi/4+\beta)$, $\kappa_1 = a_{\text{splitter},1}\sin(\pi/4+\alpha)$, $\kappa_2 = a_{\text{splitter},2}\cos(\pi/4+\beta)$, z_1 is the interferometer arm length, z_2 is the length of the feedback loop, and $a_{\text{splitter},1}$, $a_{\text{splitter},2}$, a_{loop}, a_{top}, a_{bot} are the amplitude transmissions of the first and second splitters, the feedback loop, top arm of the tunable coupler, and bottom arm of the tunable coupler, respectively. (Fig. 3-13).

The transfer function $T_i(\omega)$ for each ring was individually computed and multiplied to yield the overall system response $T(\omega) = \prod_i T_i(\omega)$. From this result we found the group delay of the system $\tau(\omega) = -\text{d}/\text{d}\omega[\arg T(\omega)]$. The group delay dispersion was calculated with a least squares linear fit to the group delay profile. Using the constrained optimization by linear approximations (COBYLA) routine in `SciPy` [88, 89], we trained the TBU parameters on an idealized model to implement a group delay dispersion of -85 ps/nm over the bandwidth of a 50 GHz ITU channel.

Figure 3-14 shows the group delay τ profiles for 500 randomly generated TDC modules implemented using TBUs with $\sigma_{\text{BS}} = \{2, 4\}\%$ before (top) and after (bottom) error

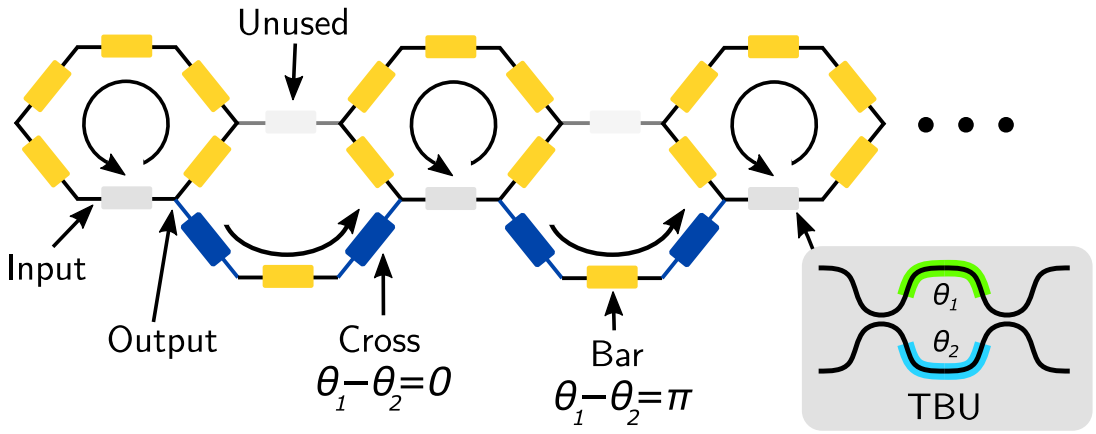

Figure 3-12: A tunable dispersion compensator (TDC) can be implemented on a recirculating waveguide mesh with 15 tunable-coupling ring resonators coupled serially to one another.

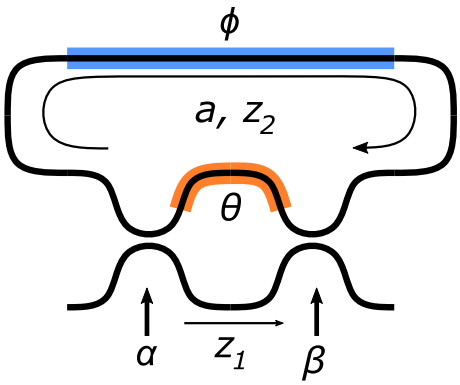

Figure 3-13: Model for tunable coupling ring. The ring coupling is set by an MZI with errors α, β and internal phase θ, and the resonance is set with a phase setting ϕ. The coupler is assumed to be lossless and the feedback loop is assumed to have a round-trip transmission a.

correction. Similar to optical neural networks, precise implementation of a TDC requires accurate phase control throughout the circuit. Fabrication errors introduce spurious phases at each resonance, which results in significant variation of the dispersion profile for even slight component errors. As our results show, correcting the parameters of each TBU locally is sufficient to restore the desired dispersion profile.

While we can correct the coupling and phase parameters for each ring, we cannot correct for errors in the closed feedback loop, which is implemented by programming each TBU to the bar state. Any error $\alpha \neq \beta$ will introduce some loss at each TBU programmed to the bar state, as the bar transmission is reduced to $\cos^2(\alpha - \beta)$. The remainder of the light is directed into unused couplers in the circuit, effectively incurring loss. This alters the critical coupling condition, resulting in the slight spread in the corrected dispersion profile observed in our simulations for $\sigma_{BS} = 4\%$. Our simulations assume α, β are independent, Gaussian random variables; in practice, however, α, β for a single device are strongly correlated [90, 65] and the bar state will be nearly perfect. Therefore, our

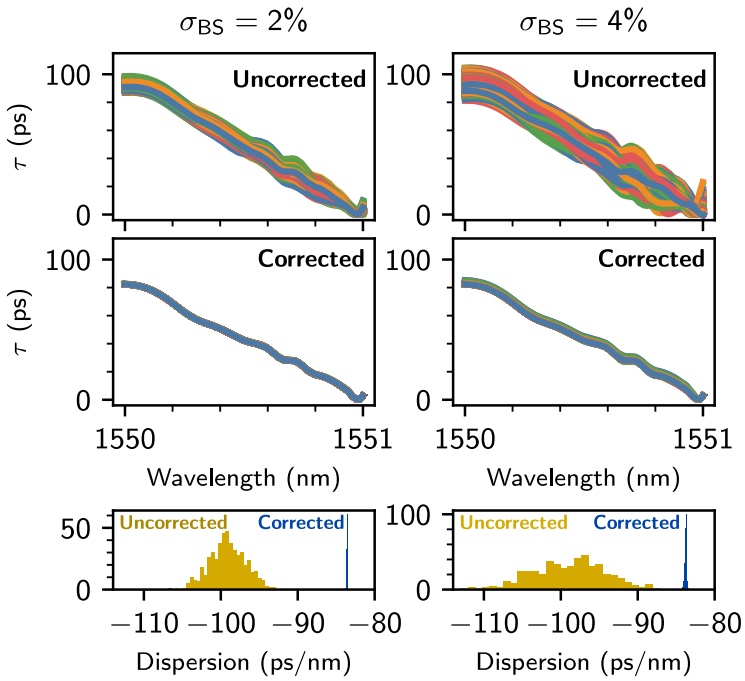

Figure 3-14: After training the mesh parameters to implement a fixed linear group delay dispersion on an ideal model, small beamsplitter errors will introduce variations in the implemented group delay τ profile. Plotted are the group delay profiles for 500 randomly generated circuits before and after correction. Correcting the settings of each TBU restores the desired performance, eliminating the need to retrain on the hardware. Also displayed is the distribution of the group delay dispersion before and after correction.

simulations likely overestimate the loss incurred at each TBU programmed to the bar state.

3.7 Modeling hardware errors from insertion loss

The intrepid experimentalist will notice that through this entire discussion, we have ignored the effect of device loss. Device loss is not inherently an obstacle for the work we have done so far; suppose that the photonic chip exhibits a uniform loss c through all paths of the mesh. If this were the case, c is a global constant we can factorize out of the matrix and everything we discuss above still applies.

In practice, the loss of each component in the chip forms a distribution that results in interfering paths having slightly different transmissivities, resulting in the circuit's transfer matrix being non-unitary. This constitutes an additional source of error we do not consider; in this section, we reproduce our earlier results taking into account variable device loss.

For feedforward circuits, loss modeling requires a slight correction to the error metric in equation (3.22). Two matrices U and cU, where c is a scalar constant $0 < c < 1$, are identical from the perspective of hardware performance, but have a Frobenius distance of

$1 - c$. We correct for this by modifying equation (3.22) as follows:

$$\epsilon = \frac{1}{\sqrt{N}} \min_{c \in [0,1]} \left[\left(\sum_{ij} |U_{\text{hardware},ij} - cU_{ij}|^2 \right)^{1/2} \right] \tag{3.21}$$

This expression returns $\epsilon = 0$ for two matrices (U, cU). For two unitary matrices $(U, U_{\text{hardware},ij})$, this expression is minimized at $c = 1$ and reproduces equation (3.22). In other cases, the error will be minimized at a value c corresponding roughly to the average transmission through all paths in the circuit.

We now require a model for the loss within a programmable circuit. Phase shifters in the SOI platform have sufficiently improved to induce no excess insertion loss beyond the waveguide propagation loss; this has been observed for semimetal (TiN) heaters suspended over the waveguide [91], where the TiN is placed sufficiently distant from the waveguide to not interact with the optical mode, and also recently for nano-optical electromechanical (NOEM) phase shifters [92]. We can therefore model the insertion loss of each phase shifter as the waveguide propagation loss and the variable optical loss as originating from the wafer-scale distribution of waveguide loss. For the phase shifters, we assume a 400 um long actuation region.

Our simulations assume three possible loss distributions:

- A "conservative" loss distribution based on [38], where efficient thermo-optic tuning is realized by driving a current directly into the waveguide to induce Joule heating. The authors characterized a wafer scale loss distribution of 0.23 $\pm$ 0.13 dB per heater. This is a relatively high loss per phase shifter, as dopants are introduced directly into the waveguide and interact with the optical mode. We choose an exponentially modified Gaussian distribution as it better fits the histogram shown in Figure 3 of [38].

- A "typical" loss distribution assuming a titanium nitride (TiN) heater suspended over the waveguide. The TiN can be placed sufficiently distant from the waveguide to not interact with the optical mode, as described in [91]. These devices can be optimized to be as efficient as those in [38], but the loss per device will be limited by the waveguide propagation loss. For waveguide loss, we use the wafer-level statistics described in [93] for a ridge (fully-etched) waveguide (2.1 $\pm$ 0.25 dB/cm). Assuming a 400 um long thermal tuner, the loss per heater is 0.084 $\pm$ 0.01 dB. We note this is a conservative estimate of the loss variation for our simulations, as the data in [93] is reported over the wafer-scale, not the die-scale. For these simulations, we assume the distribution to be Gaussian.

- A "state-of-the-art" loss distribution based on the improved waveguide loss and uniformity obtained in [93] by H_2 thermal annealing the waveguides. We assume the circuit uses rib waveguides, which exhibit a reduced loss of 0.1 $\pm$ 0.04 dB/cm. This corresponds to a thermal tuner loss of 0.004 $\pm$ 0.0016 dB.

For the directional coupler, we assume the loss to originate from waveguide propagation. Assuming a propagation length of 100 um to ensure the waveguide bends are

adiabatic, the loss per coupler would be 0.021 $\pm$ 0.0025 dB for the conservative and typical distributions, and 0.001 $\pm$ 0.0004 dB for the state-of-the-art.

Figure 3-15 shows the matrix error ϵ before and after correction for circuits with variable optical loss. For all loss distributions, we find that hardware error correction improves ϵ, and the state-of-the-art reproduces closely the results presented earlier for unitary circuits. However, typical loss distributions exhibit a reduced benefit to ϵ from error correction, and a more significant penalty is observed for the conservative distribution. We attribute this drop in performance to non-unitary (loss-induced) errors that cannot be corrected for by adjusting the parameters of each MZI. To confirm this, we attempted to numerically optimize each MZI's phase shifter settings after applying hardware error correction, but found only marginal improvements in ϵ of less than 1%.

Upon first look, this data would suggest that error correction is not useful for any realistic device, as the improvement in ϵ is quite small. However, as we noted before, ϵ is an abstract metric of "error." To truly understand the impact of non-unitary errors, we need to benchmark on a useful application, such as the optical neural network studied earlier. The results of these simulations are shown in Figure 3-16; remarkably, the neural network performance for the typical and state-of-the-art distributions are indistinguishable from those for unitary circuits. Moreover, circuits drawn from the conservative distribution perform nearly as well following error correction. These circuits are slightly less robust to error, but can still preserve over 90% accuracy when $\sigma_{\text{BS}} = 5\%$, which is well above typical foundry tolerances.

Finally, for completeness, we repeated our simulations for a tunable dispersion compensator (TDC) implemented in a recirculating mesh. These results are shown in Figure 3-17, and the typical and state-of-the-art distributions match well with the lossless results presented earlier. For the conservative distribution, there still remains significant variation in the group delay profile after correction; however, as the histogram shows, optical loss appears to introduce only a static group delay and does not affect the group delay dispersion. This is likely due to the changes in resonator coupling induced by device loss, and is particularly significant for the conservative case, where an average of 0.23 dB insertion loss per phase shifter would imply an additional $\sim$1 dB loss in the feedback loop. However, even for this case, the error in the group delay dispersion is greatly reduced.

Our results suggest that despite the apparent increase in ϵ due to optical loss, the benefits of error correction for optical neural networks are nearly unaffected. Interestingly, it implies that "coherent" errors, such as those introduced by imperfect beamsplitters, are more likely to cause problems for classification tasks than "incoherent" errors. This suggests that hardware error correction can greatly improve the performance of both feedforward and recirculating circuits, even for devices with relatively high optical losses. As fabrication processes improve, the effect of these losses on circuit performance will diminish further. Moreover, it has recently been shown that arbitrary feedforward circuits can be programmed using MZIs that omit the external phase shifter ϕ and instead program both internal arms of the interferometer [94]. This effectively halves the circuit depth and would further reduce the impact of device losses on circuit error.

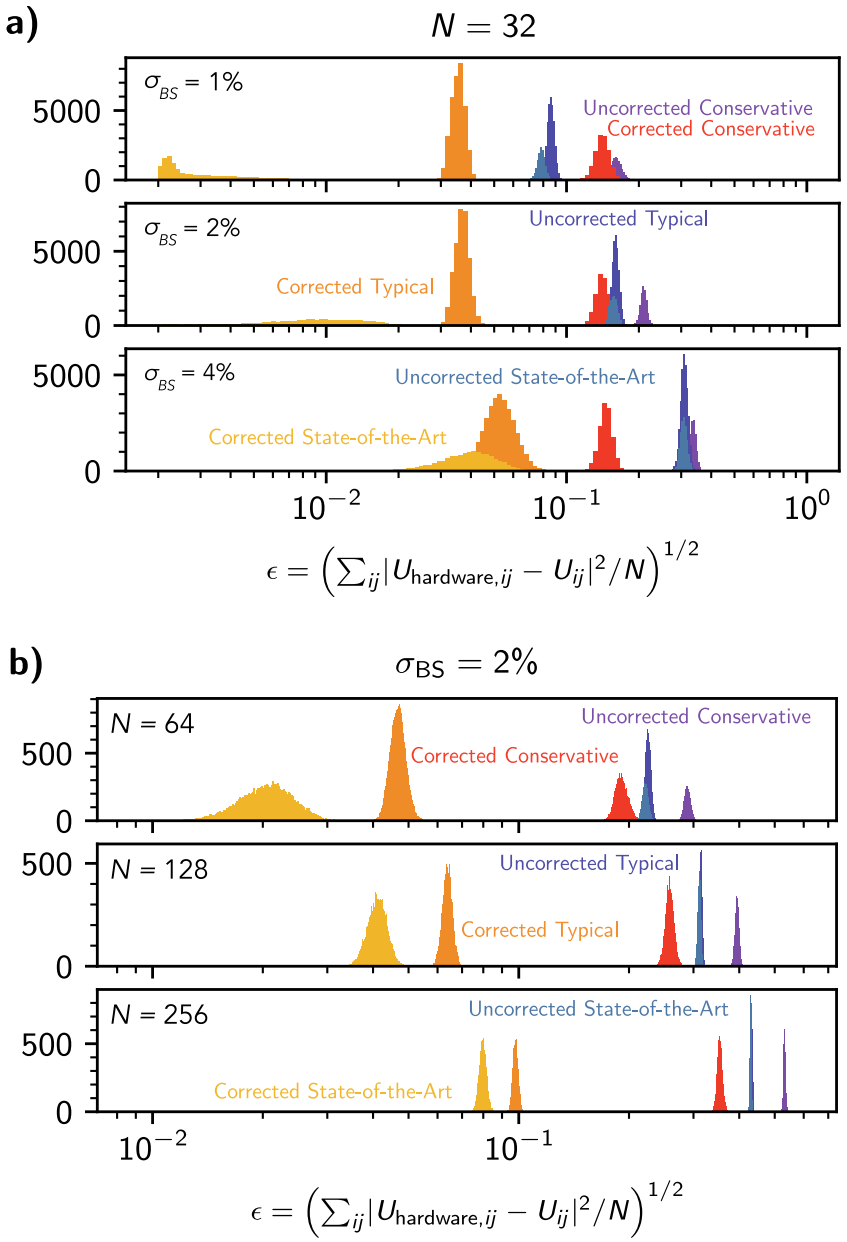

Figure 3-15: a) Matrix error ϵ for 100 random unitaries implemented on 100 random circuits for $N = 32$ assuming different loss distributions. The typical and state-of-the-art distributions overlap very closely. b) Matrix error ϵ as a function of N for $\sigma_{\text{BS}} = 2\%$ and different loss distributions.

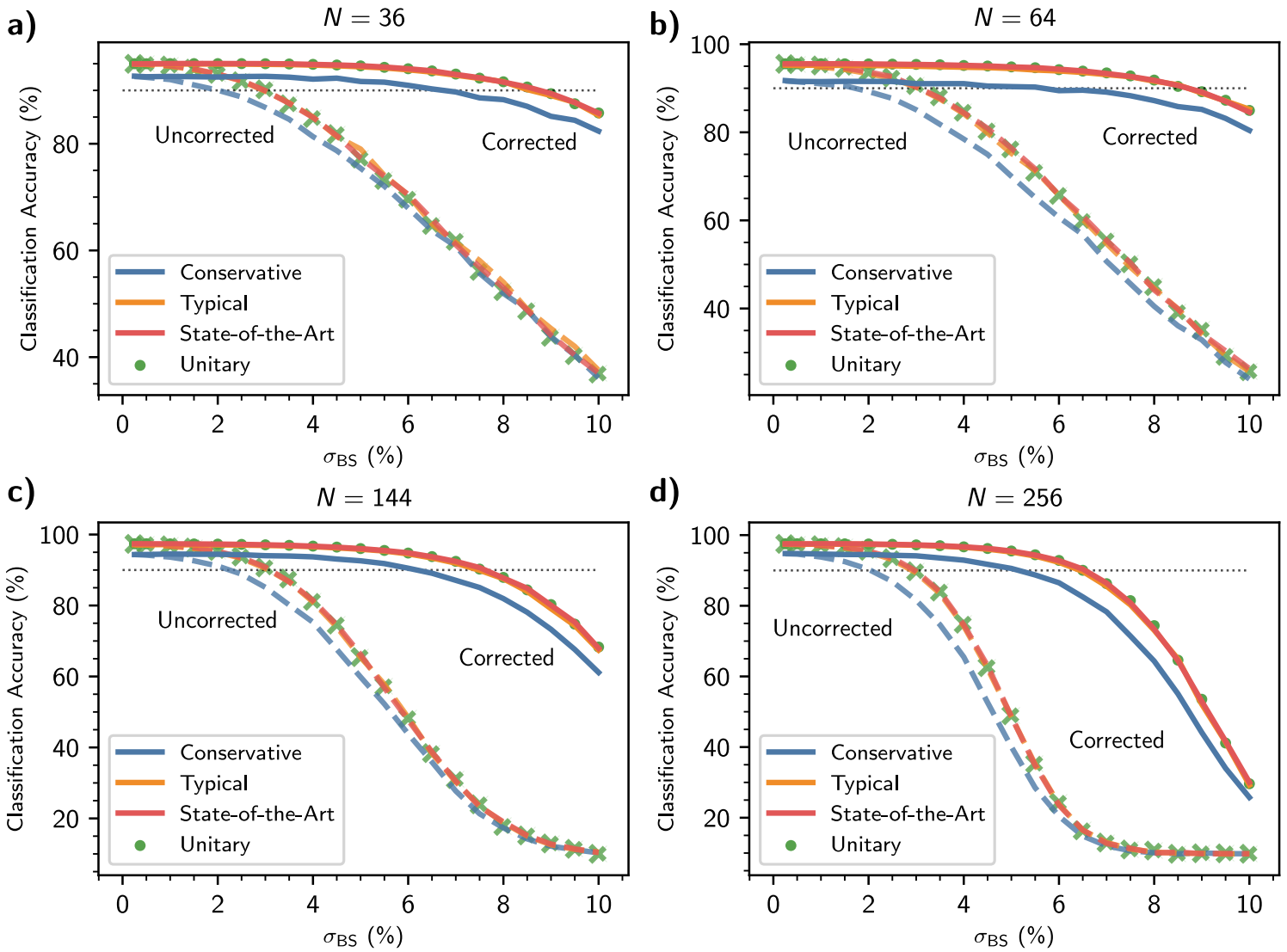

Figure 3-16: MNIST classification accuracy for a two layer optical neural network with a) 36; b) 64; c) 144; and d) 256 modes assuming variable optical loss. The results for a unitary circuit presented earlier are plotted for comparison. The typical (orange) and state-of-the-art (red) results overlap very closely with the results for unitary circuits.

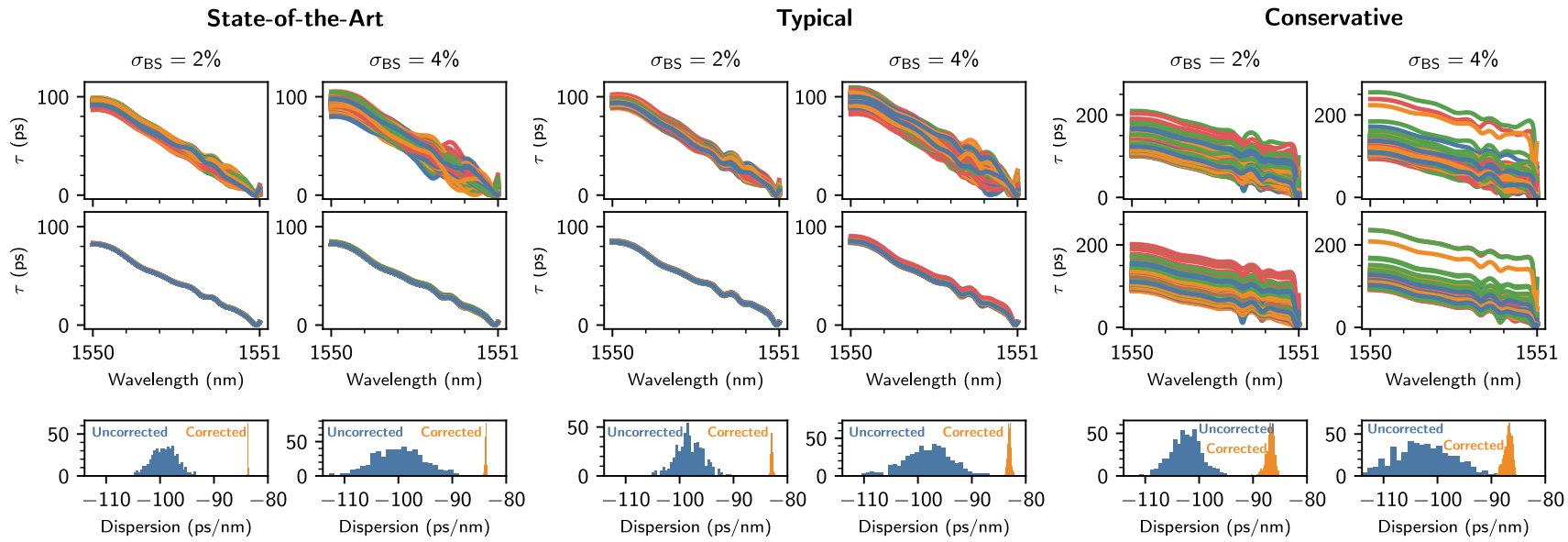

Figure 3-17: Simulations of a tunable dispersion compensator (TDC) implemented on a recirculating waveguide mesh assuming state-of-the-art, typical, and conservative device losses. The top plots show the group delay profile implemented before and after correction, while the bottom histograms show the group delay dispersion. For all loss distributions hardware error correction obtains the desired group delay dispersion, albeit with some additional spread introduced by the loss within the devices. While the group delay profiles for circuits drawn from the conservative distribution appear to show little effect from error correction, it still recovers the required group delay dispersion with high accuracy.

3.8 Scaling to larger circuits

This chapter has presented a local error correction algorithm for programmable photonic circuits and, through Monte Carlo simulations of hardware, demonstrated that it greatly improves the accuracy of these systems under realistic conditions of manufacturing error.

Monte Carlo simulations, however, are generally unsatisfying–we can empirically estimate how our approach scales, but it doesn't lend us any insight into what the ultimate limits of our approach are. Claiming a true improvement over existing devices requires some quantitative analysis of the system; to this end, near the latter stages of the project, we began developing analytical expressions for error scaling in programmable photonics.

We start with the Frobenius norm ϵ, which is how we benchmarked matrix error. The hardware error ϵ between a desired unitary matrix U and the implemented matrix U_{hardware} can be quantified by the Frobenius norm:

$$\epsilon = \frac{1}{\sqrt{N}} \left(\sum_{ij} |U_{\text{hardware},ij} - U_{ij}|^2 \right)^{1/2} \tag{3.22}$$

Unitary circuits decompose arbitrary matrices into a product of unitary matrices T_{ij}:

$$U = D \prod_{ij} T_{ij}(\theta, \phi, \alpha, \beta) \tag{3.23}$$

Let's isolate the contribution of a single imperfect beamsplitter in the circuit. The matrix error induced by a single beamsplitter error α can be computed as:

$$\epsilon = \frac{1}{\sqrt{N}} \left(\sum_{ij} |T_{ij}(\theta, \phi, \alpha = 0, \beta = 0) - T_{ij}(\theta, \phi, \alpha, \beta = 0)|^2 \right)^{1/2} \tag{3.24}$$

The Frobenius norm is unitarily invariant, which originates from the cylic property of the trace; thus, only the unitary matrix corresponding to the beamsplitter error needs to be considered in the calculation of ϵ:

$$\epsilon^2(\alpha) = \frac{1}{N} \sum_{ij} |H_{1,ij}(\phi, \alpha) - H_{1,ij}(\phi, 0)|^2 \tag{3.25}$$

$$= \frac{1}{N} \sum_{ij} \text{Tr} \left[(H_{1,ij}(\phi, \alpha) - H_{1,ij}(\phi, 0))^\dagger (H_{1,ij}(\phi, \alpha) - H_{1,ij}(\phi, 0)) \right] \tag{3.26}$$

$$= \frac{1}{N} \text{Tr} \left[2I - H_{1,ij}(\phi, \alpha)^\dagger H_{1,ij}(\phi, 0) - H_{1,ij}(\phi, 0)^\dagger H_{1,ij}(\phi, \alpha) \right] \tag{3.27}$$

$$= \frac{1}{N} \left(2N - 2\text{Re} \left[\text{Tr} \left[H_{1,ij}(\phi, \alpha)^\dagger H_{1,ij}(\phi, 0) \right] \right] \right) \tag{3.28}$$

$$= \frac{1}{N} \left(2N - 2 \left(2 \cos \alpha + N - 2 \right) \right) \tag{3.29}$$

$$= \frac{4}{N} \left(1 - \cos \alpha \right) \approx \frac{2\alpha^2}{N} \tag{3.30}$$

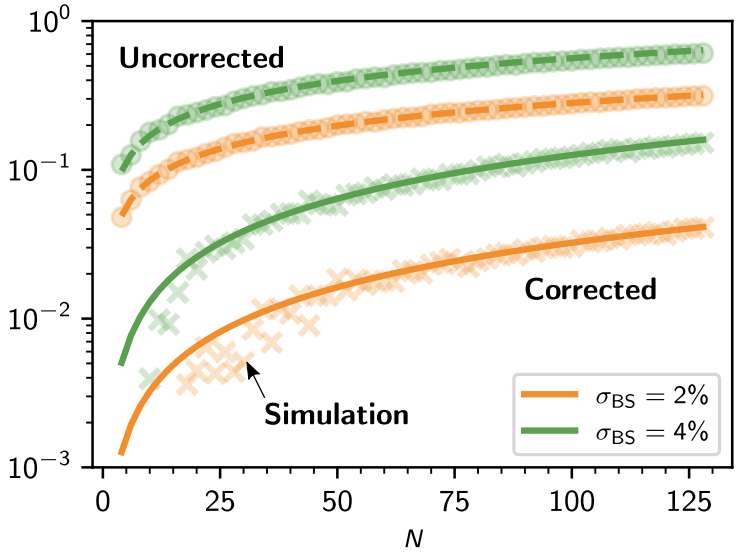

Figure 3-18: Equations (3.33) and (3.49) for the uncorrected and corrected beamsplitter errors as a function of circuit size N. The scatter plot shows the median error for 12 simulations, showing excellent agreement with the derived expressions.

Repeating this calculation for β yields the same result.

In a unitary circuit with $N(N-1)/2$ interferometers, the average error is therefore:

$$\langle\epsilon\rangle = \sqrt{\frac{N(N-1)}{2}\left(\langle\epsilon^2(\alpha)\rangle + \langle\epsilon^2(\beta)\rangle\right)} \tag{3.31}$$

$$= \sqrt{(N-1)\left(\langle\alpha^2\rangle + \langle\beta^2\rangle\right)} \tag{3.32}$$

$$= \sqrt{2\,(N-1)}\sigma_{BS} \tag{3.33}$$

Figure 3-18 shows the expression in Equation (3.33) plotted against simulation results; they show excellent agreement with the derived expression.

If we can correct all errors in θ, then $\epsilon_{\text{corrected}} \to 0$. We can therefore estimate the expected $\epsilon_{\text{corrected}}$ by computing the fraction of MZIs that cannot be programmed to the required splitting value, i.e. the condition in equation (3.10).

Consider a device for which we can correct ϕ, ψ_1, ψ_2, but are unable to correct θ. Any unitary U can be decomposed into a product of matrices $U = D\prod T_{ij}$, where D is diagonal and T_{ij} is a $N \times N$ block matrix with non-trivial entries:

$$\begin{bmatrix} e^{i\psi_1} & 0 \\ 0 & e^{i\psi_2} \end{bmatrix} \begin{bmatrix} \sin(\theta/2) & \cos(\theta/2) \\ \cos(\theta/2) & -\sin(\theta/2) \end{bmatrix} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \tag{3.34}$$

An error $\theta \to \theta + \Delta$ produces a contribution to $\epsilon_{\text{corrected}}$ of:

$$\epsilon^2(\Delta) = \frac{1}{N}\left(2N - 2(2\cos(\Delta/2) + N - 2)\right) \tag{3.35}$$

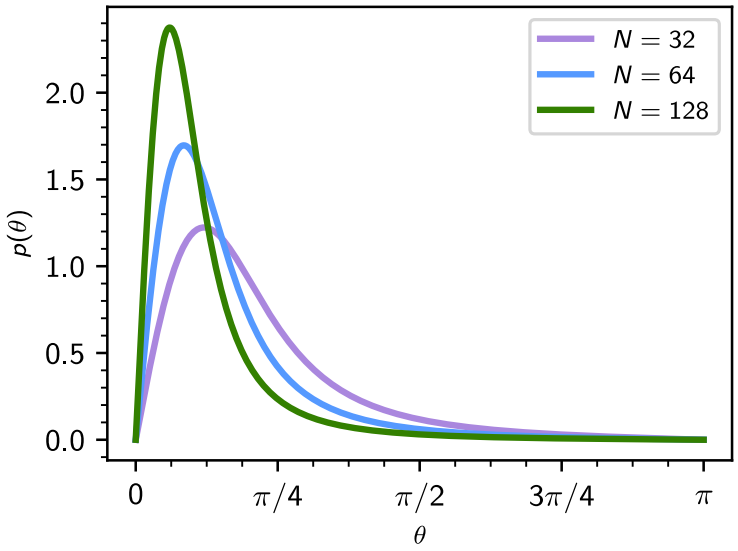

Figure 3-19: The probability density function of the internal phase shifter setting θ for $N = \{32, 64, 128\}$. As N increases, $\langle\theta\rangle$ is further biased towards 0.

$$= \frac{8}{N}\sin^2(\Delta/4) \approx \frac{\Delta^2}{2N} \tag{3.36}$$

On average, given θ cannot be realized, $\langle\Delta^2\rangle = 2(\langle\alpha^2\rangle + \langle\beta^2\rangle) = 4\sigma_{\text{BS}}^2$ and the error per device will be $\langle\epsilon^2(\Delta)\rangle = 2\sigma_{\text{BS}}^2/N$. The total error for the circuit is therefore:

$$\langle\epsilon_{\text{corrected}}\rangle = \sqrt{(N-1)\sigma_{\text{BS}}^2 P(\theta < 2|\alpha+\beta|)} \tag{3.37}$$

where $P(\theta < 2|\alpha+\beta|)$ is the probability that a device in the circuit needs to be programmed to a splitting that cannot be realized.

The distribution of internal phase shifter settings θ for a unitary circuit can be determined from the Haar measure. For a given MZI, ref. [77] shows that:

$$p_{n,i}(\theta) = (n-i)\sin(\theta/2)\cos^{2(n-i)-1}(\theta/2) \tag{3.38}$$

where $n \in [2, N]$, $i \in [1, N-n+1]$ are indices denoting the position of the MZI in the network (see [77] for the mapping). The distribution of θ over the entire circuit can therefore be written as (Fig. 3-19):

$$p(\theta) = \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} k\sin(\theta/2)\cos^{2k-1}(\theta/2) \tag{3.39}$$

Integrating this expression yields the fraction of beamsplitters with a required splitting below ξ:

$$P(\theta < \xi) = \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} \int_0^{\xi} k\sin(\theta/2)\cos^{2k-1}(\theta/2)\, \mathrm{d}\theta \tag{3.40}$$

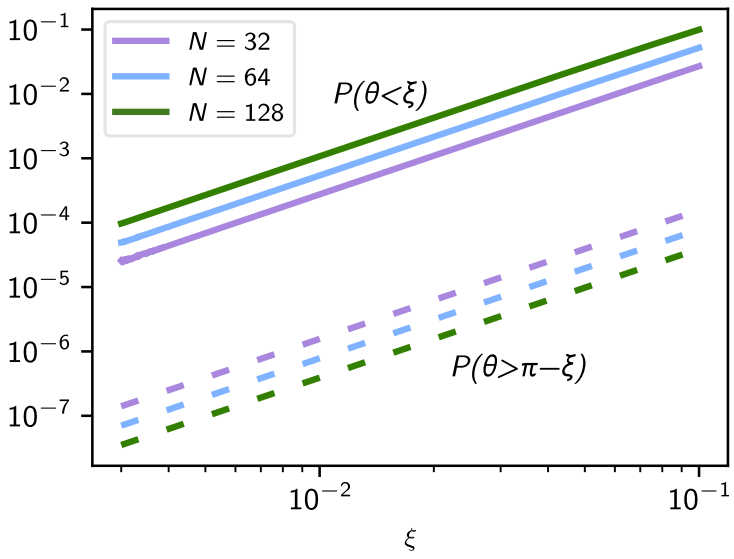

Figure 3-20: The probability an MZI must be programmed to a splitting $\theta < \xi$, $\theta > \pi - \xi$ for $N = \{32, 64, 128\}$. $P(\theta > \pi - \xi)$ is orders of magnitude smaller than $P(\theta < \xi)$; thus, we can neglect it when computing the expected corrected hardware error.

$$= \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} \left(1 - \cos^{2k}(\xi/2)\right) \tag{3.41}$$

$$= \frac{N+1}{N-1} - \frac{4\left(N + \cot^2(\xi/2)\left(\cos^{2N}(\xi/2) - 1\right)\right)}{N(N-1)(1-\cos\xi)} \tag{3.42}$$

For small device errors, equation (3.41) can be Taylor expanded to:

$$\sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} \left(\frac{k\xi^2}{4}\right) = \frac{N+1}{12}\xi^2 = \frac{2(N+1)}{3}\sigma_{BS}^2 \tag{3.43}$$

On the other hand, the probability that $\theta > \pi - 2|\alpha - \beta|$ is:

$$P(\theta > \pi - 2|\alpha-\beta|) = \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} \int_{\pi-2|\alpha-\beta|}^{\pi} k \sin(\theta/2)\cos^{2k-1}(\theta/2)\, d\theta \tag{3.44}$$

$$= \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} \cos^{2k}\left(\frac{\pi}{2} - |\alpha-\beta|\right) \tag{3.45}$$

$$\approx \sum_{k=1}^{N-1} \frac{2(N-k)}{N(N-1)} 2^k \sigma_{BS}^{2k} \approx \frac{4\sigma_{BS}^2}{N} \tag{3.46}$$

For moderately large N, this quantity is order of magnitudes smaller than $P(\theta < 2|\alpha+\beta|)$; we can therefore disregard it when estimating the average corrected error (Fig. 3-20).

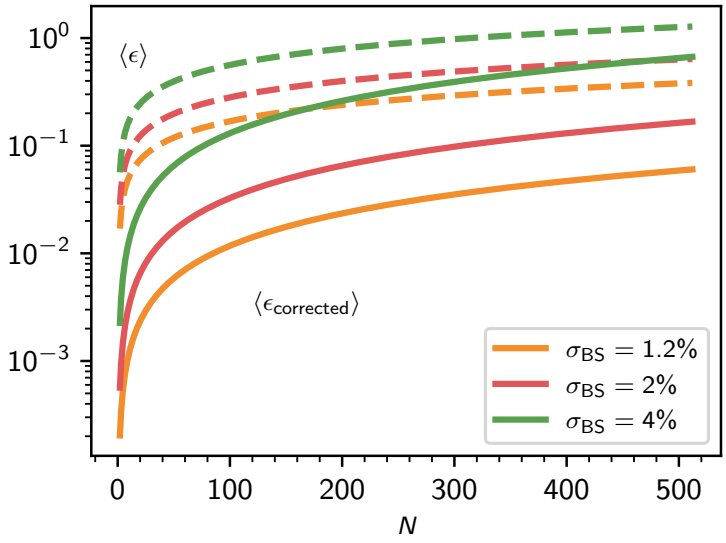

Figure 3-21: $\langle\epsilon\rangle$, $\langle\epsilon_{\text{corrected}}\rangle$ as a function of circuit size N for $\sigma_{\text{BS}} = \{1.2, 2, 4\}\%$.

The average corrected error is therefore:

$$\langle\epsilon_{\text{corrected}}\rangle = \sqrt{(N-1)\sigma_{\text{BS}}^2 P(\theta < 2|\alpha+\beta|)} \tag{3.47}$$

$$= \sqrt{(N-1)\sigma_{\text{BS}}^2 \left(\frac{2(N+1)}{3}\sigma_{BS}^2\right)} \tag{3.48}$$

$$= \sigma_{\text{BS}}^2 \sqrt{\frac{2(N^2-1)}{3}} \tag{3.49}$$

This expression is plotted in Figure 3-18 and also shows excellent agreement with simulation results.

We find that error correction effectively reduces the hardware error from ϵ to $\approx (1/\sqrt{6})\epsilon^2$. The expected error improvement is:

$$\frac{\langle\epsilon\rangle}{\langle\epsilon_{\text{corrected}}\rangle} \approx \frac{\sqrt{3}}{\sigma_{\text{BS}}\sqrt{N+1}} \tag{3.50}$$

$\langle\epsilon\rangle$ and $\langle\epsilon_{\text{corrected}}\rangle$ as a function of N are plotted in Figure 3-21. We consider $\sigma_{\text{BS}} = 1.2\%$, which is the state-of-the-art reported in [60], as well as more relaxed tolerances $\sigma_{\text{BS}} = \{2, 4\}\%$. For σ_{BS} as high as 4%, error correction produces at least a factor of two (and often more) improvement in the error for circuits as large as $N = 500$. We therefore expect our approach to have wide applicability in the near term as the size of programmable photonic circuits scale up.

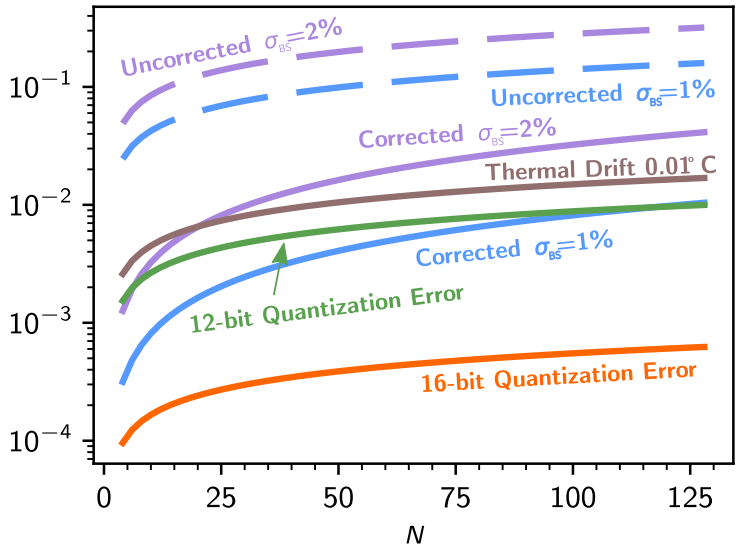

Figure 3-22: The relative error contributions from beamsplitter error, thermal drift, and quantization error as a function of circuit size N. If the component errors are left uncorrected, then even small beamsplitter variations produce errors significantly larger than those produced by dynamic effects. Hardware error correction suppresses these component errors to a point where dynamic effects begin to play an important role, particularly if the DAC resolution is low.

3.9 What about other types of errors?

Our analysis so far has considered beamsplitter imperfections and loss. There are other practical errors, however, that we must contend with when using programmable photonics for signal processing. In this section, we consider some of these other sources of error; we find that, in practice, they are far smaller sources of error than beamsplitter variation.

There can also be errors in the phase shifter settings; however, the primary source of these errors is a static error originating from microscopic changes in waveguide geometry between the interferometer arms [90]. This static error is calibrated out in the first step of the characterization protocol.

This calibration cannot account for dynamic errors, however. Potential sources of dynamic phase errors include thermal drift, thermal crosstalk between phase shifters, and quantization error. In this section, we show that the contribution of these effects to the hardware error is significantly smaller than the static errors considered earlier.

To start, we find that any error Δ induced in a single phase setting by these effects can be computed to be:

$$\epsilon^2(\Delta) = \frac{1}{N}\left(2N - 2\mathrm{Re}\left[\mathrm{Tr}\left[H_{2,ij}(\theta+\Delta,0)^\dagger H_{2,ij}(\theta,0)\right]\right]\right) \tag{3.51}$$

$$= \frac{1}{N}\left(2N - 2\left(\cos\Delta + N - 1\right)\right) \tag{3.52}$$

$$= \frac{1}{N}\left(2 - 2\cos\Delta\right) \tag{3.53}$$

$$\approx \frac{\Delta^2}{N} \tag{3.54}$$

We now consider the error induced by each of these effects.

- **Thermal drift**: Typical thermo-electric cooling (TEC) systems can maintain chip temperature stabilities better than $< 0.01^\circ$ C [95]. The thermo-optic coefficient $\mathrm{d}n/\mathrm{d}T$ of silicon is 1.8×10^{-4} K^{-1} [96]; for an $L = 200$ μm long phase shifter, a temperature gradient of $< 0.01^\circ$ C induces a phase error of $2\pi(\mathrm{d}n/\mathrm{d}T)L(\Delta T)/\lambda \approx 1.5 \times 10^{-3}$ at $\lambda = 1550$ nm, which is an order of magnitude smaller than the expected beamsplitter error.

- **Thermal crosstalk**: Thermal crosstalk is largely deterministic and dominated by the nearest-neighbor crosstalk, which can be accounted for in the phase shifter characterization. Additionally, crosstalk can be suppressed by spacing interferometers sufficiently apart on the chip [38]; a spacing of 135 μm, for instance, has been measured to generate a crosstalk with the neighboring MZI of less than 0.02 rad/rad [91]. Since thermal crosstalk decays with increasing separation, we expect with careful design this effect should not dominate hardware error.

- **Quantization error**: Quantization error originates from the digital-to-analog converters (DACs) used to program voltages into the phase shifters. Consider an N-bit DAC whose 2^N codewords range from zero voltage to the voltage $V_{2\pi}$ required for a 2π phase shift. Programming the M-th ($0 \leq M \leq 2^N - 1$) codeword will produce a voltage sampled uniformly over the distribution:

$$V_M = \frac{V_{2\pi}}{2^N}\left(M + \frac{1}{2}\right) \pm \underbrace{\frac{V_{2\pi}}{2^{N+1}}}_{N \text{ bits}} \tag{3.55}$$

  In a thermo-optic phase shifter, relative phase is a function of the voltage squared; the phase setting for the M-th codeword is therefore:

$$\phi_M = \frac{2\pi}{2^{2N}}\left(M + \frac{1}{2} \pm \frac{1}{2}\right)^2 \tag{3.56}$$

$$\approx \frac{2\pi}{2^{2N}}\left(M + \frac{1}{2}\right)^2 \pm \frac{2\pi}{2^{2N}}\left(M + \frac{1}{2}\right) \tag{3.57}$$

  The uncertainty in ϕ is maximum at $M = 2^N - 1$, where the phase setting is:

$$\phi \approx 2\pi \pm \underbrace{\frac{2\pi}{2^N}}_{N-1 \text{ bits}} \tag{3.58}$$

  which is one fewer bit of accuracy than for the voltage setting.

  The square-law dependence of phase on voltage therefore results in an N-bit DAC setting the phase to roughly $N - 1$ bits of accuracy. A 12-bit DAC will suppress

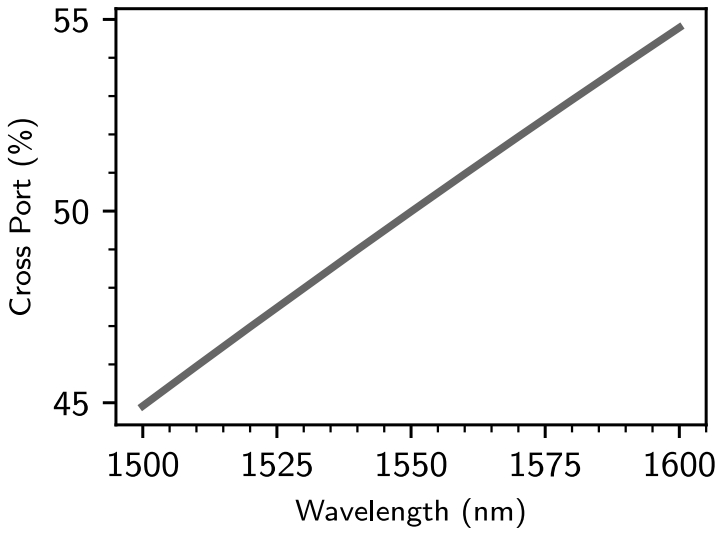

Figure 3-23: Wavelength vs. cross coupling for the optimally tolerant directional coupler design reported in [60].

worst-case quantization error per phase shifter to $\approx 9 \times 10^{-4}$, and 16 bits are sufficient to suppress error to below 6×10^{-5}.

In Figure 3-22 we plot the relative error contributions of these effects compared to static beamsplitter error. These estimates suggest that *uncorrected* component imprecision dominates the hardware error in programmable photonic circuits. However, once component errors are corrected, dynamic effects play a more significant role in the total hardware error. Accounting for these errors, now that we can resolve static hardware errors, would be an interesting area of future research.

3.10 Improving the bandwidth of photonic signal processing

A key advantage of optics for signal processing is its inherent parallelism. For instance photonic components, which can exhibit broadband performance over tens of nanometers in wavelength, correspond to intrinsic bandwidths of tens of terahertz. In comparison, digital signal processors struggle to reach clock rates exceeding a few gigahertz, due to the high insertion losses in these frequencies and the immense energy consumption required to charge and discharge a transmission line so quickly. Consequently, it would be a great advantage to have a photonic signal processor parallel processing data across many wavelength channels simultaneously. A typical digital ASIC for machine learning can reach throughputs of hundreds of TOPS (tera-operations per second); a comparable photonic system, multiplexing across the hundred wavelengths in the DWDM (dense wavelength-division multiplexing) standard, could reach throughputs of tens of petaops per second. The issue, however, is that in photonic circuits even slight differences in

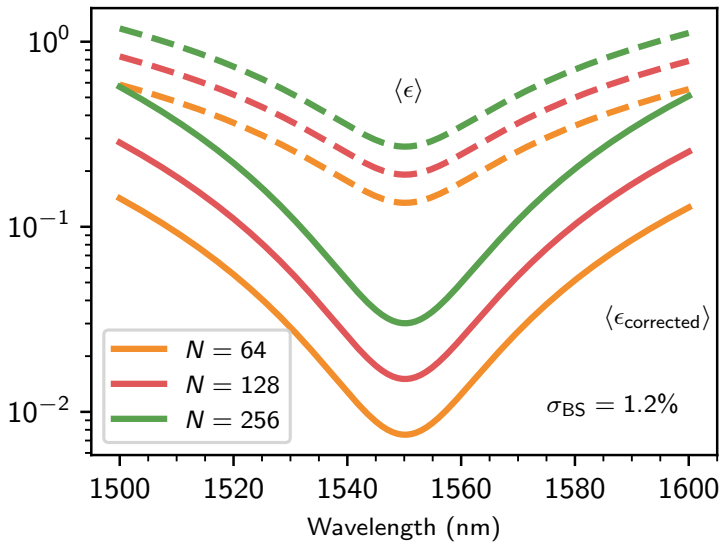

Figure 3-24: Average circuit error as a function of wavelength for $N = \{64, 128, 256\}$ using the optimal directional coupler design in [60].

component behavior, such as wavelength dependence, can produce significant disparities in performance across wavelength channels. This problem is particularly acute for the directional couplers that make up the 2×2 splitters in these circuits, which are strongly wavelength dependent.

We found that the $\epsilon \to \epsilon^2$ gain of error correction greatly improves the optical bandwidth of these systems. Since directional couplers are highly wavelength sensitive (Figure 3-23), dense wavelength-division multiplexing (DWDM) requires re-fabricating the same circuit with components optimized at each wavelength channel. Our approach, however, enables the use of the same hardware across a wide wavelength range. In Figure 3-24 we show the expected hardware errors for large circuits across a 100 nm bandwidth using the optimal splitter ($\sigma_{\text{BS}} = 1.2\%$) design in [60]. We find that the corrected error for an $N = 256$ circuit across a 60 nm bandwidth (1520-1580 nm) will be lower than the *uncorrected* error at the design wavelength $\lambda = 1550$ nm. Even lower errors could be achieved using multimode interference (MMI) couplers; these devices have large bandwidths but often suffer from static splitting imbalances [97], i.e., α, β are invariant to wavelength, but $\langle\alpha\rangle, \langle\beta\rangle \neq 0$. A circuit with large-bandwidth MMI couplers can thus use error correction to achieve a large instantaneous bandwidth, for instance to compute over many parallel wavelength channels.

3.11 Can this scale?

The results in Figure 3-21 suggest a fundamental error bound achievable with local correction for unitary circuits. Our approach yields comparable results to those achieved with self-configuration procedures [74, 73] but does not require a specific structure for the circuit or photodiodes within each device. If the condition in equation (3.10) is satisfied,

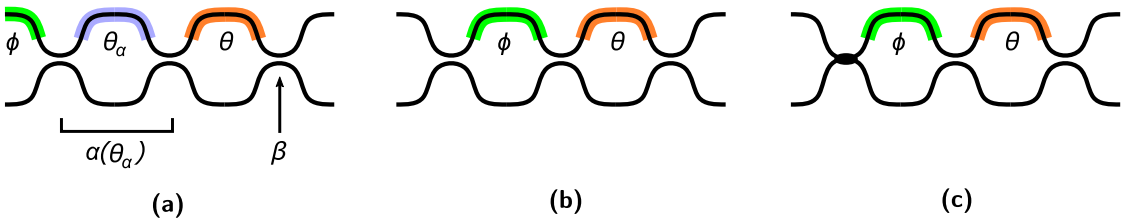

Figure 3-25: Alternate MZI unit cells with superior error scaling. (a), which incorporates an extra phase shifter, can guarantee zero error for splitting errors as high as 70-30. (b) and (c), which incorporate an additional 50-50 splitter and waveguide crossing, respectively, realize "asymptotic fault tolerance," where the error diminishes with increasing circuit size N.

local correction obtains $\epsilon_{\text{corrected}} = 0$ in $O(1)$ time. If this condition is not satisfied, we empirically found that it is sometimes possible to achieve a larger reduction in error with a *global* optimization approach, for instance with gradient descent [66, 71].

The problem is that these approaches, which require photodiodes within each device or output measurements whose number scale nonlinearly with the number of modes, become increasingly inaccessible experimentally as N scales up. For example, *in situ* backpropagation is quite challenging to realize on these photonic meshes with high accuracy and requires internal photodiodes/TIAs/ADCs at every MZI. Nonlinear optimization algorithms have even worse scaling behavior. Optimizing on a matrix error, for instance, requires N measurements per iteration to reconstruct the matrix, and most nonlinear optimization routines then require at minimum N^2 function evaluations per iteration. This means that optimization without internal detectors scales as $O(N^3)$ measurements per iteration. Some methods, such as Powell's methods, can have scaling as poor as $O(N^5)$, since each line search can require up to an additional N^2 function evaluations. Certain heuristic approaches, such as the Nelder-Mead method, do require only a few measurements per iteration. However, Nelder-Mead is believed to be inefficient for large numbers of parameters, is sensitive to the initial simplex, and has no guarantee of convergence. The advantage of local correction is that it requires minimal overhead and can guarantee a minimum error given certain guarantees on the component performance, making it ideal for standardizing performance across large numbers of chips.

Also, this error bound applies only to feedforward, unitary circuits with no redundant devices. ϵ lower than this bound can be achieved by incorporating additional, redundant MZIs; for instance, one can implement "perfect" optical gates by incorporating an additional phase shifter into the MZI, as shown in Figure 3-25a. This device can be trained with optimization to implement any desired unitary $T_{ij}(\theta, \phi)$ perfectly [98, 99]. The error correction formalism enables calculation of these settings analytically. One of the two constituent splitters is a passive component with error β, while the other splitter is an MZI that implements a tunable error $\alpha(\theta_\alpha)$. Any desired 2×2 unitary with a required splitting θ can then be implemented by setting θ_α such that $2|\alpha(\theta_\alpha) + \beta| < \theta < 2|\alpha(\theta_\alpha) - \beta|$ and correcting the resultant phase errors. For recirculating meshes the phase shifter settings are not constrained by the Haar measure, and so the benefit gained from error correction is not expected to diminish with increasing N. Error correction can play an important role

in scaling up the size of these circuits as well.

We do not necessarily even require redundant active components. In later work [57] we found that simple changes to the MZI unit cell, such as adding an extra 50-50 splitter on the input (Fig. 3-25b) or a waveguide crossing (Fig. 3-25c), produced an *error scaling that decreased with N*. In other words, as $N \to \infty$, the error $\epsilon_{\text{corrected}} \to 0$. This surprising find, which we termed *asymptotic fault tolerance*, resolved the scaling issues we unearthed in this work.

3.12 Conclusion

In conclusion, we have presented a protocol to correct for hardware errors in programmable photonic circuits. Unlike optimization-based approaches, our protocol utilizes a one-time calibration procedure to flexibly implement any desired functionality up to the limits of the hardware. We find that applying our approach to key application areas of programmable photonics, such as optical neural networks and programmable coupled-ring systems, enables resilience to fabrication errors well beyond modern-day process tolerances. Error correction also greatly reduces the overhead for programmable photonics that require optimization to deduce the hardware settings, as it eliminates the need to retrain for each individual set of hardware with unknown fabrication errors. Current process tolerances suggest that our approach enables improved functionality for systems of up to hundreds of modes, providing a new avenue for scaling up programmable photonics.

4

Alignment-free photonic interconnects

This chapter is adapted from work[1] reported in ref. [100].

4.1 Introduction

An early motivation for integrated photonics was the high cost of packaging. Early photonic systems comprised multiple discrete optical components co-packaged together; integrating as many of these components together onto a single chip would greatly reduce the cost per unit to the end user, improving scalability and adoption [101].

Today, the cost of integrated photonic systems is still dominated by the cost of the photonic packaging. Much of this originates from the need to align optical fibers to the chip for external input/output (I/O) [101]. As anyone who has worked with photonic chips in the lab will know, fiber coupling in and out of these devices is a slow, finicky process, requiring careful alignment to sub-micron accuracy. This is an entirely different paradigm from electronics packaging, which requires much more relaxed alignment tolerances of ± 10 μm that can be assembled by high-volume, automated pick-and-place tools capable of packaging hundreds of thousands of components per hour [101]. Optical alignment is far slower, requiring specialized, high-precision tools with far lower throughputs[2].

Moreover, during my PhD, a new set of trends emerged in the photonics community that made the packaging problem even more urgent:

- **Hybrid and heterogeneous integration**: While the goal of integrating all photonic components on a single chip continues to be pursued, it is becoming increasingly apparent that there is no photonic platform that can meet all possible requirements. Silicon photonics is the state-of-the-art for volume manufacturing, high-density integration, and compatibility with CMOS electronics. It however, lacks

[1]I thank Dr. Carlos Errando Herranz, Dr. Mohamed ElKabbash, Dr. Genevieve Clark, and Dr. Alexander Sludds for useful discussions.

[2]This problem is acutely felt in the photonics industry today, with major industry players frequently announcing new technologies for photonic coupling. For example, a few months before this thesis was written, in late 2022, Intel demoed live at their Innovation Day a new pluggable connector for interfacing fiber directly to bare die [102].

easy integration of light sources and a second-order nonlinearity (useful for Pockels modulation and nonlinear optics). While there has been a great deal of interest in academia on new platforms, such as lithium niobate, they lack the inherent scalability of silicon photonics, both in terms of manufacturability and the ability to densely integrate components.

The solution the community has converged upon is integrating photonic components from multiple material platforms together into a single chip. Examples abound in the literature–for instance, integrating a light source fabricated in indium phosphide onto a silicon photonic chip [103], coupling an external laser into a silicon nitride PIC [104], integrating diamond single photon sources onto an aluminum nitride PIC [105], or wafer bonding a lithium niobate film onto a silicon photonic chip [106].

- **Growing channel counts**: Nearly all commercial silicon products today are on the order of a few optical channels. However, future information processing systems in photonics could require the ability to interface tens to hundreds of optical channels. Scalable, alignment-tolerant photonic interfaces are critical to realizing such systems.

- **Increasing demands on performance**: As the depth of photonic circuits increase, coupling light in and out of these systems will have more stringent requirements on insertion loss, bandwidth, and alignment tolerance. A photonic coupler that is theoretically rated for no insertion loss is not useful if it cannot be feasibly aligned to in a scalable fashion.

As a result, progress in scaling photonic systems has been held back by a lack of reliable and easy-to-align photonic interconnects between components at the inter-chip and intra-chip levels. We confront this issue on a daily basis in our group, which led us to begin working on new approaches for optical interconnects.

The desired attributes in optical interconnects are:

- **Single mode propagation**: single mode systems are often far simpler to control and design for than multimode devices.

- **Low loss**: Insertion loss is critical for scaling to large circuit depths and especially when heterogeneously integrating multiple photonic platforms together.

- **Ease of manufacturing**: Scalable packaging will require compatibility with existing high-volume assembly tools.

- **Compatibility with high-density electrical interconnects**: Photonic systems will not exist in isolation, but will need to easily interface to CMOS electronic systems.

Tapered adiabatic couplers [107, 108, 109] are a common choice for interfacing to waveguides on a photonic integrated circuit (PIC), but at the expense of high demands on nanofabrication, wavelength-scale and mostly manual alignment, and difficult scaling. Alternative approaches have been proposed, including photonic "wirebonds" that connect

PICs through flexible polymer waveguides [110, 111, 112], integrated optical microlenses [113, 114], pitch-reducing interposers and fiber arrays [115, 116], and bulk optical components such as parabolic reflectors microfabricated into polymer films [117, 118, 119]. Small numbers of PICs can be also be connected by conventional fiber with edge coupling [120] or grating coupling [121, 122], but scaling to tens of channels becomes extremely challenging.

In this chapter, we introduce a photonic interconnect technology that is largely insensitive to misalignment. Our approach relies on the interaction between two waveguides crossing at an angle, which is optimized for efficient evanescent coupling at their intersection. This coupler is *invariant* to translational misalignment, as the intersection between two lines is invariant to any in-plane translation $\Delta r_{||}$. In addition to translational invariance, the coupling efficiency is far more insensitive to angular misalignment $\Delta\theta$ than conventional approaches such as edge coupling. Furthermore, we analyze this approach to photonic coupling and demonstrate alignment tolerance that is fundamentally impossible to achieve with edge or grating couplers, which exhibit a fundamental tradeoff $(\Delta r_{||}\Delta\theta)_{3\ \mathrm{dB}} = \lambda/\pi n$, where λ/n is the wavelength in an effective index n.

4.2 Photonic circuit boards

As a particular use case enabled by our approach, we introduce a "self-aligning photonic circuit board" (SAPCB) that unifies photonic integrated circuits, microchiplets, and electronics onto a single optoelectronic substrate. Similar to other optical PCBs [107, 108, 114, 118, 119], the SAPCB's waveguides are made of polymer, making them easy and scalable to fabricate. However, unlike other optical PCBs, which typically require defining complex waveguide routing in polymer to carry signals between components, the SAPCB consists solely of a linear array of waveguides, making it far easier to manufacture. By fabricating an array of waveguides with variable widths, one can create a universal connector to match PIC waveguides of varying materials or dimensions, thereby facilitating the assembly of diverse photonic and electronic components into high-density systems.

Figure 4-1 illustrates potential applications of the alignment-free coupler. We envision an SAPCB comprising a polymer-laminate film bonded to an electrical PCB onto which the photonic chips are placed. The polymer film incorporates a closely-spaced, linear array of single-mode waveguides used for optical interconnections between PICs. Waveguides on the polymer film are evanescently coupled to those on the PIC over a vertical gap g.

The critical requirement of the SAPCB architecture is a board-to-PIC coupler with high efficiency and alignment tolerance. Approaches such as adiabatic coupling or edge coupling have demanding requirements ($< 5\ \mu$m) for alignment precision [107, 108, 109]; moreover, the strict alignment tolerance requires that the photonic circuit and substrate be co-designed to ensure the placement of the polymer waveguides are matched to those of the PICs with micron-scale precision.

To solve these problems, we introduce the alignment-free, "hockey stick" coupler illustrated in Fig 4-1(i). These devices consist of a PIC waveguide that runs parallel to the polymer film until taking a turn to intersect the polymer waveguides at an angle θ and length L. The angle θ is chosen to efficiently transfer optical power through the

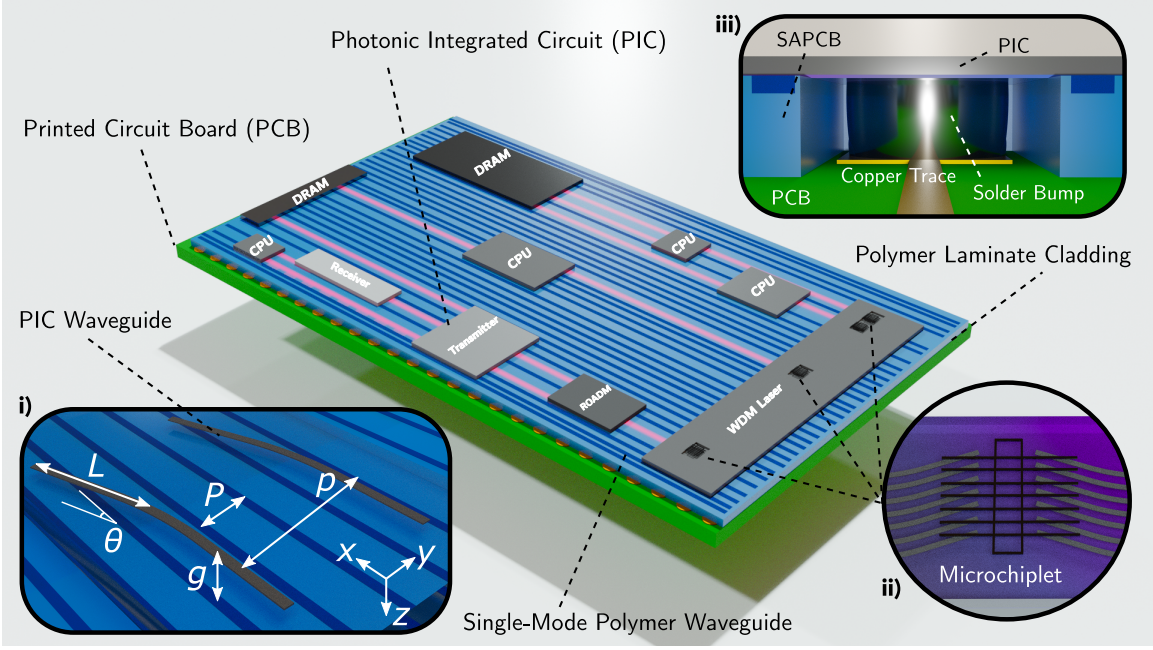

Figure 4-1: The SAPCB consists of a polymer-laminate film bonded onto an electrical PCB. PICs are flip-chip bonded to the polymer film, which includes a linear, closely-spaced array of single-mode waveguides that carry signals between chips. i): The SAPCB consists of efficient, board-level optical interconnects by making use of an alignment-free "hockey stick" coupler that intersects the polymer waveguides at an angle θ. This approach makes the efficiency of our architecture insensitive to in-plane displacements and permits coupling over a wide range of waveguide pitches. Additionally, intersecting the two waveguides at an angle eliminates the requirement to place PICs onto the SAPCB with sub-micron placement accuracy. ii) The alignment-free coupler also simplifies "pick-and-place" integration of microchiplets into PICs, which enables the introduction of gain, detectors, and single-photon sources into a single chip. iii): Electrical connections can be made in our architecture by punching holes through the polymer film, which permits bump bonding to pads on the electrical PCB.

interaction of their evanescent fields. Off-axis alignment is usually considered undesirable for optical coupling; here, our approach intentionally applies it to achieve one critical benefit: this geometry is *invariant* to any longitudinal displacement Δx and any transverse displacement $\Delta y < L \sin\theta$. Moreover, the transverse displacement tolerance can be increased arbitrarily by increasing the length of the coupler L.

Angled coupling introduces other benefits during assembly. Suppose the polymer and PIC waveguides have differing pitches P, p, respectively. No matter their respective pitches, as long as the two waveguides are coarsely aligned within $L \sin\theta$, they will always intersect at some point with no transmission penalty. The SAPCB could therefore serve as an off-the-shelf, universal connector interfacing PICs of different designs and with differing port locations. The only restriction on the polymer waveguide pitch is that P must be smaller than $L \sin\theta$, which ensures that no waveguide on the PIC couples to more than one polymer waveguide.

In addition to board-level assembly, the alignment-free coupler also enables simplified

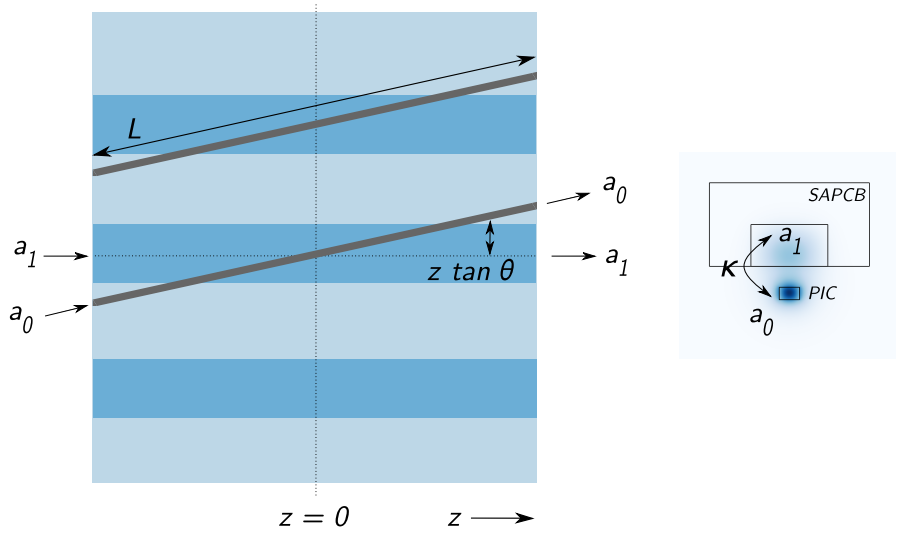

Figure 4-2: The alignment-free coupler can be modeled as two waveguides weakly coupled vertically by an evanescent interaction strength κ to one another at an off-axis angle θ. At an arbitrary point z along the propagation, the coupling constant κ will decay exponentially by the vertical offset $z \tan\theta$ with a characteristic decay length γ, i.e. $\kappa(z) = \kappa_0 e^{-\gamma|z|\tan\theta}$.

"pick-and-place" integration of microchiplets into photonic circuits. Microchiplets, which are miniaturized photonic chips integrated into larger circuits, have recently drawn interest as an approach for integrating gain [123, 124, 125, 126], photodetectors [127, 128], or single-photon sources [105] into PICs. This integration is illustrated in Figure 4-1(ii), where the PIC backbone has windows etched into the cladding for coupling chiplets to the circuit. Finally, the SAPCB is also compatible with state-of-the-art electrical interconnect technologies such as flip-chip bonding. We illustrate this compatibility in Figure 4-1(iii), which shows how holes can be punched in the polymer film to enable bump bonding between the PIC and PCB.

4.3 Theory

We begin by analyzing the dynamics of the alignment-free coupler using a coupled mode theory approach [36, 129] (Fig. 4-2). Consider two waveguides weakly coupled vertically to one another at an off-axis angle θ and intersecting at $z = 0$. When the two waveguides intersect one another, their interaction can be described by a coupling constant per unit length κ and a wavevector mismatch Δk. At an arbitrary z, Δk remains unchanged, but κ exponentially decays with the transverse offset $|z|\tan\theta$ [30]. The waveguide coupling can therefore be modeled as $\kappa(z) = \kappa e^{-\gamma|z|\tan\theta} = \kappa e^{-\gamma'|z|}$, where $\gamma' = \gamma\tan\theta$ describes the decay of κ with transverse offset per unit length.

The coupled mode equations describing the system are therefore:

$$\frac{\mathrm{d}}{\mathrm{d}z}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} -i\beta_0 & -i\kappa e^{-\gamma|z|} \\ -i\kappa e^{-\gamma|z|} & -i\beta_1 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \mathbf{C}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \tag{4.1}$$

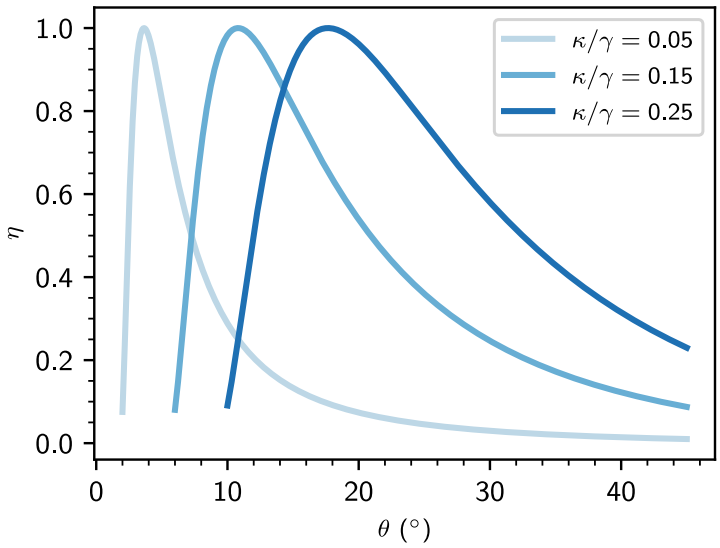

Figure 4-3: The theoretical power transfer efficiency η of the alignment-free coupler vs. θ for varying values of κ/γ. At small values of θ, η will oscillate rapidly from minimum to maximum power transfer. The alignment-free coupler should not be used in this regime and it is omitted from the plot for clarity.

$$\mathbf{C} = -i\beta\mathbf{I} + i\begin{bmatrix} \Delta & -\kappa e^{-\gamma|z|} \\ -\kappa e^{-\gamma|z|} & -\Delta \end{bmatrix} \tag{4.2}$$

The first term in $\mathbf{C}$ is a trivial phase evolution, which we disregard. We can iteratively solve the mode coupling for small displacements Δz as:

$$\begin{bmatrix} a_0(z+\Delta z) \\ a_1(z+\Delta z) \end{bmatrix} = \exp(\mathbf{C}(z)\cdot\Delta z)\begin{bmatrix} a_0(z) \\ a_1(z) \end{bmatrix} \tag{4.3}$$

Now assume that the two waveguides have the same effective modal index, i.e. $\Delta = 0$. In this case, the above expression simplifies to:

$$\begin{bmatrix} a_0(z+\Delta z) \\ a_1(z+\Delta z) \end{bmatrix} = \exp(\mathbf{C}(z)\cdot\Delta z)\begin{bmatrix} a_0(z) \\ a_1(z) \end{bmatrix} \tag{4.4}$$

$$= \begin{bmatrix} \cos\kappa\Delta z & -i\sin\kappa\Delta z \\ -i\sin\kappa\Delta z & \cos\kappa\Delta z \end{bmatrix}\begin{bmatrix} a_0(z) \\ a_1(z) \end{bmatrix} \tag{4.5}$$

This is a rotation matrix, which has the convenient property that:

$$\begin{bmatrix} \cos\kappa_1\Delta z & -i\sin\kappa_1\Delta z \\ -i\sin\kappa_1\Delta z & \cos\kappa_1\Delta z \end{bmatrix}\begin{bmatrix} \cos\kappa_2\Delta z & -i\sin\kappa_2\Delta z \\ -i\sin\kappa_2\Delta z & \cos\kappa_2\Delta z \end{bmatrix} \tag{4.6}$$

$$= \begin{bmatrix} \cos(\kappa_1+\kappa_2)\Delta z & -i\sin(\kappa_1+\kappa_2)\Delta z \\ -i\sin(\kappa_1+\kappa_2)\Delta z & \cos(\kappa_1+\kappa_2)\Delta z \end{bmatrix} \tag{4.7}$$

Therefore:

$$\begin{bmatrix} a_0(z_2) \\ a_1(z_2) \end{bmatrix} = \begin{bmatrix} \cos \int_{z_1}^{z_2} \kappa(z)\, \mathrm{d}z & -i \sin \int_{z_1}^{z_2} \kappa(z)\, \mathrm{d}z \\ -i \sin \int_{z_1}^{z_2} \kappa(z)\, \mathrm{d}z & \cos \int_{z_1}^{z_2} \kappa(z)\, \mathrm{d}z \end{bmatrix} \begin{bmatrix} a_0(z_1) \\ a_1(z_1) \end{bmatrix} \tag{4.8}$$

For initial conditions $a_0(-\infty) = 1, a_1(-\infty) = 0$, we get that:

$$|a_1(\infty)|^2 = \sin^2 \left(\int_{-\infty}^{\infty} \kappa e^{-\gamma |z|}\, \mathrm{d}z \right) = \sin^2 \left(\frac{2\kappa}{\gamma} \right) = \sin^2 \left(\frac{2\kappa}{\gamma' \tan \theta} \right) \tag{4.9}$$

Figure 4-3 plots the theoretical transmission efficiency η vs. angle θ for varying values of κ/γ. In addition to potentially arbitrary lateral tolerance, depending on the value of L, the alignment-free coupler has high angular tolerance. This coupling scheme therefore has two major advantages over conventional optical couplers:

- **High angular tolerance**: The $1/\tan\theta$ dependence of η produces a large angular tolerance $\Delta\theta = (4/3)\theta_{\text{opt}}$. Moreover, η has a long tail that ensures modest coupling even at very large angular errors, greatly simplifying initial alignment. Coupling the waveguides more strongly (increasing κ/γ) further increases $\Delta\theta$.

- **Robust design**: no matter the values of κ, γ, the coupling efficiency reaches unity at some angle. Fabrication-induced variation in κ can therefore *always* be corrected during alignment. No matter the design, the angled coupler allows efficient power transfer by rotating one waveguide relative to the other. By contrast, errors in κ from the designed value reduce the efficiency of conventional adiabatic and directional couplers, and these errors *cannot be corrected after fabrication*.

4.4 Mismatched couplers: what's the efficiency?

We assumed in the derivation above that the two waveguides have identical effective indices, and thus that they are phase-matched, i.e. $\Delta = 0$. This behavior can be analogized to resonant driving of a two-level system; however, in practice, the two waveguides might have slightly different effective indices due to fabrication error. This is even more likely to be the case when the two waveguides are fabricated in different photonic platforms. In this section, we analyze the efficiency when $\Delta \neq 0$.

To solve this, we borrow a strategy from time-dependent perturbation theory in quantum mechanics. The matrix $\mathbf{C}$ above can be analogized to the Hamiltonian of a two-level system, which can be separated into spatially-independent and dependent components:

$$\hat{H} = \hat{H}_0 + \hat{H}(z) = \begin{bmatrix} \beta_0 & 0 \\ 0 & \beta_1 \end{bmatrix} + \begin{bmatrix} 0 & \kappa e^{-\gamma |z|} \\ \kappa e^{-\gamma |z|} & 0 \end{bmatrix} \tag{4.10}$$

The dynamics of the system are governed by:

$$i \frac{\mathrm{d}}{\mathrm{d}z} |\mathbf{a}(z)\rangle = (\hat{H}_0 + \hat{H}(z)) |\mathbf{a}(z)\rangle \tag{4.11}$$

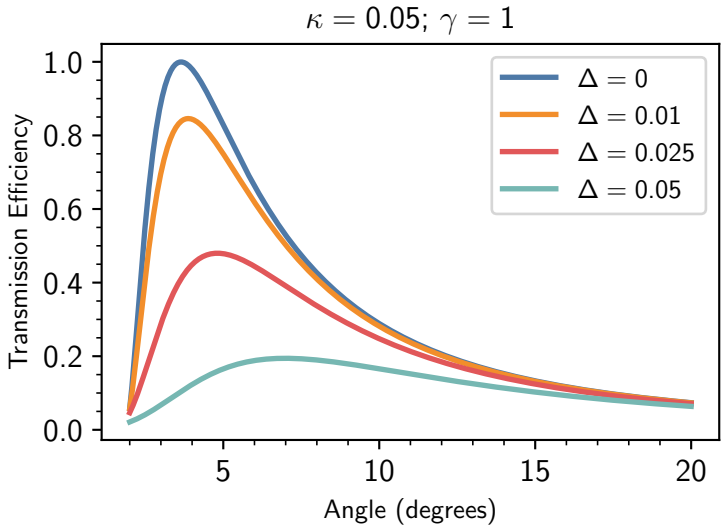

Figure 4-4: Angular dependence of coupling efficiency for varying levels of Δ when $\kappa = 0.05$; $\gamma = 1$.

and the state evolves as $e^{i\hat{H}z}\,|\mathbf{a}(z)\rangle$, which is identical to the coupled mode formulation presented earlier.

Now suppose the system is initialized at $z = -\infty$, where the coupling between waveguides is zero, into one of the eigenstates of $\hat{H}_0$—namely $a_0 = 1, a_1 = 0$. We can switch to the interaction picture:

$$|\tilde{\mathbf{a}}(z)\rangle = e^{i\hat{H}_0 z}\,|\mathbf{a}(z)\rangle \tag{4.12}$$

which simplifies the equation of evolution to:

$$i\frac{\mathrm{d}}{\mathrm{d}z}\,|\tilde{\mathbf{a}}(z)\rangle = \tilde{H}(z)\,|\tilde{\mathbf{a}}(z)\rangle \tag{4.13}$$

where:

$$\tilde{H}(z) = e^{i\hat{H}_0 z} H(z) e^{-i\hat{H}_0 z} = \begin{bmatrix} 0 & e^{-2i\Delta z}\kappa e^{-\gamma|z|} \\ e^{2i\Delta z}\kappa e^{-\gamma|z|} & 0 \end{bmatrix} \tag{4.14}$$

From this, if we expand $|\tilde{\mathbf{a}}(z)\rangle$ into the uncoupled basis a_0, a_1, we get the coupled equations:

$$i\frac{\mathrm{d}}{\mathrm{d}z}a_0(z) = \kappa e^{-2i\Delta z - \gamma|z|}a_1(z) \tag{4.15}$$

$$i\frac{\mathrm{d}}{\mathrm{d}z}a_1(z) = \kappa e^{2i\Delta z - \gamma|z|}a_0(z) \tag{4.16}$$

Now, we apply a perturbative expansion by introducing a unit-free parameter λ, such that:

$$\hat{H} = \hat{H}_0 + \lambda\hat{H}(z) \tag{4.17}$$

$$i\frac{\mathrm{d}}{\mathrm{d}z}\,|\tilde{\mathbf{a}}(z)\rangle = \lambda\tilde{H}(z)\,|\tilde{\mathbf{a}}(z)\rangle \tag{4.18}$$

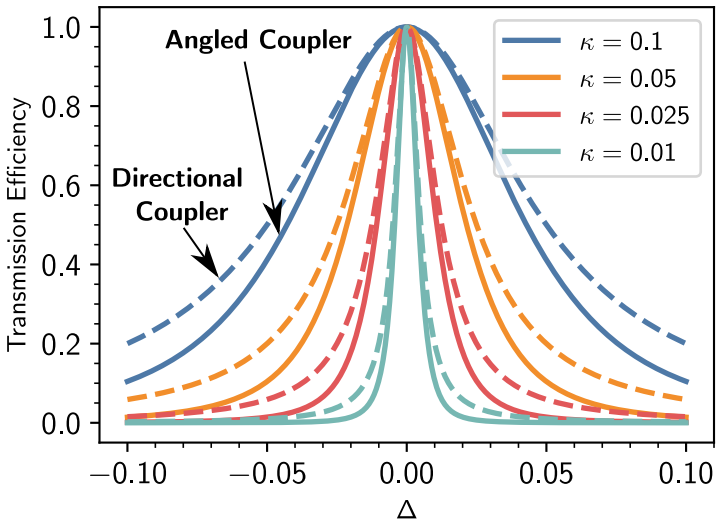

Figure 4-5: Phase-matching bandwidth of angular (solid line) and directional (dashed line) coupler for varying levels of κ

We expand $|\tilde{\mathbf{a}}(z)\rangle$ in powers of the parameter λ:

$$|\tilde{\mathbf{a}}(z)\rangle = |\tilde{\mathbf{a}}^{(0)}(z)\rangle + \lambda\,|\tilde{\mathbf{a}}^{(1)}(z)\rangle + \lambda^2\,|\tilde{\mathbf{a}}^{(2)}(z)\rangle + \ldots \tag{4.19}$$

Substituting into equation (4.18), we get that:

$$i\partial_z\,|\tilde{\mathbf{a}}^{(0)}(z)\rangle + i\partial_z\lambda\,|\tilde{\mathbf{a}}^{(1)}(z)\rangle + i\partial_z\lambda^2\,|\tilde{\mathbf{a}}^{(2)}(z)\rangle + \ldots = \lambda\tilde{H}(z)\,|\tilde{\mathbf{a}}^{(0)}(z)\rangle + \lambda^2\tilde{H}(z)\,|\tilde{\mathbf{a}}^{(1)}(z)\rangle + \ldots \tag{4.20}$$

Equating terms with the same power of λ, we find that:

$$i\partial_z\,|\tilde{\mathbf{a}}^{(0)}(z)\rangle = 0 \tag{4.21}$$

$$i\partial_z\,|\tilde{\mathbf{a}}^{(1)}(z)\rangle = \tilde{H}(z)\,|\tilde{\mathbf{a}}^{(0)}(z)\rangle \tag{4.22}$$

$$i\partial_z\,|\tilde{\mathbf{a}}^{(2)}(z)\rangle = \tilde{H}(z)\,|\tilde{\mathbf{a}}^{(1)}(z)\rangle \tag{4.23}$$

and so on. Now, since the coupling is "off" at $z = -\infty$, we know that:

$$|\mathbf{a}(-\infty)\rangle = |\tilde{\mathbf{a}}(-\infty)\rangle = |\tilde{\mathbf{a}}^{(0)}(-\infty)\rangle + \lambda\,|\tilde{\mathbf{a}}^{(1)}(-\infty)\rangle + \lambda^2\,|\tilde{\mathbf{a}}^{(2)}(-\infty)\rangle + \ldots \tag{4.24}$$

which provides the initial conditions:

$$|\tilde{\mathbf{a}}^{(0)}(-\infty)\rangle = |\mathbf{a}(-\infty)\rangle \tag{4.25}$$

$$|\tilde{\mathbf{a}}^{(1)}(-\infty)\rangle = 0 \tag{4.26}$$

$$|\tilde{\mathbf{a}}^{(2)}(-\infty)\rangle = 0 \tag{4.27}$$

We found earlier that $|\tilde{\mathbf{a}}^{(0)}\rangle$ is spatially-independent; therefore:

$$|\tilde{\mathbf{a}}^{(0)}(z)\rangle = |\tilde{\mathbf{a}}^{(0)}(-\infty)\rangle = |\mathbf{a}(-\infty)\rangle \tag{4.28}$$

Using this result, we can solve for $|\tilde{\mathbf{a}}^{(1)}\rangle$:

$$i\partial_z |\tilde{\mathbf{a}}^{(1)}(z)\rangle = \tilde{H}(z)\,|\mathbf{a}(-\infty)\rangle \tag{4.29}$$

$$|\tilde{\mathbf{a}}^{(1)}(z)\rangle = \int_{-\infty}^{z} -i\tilde{H}(z)\,|\mathbf{a}(-\infty)\rangle \;\mathrm{d}z \tag{4.30}$$

$|\tilde{\mathbf{a}}^{(2)}\rangle$ can then be found with a nested integral expression, as can all higher-order terms:

$$|\tilde{\mathbf{a}}^{(2)}(z)\rangle = \int_{-\infty}^{z} -i\tilde{H}(z_2)\left(\int_{-\infty}^{z_2} -i\tilde{H}(z_1)\,|\mathbf{a}(-\infty)\rangle \;\mathrm{d}z_1\right)\mathrm{d}z_2 \tag{4.31}$$

We are looking for a coupling efficiency, which can be analogized to a transition probability from eigenstate $0 \to 1$. This is:

$$\eta = P_{0\to1}(z) = |\langle 1|\mathbf{a}(z)\rangle|^2 = |\langle 1|e^{-i\hat{H}_0 z}\tilde{\mathbf{a}}(z)\rangle|^2 = |\langle 1|\tilde{\mathbf{a}}(z)\rangle|^2 \tag{4.32}$$

$$= |\langle 1|\left(|\tilde{\mathbf{a}}^{(0)}(z)\rangle + |\tilde{\mathbf{a}}^{(1)}(z)\rangle + |\tilde{\mathbf{a}}^{(2)}(z)\rangle + ...\right)|^2 \tag{4.33}$$

$$= |\langle 1|\tilde{\mathbf{a}}^{(1)}(z)\rangle + \langle 1|\tilde{\mathbf{a}}^{(2)}(z)\rangle + ...|^2 \tag{4.34}$$

We can now explicitly compute the expansion of $|\tilde{\mathbf{a}}(z)\rangle$:

$$|\tilde{\mathbf{a}}^{(0)}(z)\rangle = |0\rangle \tag{4.35}$$

$$|\tilde{\mathbf{a}}^{(1)}(z)\rangle = \begin{cases} -\frac{ie^{(2i\Delta+\gamma)z}}{2i\Delta+\gamma}\kappa\,|1\rangle & \text{if } z<0 \\ \left(\frac{1-e^{(2i\Delta-\gamma)z}}{i\gamma+2\Delta} - \frac{i}{2i\Delta+\gamma}\right)\kappa\,|1\rangle & \text{if } z>0 \end{cases} \tag{4.36}$$

$$|\tilde{\mathbf{a}}^{(1)}(\infty)\rangle = -\frac{2i\gamma\kappa}{\gamma^2+4\Delta^2}\,|1\rangle \tag{4.37}$$

$$|\tilde{\mathbf{a}}^{(2)}(\infty)\rangle = -\frac{2\kappa^2(\gamma+i\Delta)}{\gamma(\gamma+2i\Delta)^2}\,|0\rangle \tag{4.38}$$

$$|\tilde{\mathbf{a}}^{(3)}(\infty)\rangle = \frac{12i\gamma\kappa^3}{40\gamma^2\Delta^2+9\gamma^4+16\Delta^4}\,|1\rangle \tag{4.39}$$

We can keep expanding terms to obtain a better approximation. The conversion efficiency is therefore:

$$\eta = \left(\frac{2\gamma\kappa}{\gamma^2+4\Delta^2} - \frac{12\gamma\kappa^3}{40\gamma^2\Delta^2+9\gamma^4+16\Delta^4} + ...\right)^2 \tag{4.40}$$

Recall that we define $\gamma = \gamma' \tan\theta$, where γ' is the decay parameter of the material system. To first order, the coupling efficiency is $\eta = 4\gamma^2\kappa^2/(\gamma^2+4\Delta^2)^2$. Notice that this corresponds to the first term of the Taylor expansion of $\eta = \sin^2(2\kappa/\gamma)$ when $\Delta = 0$. This is an excellent estimate when $\Delta > \kappa$ or when the transfer efficiency is low, but is

less accurate for high coupling efficiency. Including higher order terms will improve the accuracy of the estimate.

Figures 4-4 and 4-5 show the phase matching bandwidth calculated using up to the seventh-term of the expansion. Note that in general, the phase-matching bandwidth is narrower for an angled coupler relative to a conventional directional coupler. By more strongly coupling the waveguides, we can increase the phase-matching bandwidth and reduce the fabrication dependence.

The theory here suggests that we can couple two optical modes with exceptionally high alignment tolerance–in principle unlimited lateral tolerance, and fairly high angular tolerance. The price we pay, however, is a more stringent requirement on the fabrication tolerance of the waveguide. Practically speaking, however, it is preferable to trade fabrication tolerance for alignment tolerance. Optical alignment is quite imprecise–on the order of single μm for typical photonic assembly tools, and potentially tens of μm if we wish to transition to high-volume pick and place tools. Photonic fabrication, using photolithography, however, can be extremely precise–especially waveguide layers, which need to be precise on the order of tens to hundreds of nanometers to maintain uniform component performance across a wafer. Thus, our approach shifts the burden of high precision to tools that are already capable of reaching these requirements, while relaxing precision requirements for the high-volume tools required later in the assembly process.

4.5 Simulation

To validate this approach, we conducted 3D finite-difference time-domain simulations (Ansys Lumerical FDTD) of an example implementation of the SAPCB shown in Figure 4-6 and using the parameters in Table 4.1. These simulations assumed high-index single-mode polymer core SAPCB waveguides embedded in a low-index fluoropolymer cladding, and silicon nitride (SiN) PIC waveguides in silicon dioxide cladding. SiN is a high-index contrast waveguide platform transparent over the visible and infrared and is available in most silicon photonics and CMOS foundries [130]. The simulations assumed a wavelength $\lambda = 1550$ nm, and the optimized design exhibits less than 0.2 dB insertion loss.

Figure 4-6a plots the effective mode index mismatch $\Delta n = n_{\text{SAPCB}} - n_{\text{PIC}}$ as a function of the SiN and polymer waveguide widths. Efficient mode transfer between the waveguides requires matching their propagation constants by engineering their geometry. This requirement dominates the fabrication-induced error. Figures 4-6b-f plot the effect on transmission caused by errors in SiN width (b), SiN height (c), coupling gap (d), wavelength (e), and temperature (f). The coupler is remarkably robust to changes in all of these parameters, exhibiting less than 0.5 dB penalty for a $\pm$ 20 nm variation in waveguide dimensions and lower than 0.3 dB excess loss for a $\pm$ 100 nm change in the coupling gap g. Moreover, it has a 1-dB optical bandwidth in excess of 180 nm and exhibits less than 0.5 dB temperature sensitivity over a range of 80° C. We obtained the results in Figure 4-6 from FDTD simulations of the full structure. Fig. 4-7g shows these simulations for the parameters of Table 4.1, illustrating the power transfer from the SiN waveguide, through the alignment-free coupler, and into the polymer waveguide. Cross sectional field intensity plots along the structure are shown underneath.

Table 4.1: Simulation parameters for the alignment-free photonic coupler.

Polymer core n [131]	1.575
Polymer cladding n [132]	1.34
Polymer core $\mathrm{d}n/\mathrm{d}T$ [131]	-1.1×10^{-4} /°C
Polymer cladding $\mathrm{d}n/\mathrm{d}T$ [132]	-5×10^{-5} /°C
PIC waveguide core n [133]	2
PIC waveguide cladding n [133]	1.445
PIC core $\mathrm{d}n/\mathrm{d}T$ [133]	2.51×10^{-5} /°C
PIC cladding $\mathrm{d}n/\mathrm{d}T$ [133]	9.6×10^{-6} /°C
SiN width	462.5 nm
Polymer width	1.6 μm
SiN height	300 nm
Polymer height	1 μm
Gap (g)	1 μm
Length (L)	100 μm
θ_{opt}	4.4°
Wavelength (λ)	1550 nm

Figure 4-8 plots η vs. θ for the same structure. The waveguide intersection causes a scattering loss of $\sim$0.2 dB at the optimal coupling angle θ. Upon correcting for this loss, η agrees well with the expression in Eq. 4.9 around this region and exhibits an angular alignment (3 dB) tolerance $\Delta\theta > 5$ degrees. Additionally, η rolls off slowly for $\theta > \theta_{\mathrm{opt}}$, permitting modest coupling efficiencies at even large angular errors. This greatly simplifies initial alignment and relaxes the required precision of alignment during packaging.

The scattering loss shown in Figure 4-8 results primarily from a faster-than-adiabatic transition at the waveguide intersection and increases with θ as the transition into the hybridized modes becomes more abrupt [134, 135]. The scattering loss drops with increasing g, which makes the transition more adiabatic. Increasing g introduces two tradeoffs, however: the angular tolerance $\Delta\theta$ will drop, and the transmission will be more sensitive to errors in Δk. If higher insertion losses are acceptable, the waveguides can be coupled more strongly, which improves $\Delta\theta$. We also show in Figure 4-8 such an example, where we decreased g to 500 nm. κ/γ, and therefore the tolerance, $\Delta\theta$, nearly doubles, but at the expense of a higher insertion loss of 1 dB. The tradeoffs between insertion loss, robustness to fabrication error, and $\Delta\theta$ bound an optimal range for κ and therefore g.

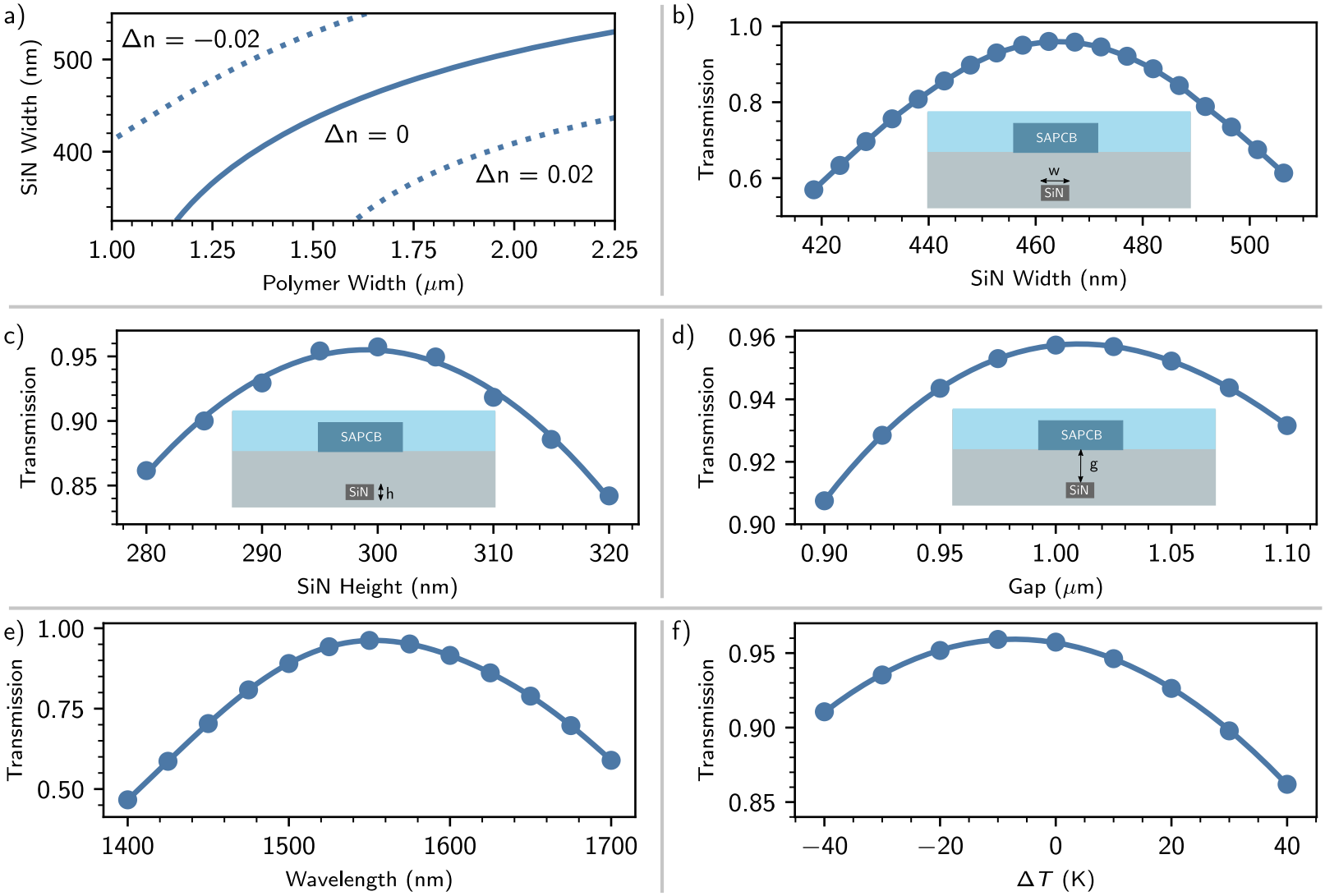

Figure 4-6: a) Effective mode index mismatch $\Delta n = n_{\text{SAPCB}} - n_{\text{PIC}}$ as a function of the PIC (SiN) waveguide width and the SAPCB (polymer) waveguide width. The two waveguide geometries should be engineered such that their modes have equal propagation constants, i.e. $\Delta k = 2\pi\Delta n/\lambda = 0$. b-f) Power transfer efficiency as a function of the PIC waveguide width (b), PIC waveguide height (c), coupling gap (d), wavelength (e), and temperature (f) for the design with parameters in Table I.

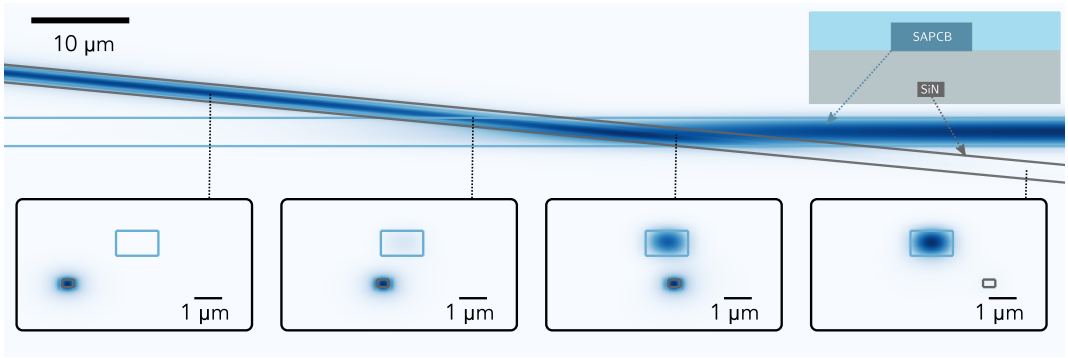

Figure 4-7: The field profile of the alignment-free coupler with parameters in Table I. The insets below show the cross-sectional field profile at varying points along the propagation.

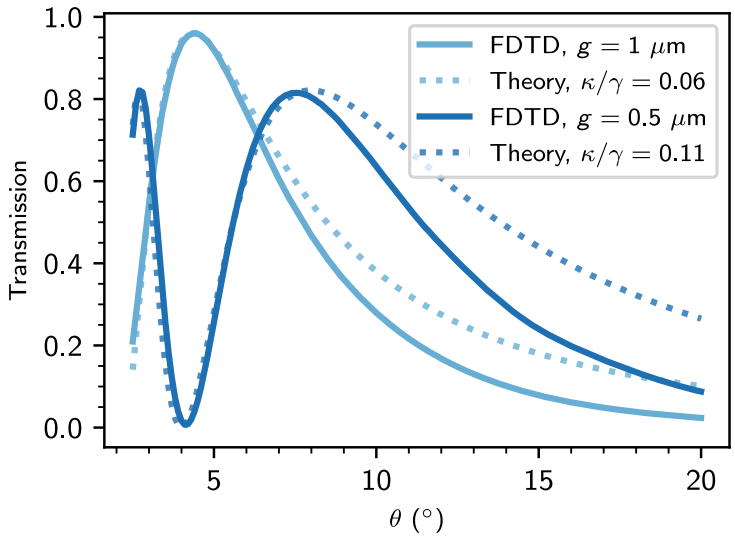

Figure 4-8: Power transfer efficiency η vs θ for designs with coupling gap $g = 1$ μm and $g = 0.5$ μm. The solid lines indicate FDTD simulation results, while the dotted lines are fit to equation (4.9).

Insertion losses can also be reduced by employing higher-index platforms that enable stronger vertical confinement of the optical mode. Figure 4-9 shows such an example, where we design a coupler to interface a 500 × 220 nm silicon photonic waveguide to a 640 × 300 nm indium phosphide (InP) waveguide ($g = 150$ nm) for hybrid integration of gain. Silicon and InP have much higher refractive indices ($n_{\text{Si}} = 3.47$; $n_{\text{InP}} = 3.17$) and therefore confine the optical mode more strongly; as a result, the optical mode is significantly less perturbed by the introduction of the other waveguide at the intersection. As Figure 4-9a shows, this enables efficient mode transfer *with no insertion loss*; moreover, we find a near-exact fit to theory. Additionally, despite the high index contrast of both the Si and InP photonic platforms, which would imply they are strongly dispersive, we find that our optimized coupler has a 1-dB optical bandwidth exceeding 230 nm (Fig. 4-9b).

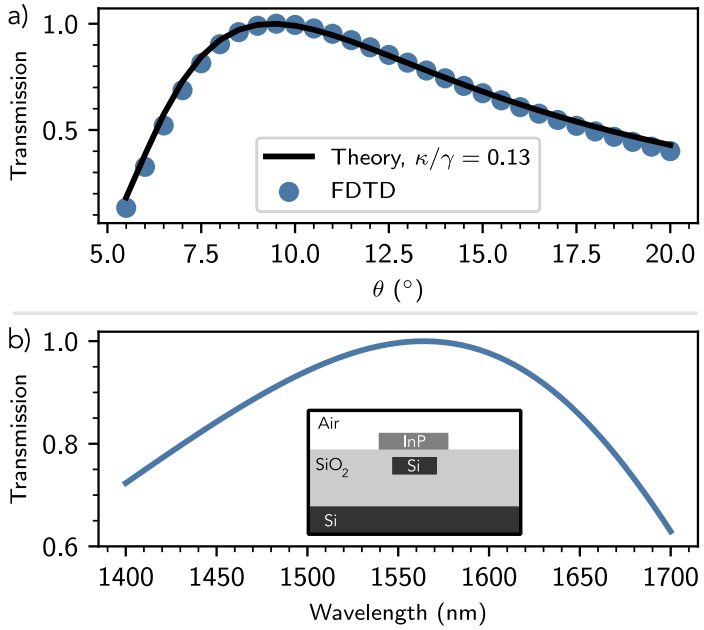

Figure 4-9: a) Transmission vs. θ for an alignment-free coupler designed to interface a 640 $\times$ 300 nm InP gain microchiplet to a 500 $\times$ 220 nm silicon photonic waveguide. The strong mode confinement in both materials eliminates scattering loss at the intersection, permitting mode transfer with no insertion loss. As a result, the transmission characteristic reproduces nearly perfectly equation (4.9). b) The transmission efficiency as a function of wavelength. The coupler has a 1-dB bandwidth exceeding 230 nm.

4.6 Discussion

In Figure 4-10, we compare the lateral and angular alignment tolerance of the alignment-free coupler to a 10 μm inverse tapered edge coupler and a tapered adiabatic coupler. For each approach, we compare the 1-dB coupling efficiency contour in the $\delta r_{||}$-$\delta\theta$ plane to that of the alignment-free coupler.

Optical coupling is typically non-perturbative: an optical beam is launched into free space from the end facet of a fiber, where it is then coupled into an inverse waveguide taper or grating coupler designed to be mode-matched to the beam. Mode-matching imposes strict requirements on the relative position of the coupler and the fiber facet, as the coupling efficiency is directly related to the overlap integral between the two modes. We assume perfect mode-matching, i.e. perfect alignment yields unity coupling.

The electric field of a Gaussian beam propagating in z with waist w_0 can be written as [136]:

$$\mathbf{E}(x,y,z) = E_0 \frac{1}{\sqrt{1+\frac{z^2}{z_R^2}}} \exp\left(-\frac{x^2+y^2}{w_0^2\left(1+\frac{z^2}{z_R^2}\right)}\right) \times$$

$$\exp\left(-i\left(kz + k\frac{x^2+y^2}{2z\left(1+\frac{z_R^2}{z^2}\right)} - \arctan\frac{z}{z_R}\right)\right) \tag{4.41}$$

where z_R is the Rayleigh length $\pi w_0^2 n/\lambda$. The mode overlap integral is conventionally [35]:

$$\eta = \frac{|\int E_1^* E_2 \, \mathrm{d}A|^2}{\int |E_1|^2 \, \mathrm{d}A \int |E_2|^2 \, \mathrm{d}A} \tag{4.42}$$

As the output at the fiber facet is located at the beam waist (i.e. $z = 0$), the expression above simplifies to:

$$\mathbf{E}(x, y, z) = E_0 \exp\left(-\frac{x^2+y^2}{w_0^2}\right) \tag{4.43}$$

Suppose we have a lateral misalignment δx of one beam relative to the other. The mode overlap integral η can be easily calculated to be:

$$\eta(\delta x) = \exp\left(-\frac{\delta x^2}{w_0^2}\right) \tag{4.44}$$

We find a lateral tolerance $\Delta x = w_0$.

Now consider an angular misalignment $\delta\theta$. Suppose, without loss of generality, that the rotation of the fiber is about the y-axis at $x = 0$. If $\delta\theta$ is small, such that $x \tan\delta\theta/z_R \ll 1$, we can neglect amplitude distortion and phase induced by the beam curvature. We then find that:

$$\mathbf{E}(x, y, z) = E_0 \exp\left(-\frac{x^2+y^2}{w_0^2}\right) \exp\left(-i\left(kx\tan\delta\theta - \arctan\frac{x\tan\delta\theta}{z_R}\right)\right) \tag{4.45}$$

$$\approx E_0 \exp\left(-\frac{x^2+y^2}{w_0^2}\right) \exp\left(-i\left(kx\tan\delta\theta - \frac{x\tan\delta\theta}{z_R}\right)\right) \tag{4.46}$$

Having made these simplifications, the overlap integral is now tractable:

$$\eta(\delta\theta) = \exp\left(-\frac{w_0^2\tan^2(\delta\theta)(kz_R-1)^2}{4z_R^2}\right) \tag{4.47}$$

The angular tolerance, if we approximate $\tan\delta\theta \approx \delta\theta$, is therefore:

$$\Delta\theta = \frac{2z_R}{w_0(kz_R-1)} \tag{4.48}$$

We observe that there is a tradeoff between lateral and angular tolerance:

$$\Delta x\Delta\theta = \frac{2z_R}{kz_R-1} \tag{4.49}$$

In practice, $kz_R = 2\pi^2(nw_0/\lambda)^2 \gg 1$. Simplifying further, the tradeoff is revealed to

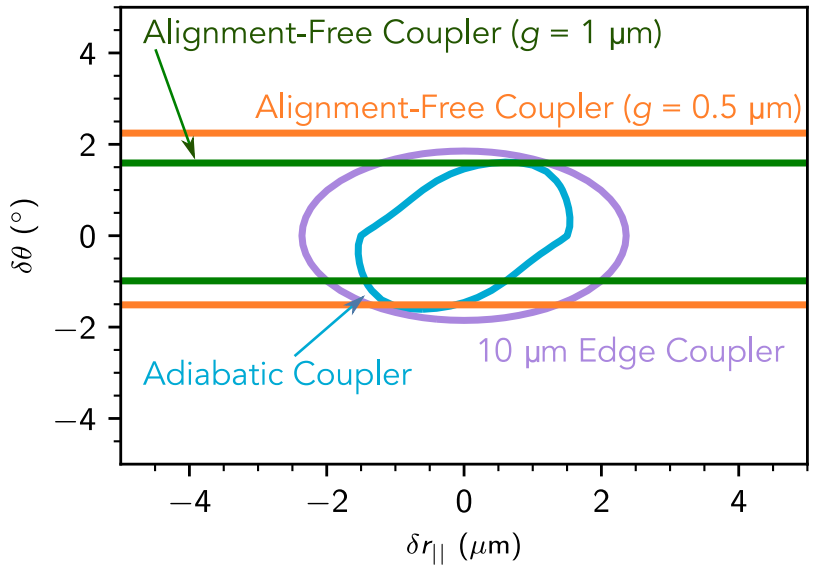

Figure 4-10: Lateral and angular alignment tolerance of the alignment-free coupler compared to inverse tapered edge couplers and tapered adiabatic couplers. The lines indicate the 1-dB coupling efficiency contour as a function of in-plane displacement $\delta r_{||}$ and angular displacement $\delta\theta$. The alignment-free coupler has a combined alignment tolerance $\Delta r_{||}\Delta\theta$ that exceeds current approaches.

be fundamental:

$$\Delta x \Delta\theta = \frac{2z_R}{kz_R - 1} \approx \frac{2z_R}{kz_R} = \frac{\lambda}{\pi n} \tag{4.50}$$

This tradeoff does not apply to the alignment-free coupler, which has both a high angular tolerance and an arbitrarily high lateral tolerance that can be increased by increasing L. As a result, the combined lateral and angular tolerance $\Delta r_{||}\Delta\theta$ of our approach exceeds the fundamental limit on alignment tolerances for edge coupling. Expanding or contracting the beam size improves the alignment tolerance of edge coupling in one dimension at the expense of the other; thus, no possible edge coupler can have *both* superior lateral and superior angular tolerance to that of an alignment-free coupler.

Adiabatic couplers, on the other hand, taper one or both waveguides to induce an avoided crossing between the two eigenmodes, which adiabatically transfers power from one waveguide to the other [36, 137]. This adiabatic transition makes the devices robust to variation in Δk, which has led to them being favored in many photonic platforms for their resilience to fabrication error. This robustness comes at the cost of alignment tolerance, however, as small lateral or angular errors render the interaction non-adiabatic, resulting in little or no power transfer. To compare relative tolerances, we designed an adiabatic coupler to transfer power from SiN to the polymer waveguide; our coupler linearly tapers the SiN waveguide width from 550 to 320 nm over a length of 200 μm ($g = 1$ μm) and achieves an efficiency of 96%, which is comparable to our optimized alignment-free coupler.

Figure 4-10 shows the transmission penalty of the adiabatic coupler as a function of

misalignment $\delta r_{||}, \delta\theta$; while $\Delta\theta$ is comparable to an alignment-free coupler of the same length, $\Delta r_{||}$ is far smaller. For sufficiently long tapers, it is also possible to achieve high coupling efficiency at a large angular error $\delta\theta$, where the taper acts effectively as an alignment-free coupler. We omit this region in Figure 4-10, as the coupling in this regime is non-adiabatic. Moreover, as the adiabatic coupler is tapered, unlike the alignment-free coupler, the lateral tolerance $\Delta r_{||}$ in this regime is far smaller.

Our results show that the combined lateral and angular tolerance $\Delta r_{||}\Delta\theta$ of our approach is higher than that of conventional optical couplers. We achieve this by making use of evanescent coupling, which does not suffer from a fundamental limitation on $\Delta r_{||}\Delta\theta$, and by intentionally engineering a system largely invariant to lateral displacements. This lateral tolerance is maximized by choosing both waveguides to not be tapered; as a result, $\Delta r_{||}$ can be arbitrarily high. While our approach does require the effective indices of the waveguides to be matched, our simulation results suggest that the dimensional tolerances needed to achieve this are well below what is realizable in current fabrication processes.

4.7 System Integration and Outlook

We show in Figure 4-11 two possible applications of the SAPCB for system-level integration. In Figure 4-11a, we consider interfacing two photonic circuits with different waveguide pitches and process stacks. As the alignment-free coupler interfaces with the polymer waveguides at an angle, the off-axis intersection guarantees that both waveguides can couple into the same waveguide on the board. The two PICs may also not have the same process stack; for instance, one process may require a larger oxide layer, resulting in a larger coupling gap g to the polymer waveguide. This can be addressed in our approach by simply modifying the coupling angle θ to preserve efficient power transfer.

Figure 4-11b demonstrates another advantage originating from the need for $\Delta k \approx 0$ for efficient coupling. Suppose a waveguide on a PIC needs to be routed over a polymer waveguide with minimal crosstalk. By engineering the dimensions of the PIC waveguide, one can ensure a strong wavevector mismatch Δk with the polymer waveguide, allowing for crosstalk-free transmission of signals over many photonic components on a board.

We envision that the alignment-free coupler would be defined on the PIC, where advanced photolithography processes define the required waveguide geometry and angle precisely. This frees the SAPCB to consist only of linear arrays of polymer waveguides, with no bends or tapering required. The simple layout of the polymer board could potentially allow it to be fabricated by fiber pulling approaches from a preform, rather than more costly lithography processes. Additionally, polymer waveguides have a wide transparency window, making the SAPCB applicable to photonics operating in both the visible and near-infrared.

4.8 Conclusion

We have presented a self-aligning photonic circuit board capable of serving as a universal connector for optoelectronic system integration. The critical element of the SAPCB

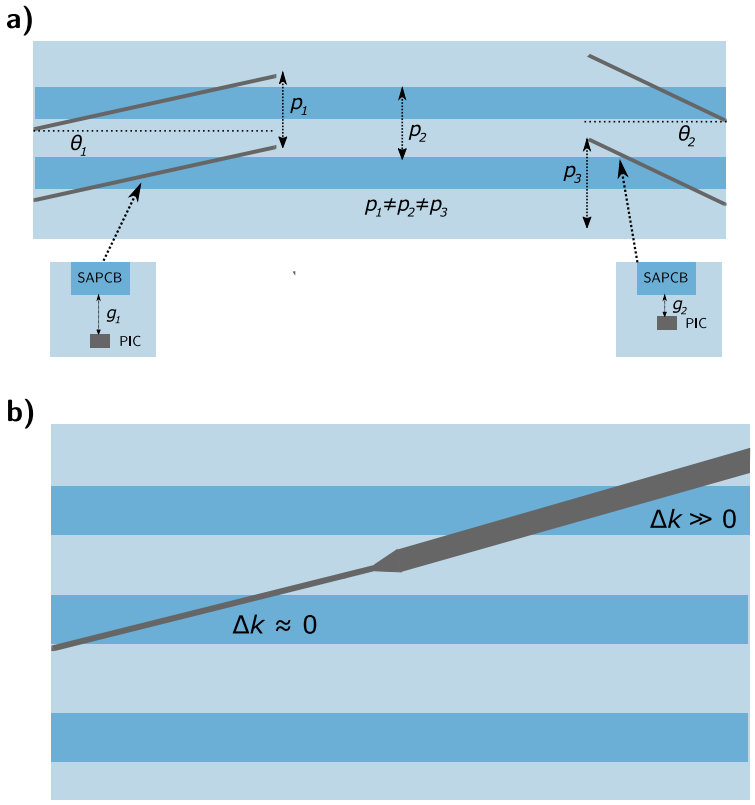

Figure 4-11: a) The alignment-free coupler can interface photonic circuits with differing waveguide pitches and process stacks. As the waveguides interact at an angle, precise matching of the waveguide pitch is not necessary. By varying the coupling angle θ, one can easily optimize the transmission for any coupling gap g. b) The requirement for phase-matching permits simplified routing with minimal crosstalk. By tapering the waveguide to ensure $\Delta k \gg 0$, waveguides can be routed over one another with negligible crosstalk.

is the alignment-free coupler, which engineers a laterally invariant system insensitive to the exact location of waveguides on the photonics, and which also exhibits high angular tolerance and arbitrarily high lateral tolerance. Our approach is robust to variations in the device geometry, and we show its combined lateral and angular tolerance exceeds that of conventional optical coupling approaches. The SAPCB allows for system integration with minimal design and alignment requirements, enabling a wide range of photonic components to interface with one another and simplifying the assembly of complex optical systems.

5

Single chip photonic neural network processors

This chapter is adapted from work[1] reported in ref. [138].

5.1 Introduction

Deep neural networks (DNNs) have revolutionized machine learning, enabling state-of-the-art performance on computation ranging from image classification [5, 6], natural language processing [7], games [8, 9], and chip design [10]. However, as these models scale to trillions of parameters, energy consumption and throughput have begun to emerge as major bottlenecks in digital electronics.

This is problematic, as a primary reason why the latest DNNs perform so well is their massive size. Further performance gains in machine learning will therefore depend on hardware that is able to scale with the size of these models. This has driven a search for new hardware architectures that are specially optimized for artificial intelligence, including electronic systolic arrays such as the Google tensor processing unit (TPU) [13], memristor crossbar arrays [139], and photonic accelerators.

As we have discussed in earlier chapters, optical systems are particularly promising for DNN accelerators. DNNs require massive amounts of computation that is mostly comprised of linear algebra. Optical systems can perform linear matrix operations at exceptionally high rate and efficiency [140], motivating recent demonstrations of low latency linear algebra [15, 141, 18, 17, 142] and optical energy consumption [143] below a photon per multiply-accumulate operation.

A system that processes a deep neural network entirely in optics would be particularly optimal for applications that require processing data natively in the optical domain and with ultra-low latency. While current electronic accelerators can achieve high throughput, they often do so through techniques such as batching that increase the inference latency (i.e. time between an input to the DNN and the output), making them unsuitable for applications that require real-time inference. Such applications can include:

[1] This was a major experimental effort that benefitted from the contributions of many talented collaborators. I will acknowledge their contributions throughout this chapter.

- Self-driving cars, which making split-second decisions by processing from LiDAR sensor data; [144]

- Scientific research in astronomy [145, 146] and particle physics [147], which generate massive amounts of data that require near-instantaneous classification; and

- "Smart" optical transceivers that rely on machine learning to receive, process, and route data at line rates exceeding hundreds of gigabits per second [148].

Applications such as these, which are latency-constrained, would benefit from real-time inference and training directly on optical signals, eliminating the need for slow and energetically-expensive optical-to-electrical conversions. While mapping matrix operations to optical hardware has been relatively straightforward [15], implementing all of the computation (both linear and nonlinear operations) for DNNs in optics has proven far more difficult. As a result, co-integrating optical linear and nonlinear computing units into an end-to-end photonic DNN processor has remained an outstanding challenge.

In this chapter, we report the realization of this goal in a fully-integrated, coherent optical neural network (FICONN) that performs coherent optical inference and training of DNNs on a single chip. Our system, which was fabricated in a commercial CMOS process, integrates multiple reconfigurable optical processing units for matrix algebra and nonlinear functions on-chip to implement a three-layer DNN. The system latency, limited by time-of-flight, is less than 500 ps, unlocking new applications that require ultra-fast, coherent processing of optical signals.

5.2 Architecture

Figure 5-1 shows the architecture of a coherent optical deep neural network processor[2]. Input data for classification is modulated onto the optical field by a bank of transmitter channels. Each layer of the device optically computes matrix-vector products, representing synaptic connections in the nural network, and then applies an optical nonlinear activation function, representing the action potential (firing threshold) of a neuron.

In principle, matrix-vector products can be computed passively through optical interference in a programmable photonic circuit (which applies a defined unitary weight matrix U). Following the computation of the first layer, the optical output signals are transmitted directly into the following layer. Each layer of the circuit feeds directly into the following layer, implementing a deep neural network entirely in optics and with ultra-low latency limited by the time-of-flight through the system. The output of the system can be

[2]An integrated photonic system that processes an entire DNN in optics was proposed by Shen and Harris in 2017 in ref. [15]. That experiment, which was performed on a first-generation programmable nanophotonic processor (PNP) developed within our group, showed that matrix-vector products for DNNs could be computed with high accuracy in coherent silicon photonics.

Following that paper, our group started work on realizing the full system proposed in that paper, where an entire DNN is optically processed in a single silicon photonic chip. The early development and validation of this architecture was led by Dr. Nicholas Harris and Dr. Darius Bunandar.

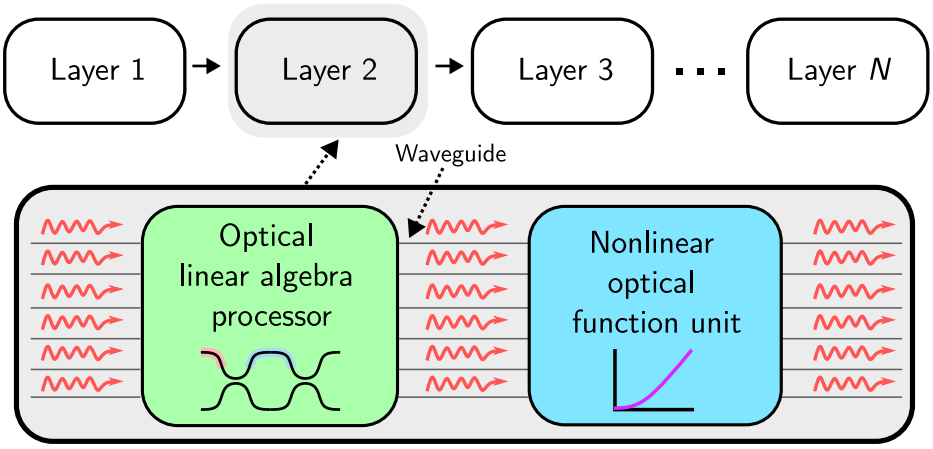

Figure 5-1: A coherent optical deep neural network processor processes the entire model in optics, including linear algebra and nonlinear functions. Each layer directly feeds the optical outputs into the next, enabling processing of an entire DNN with ultra-low latency.

converted back to the electrical domain using a coherent receiver array that photodetects the optical field at each channel.

This architecture has the following advantages over conventional DNN processors:

- In a photonic circuit, which can densely integrate optical components together into a small footprint, the latency can be on the order of nanoseconds, which is several orders of magnitude lower than existing digital processors.
- The energy consumption of photonic links is dominated by optical-to-electrical (O/E) conversion. Performing the entire DNN in optics, and performing O/E conversion only at the input and output of the system, minimizes expensive O/E/O conversions at each layer.
- For digital systems, the energy per operation (OP) scales as $O(N^2)$ for a neural network with N neurons. As we discuss later in this chapter, the energy per operation of this architecture scales as $O(N)$.
- Memory access, for instance to fetch weights and program them onto the hardware, dominates the energy consumption of modern DNN processors. Many electrical and optical systems for DNN processing repeatedly stream weights to/from memory onto the hardware, introducing excess energy consumption and latency. While this enables quick scaling to large model sizes, the power consumption and latency of these systems will ultimately be bottlenecked by expensive memory access[3].

[3]Google, for instance, has noted that their first generation tensor processing unit (TPU) was limited by memory access, not compute [13]. While later iterations increased the memory bandwidth, high-bandwidth memory (HBM) continues to be quite expensive, limiting the amount that can be realistically integrated on chip.

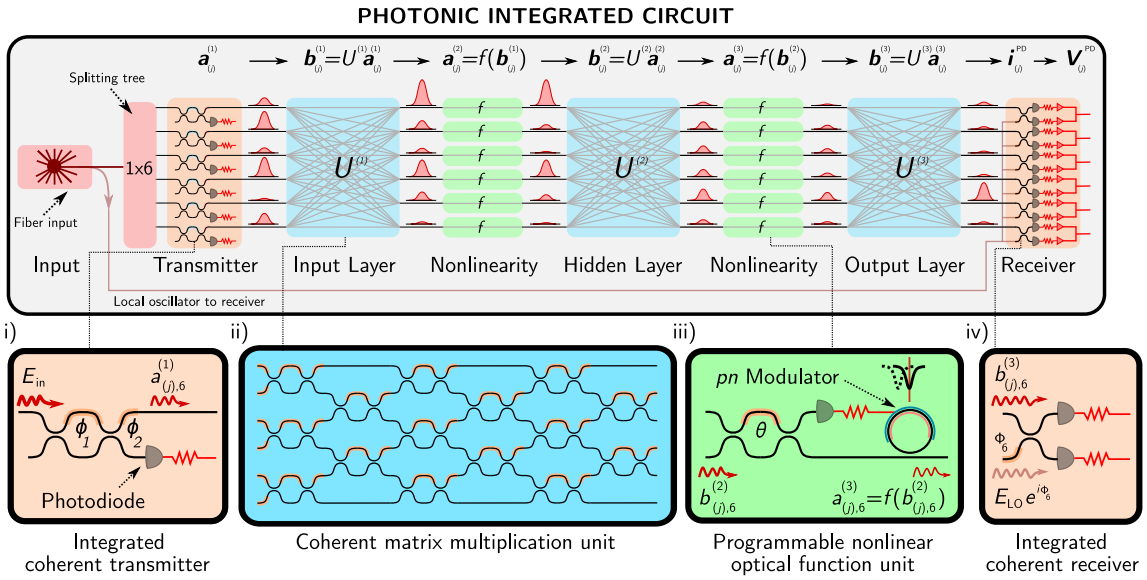

Figure 5-2: Architecture of the fully-integrated coherent optical neural network (FICONN). Inference is conducted entirely in the optical domain, without readout or amplification between layers. Light is fiber coupled into a single input on the chip and fanned out to the six channels of the transmitter **(i)**. Each channel encodes the amplitude and phase of one element of the input $\mathbf{x}_{(j)}$ into the optical field $\mathbf{a}_{(j)}^{(1)}$ with a Mach-Zehnder modulator and an external phase shifter. The coherent matrix multiplication unit **(ii)**, consisting of a Mach-Zehnder interferometer mesh, implements linear transformations. Programmable nonlinear optical function units **(iii)** realize activation functions $\mathbf{a}_{(j)}^{(n+1)} = f(\mathbf{b}_{(j)}^{(n)})$ by tapping off part of the signal to a photodiode, which drives a cavity off-resonance by injecting carriers into the waveguide. An integrated coherent receiver **(iv)** reads out the DNN output by homodyning the output field with a local oscillator.

> In this architecture, once the weights are programmed onto the hardware, they stay there. There is an upfront cost to fetch the model parameters from memory and program them into the phase shifters; once they are programmed, however, an arbitrary number of inferences can be performed.

5.3 A fully-integrated coherent optical neural network

We experimentally realized a coherent optical architecture for DNN processing in a custom application-specific photonic integrated circuit. This PIC was fabricated in a commercial silicon photonic foundry process incorporating low-loss fiber-to-chip couplers and waveguides, efficient phase shifters, and high-speed modulators and waveguide integrated photodiodes[4].

[4]I am immensely grateful to our collaborators at Elenion Technologies, with whom we worked to design and fabricate this circuit. I would particularly like to thank Dr. Michael Hochberg and Dr. Matthew Streshinsky, who gave invaluable input on the design of this system and for many useful discussions throughout the course of this project.

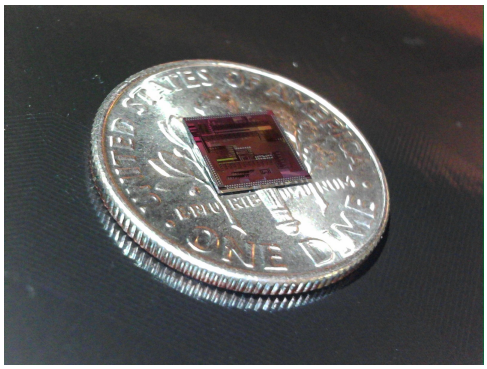

Figure 5-3: The fabricated photonic integrated circuit. This circuit, which consumes a footprint of 6×5.7 mm^2, implements an end-to-end photonic DNN processor and was fabricated in a commercial CMOS foundry.

The implemented architecture, which is monolithically integrated into a single chip, is shown in Figure 5-2. We optically process a deep neural network through the following stages:

1. The *transmitter* (TX) maps input vectors $\mathbf{x}_{(j)}$ to an optical field vector $\mathbf{a}_{(j)}^{(1)}$ by splitting an input laser field into MZI modulators, each of which encode one element of the vector into the amplitude and phase of the optical field.

2. The *coherent matrix multiplication unit* (CMXU), consisting of a programmable photonic mesh of Mach-Zehnder interferometers [29, 15, 149], implements the linear transformation $\mathbf{a}^{(1)} \rightarrow \mathbf{b}^{(1)} = U^{(1)}\mathbf{a}^{(1)}$ through passive optical interference.

3. The *programmable nonlinear optical function unit* (NOFU) implements the activation function to yield the input to the next layer, $\mathbf{a}^{(2)} = f(\mathbf{b}^{(1)})$. Following the input layer, the PIC directly inputs the optically-encoded signal into a hidden layer, composed of another CMXU and six NOFUs, that applies the optical transformation $\mathbf{a}^{(3)} = f(U^{(2)}\mathbf{a}^{(2)})$. The final layer $U^{(3)}$, realized using a third CMXU, maps $\mathbf{a}^{(3)}$ to the output $\mathbf{b}^{(3)}$. Inference therefore proceeds entirely in the optical domain without photodiode readout, amplification, or digitization between layers.

4. An *integrated coherent receiver* (ICR), shown in Figure 5-2(iv), reads out the amplitude and phase of the DNN output by homodyning each element of the output field $\mathbf{b}^{(3)}$ with a common local oscillator field E_{LO}. The DNN output is read out by transimpedance amplifiers that convert the photocurrent vector $\mathbf{i}^{PD}$ to a voltage vector $\mathbf{V}^{PD}$. $\mathbf{V}^{PD}$ is digitized and then normalized by the sum of voltages measured across all channels $\sum \mathbf{V}^{PD}$ to yield a quasi-probability distribution $\mathbf{V}^{\text{norm}}$ for a classification task. Each sample $\mathbf{x}^{(i)}$ is assigned the label corresponding to the highest probability, i.e. argmax($\mathbf{V}^{\text{norm}}$).

The fabricated PIC, shown in Figure 5-3, requires simultaneous control of 169 active devices, implements computation through 90 layers of optical devices, and comprises over 2,000 optical components.

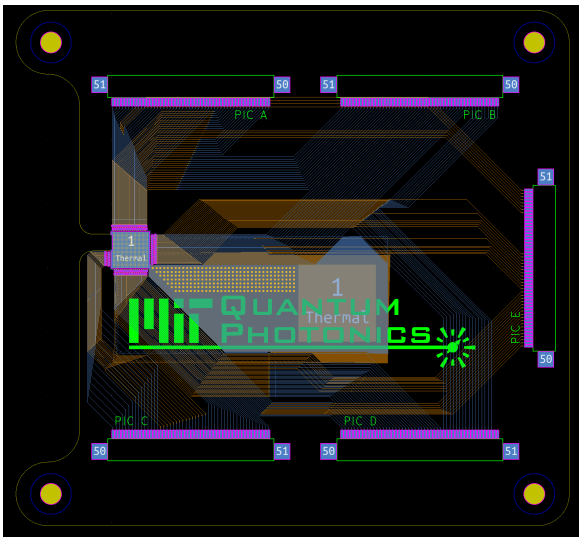

Figure 5-4: The printed circuit board interfacing on-chip electronics to the drivers.

5.4 System Packaging, Characterization, and Control

In this section, we discuss the custom evaluation board designed to demonstrate the FICONN, the constituent subsystems and their characterization, and the custom electronics designed for the system control.

5.4.1 Evaluation board

Testing the PIC required stable optical coupling and individual access to 169 electrical channels, which control the on-chip transmitter, receiver, and model parameters. Moreover, the package must be thermally stabilized, as data is processed coherently in interferometric circuits and temperature gradients can impact the system's performance.

We designed a custom printed circuit board (PCB) in a high-resolution (50 μm minimum feature size) manufacturing process to interface to the electrical devices on chip. The PCB, shown in Figure 5-4, is designed to map each pad on the PIC to a corresponding pad on the evaluation board with a 1:1 pitch. Each channel is then routed to a 50-channel, flexible ribbon cable (FFC) that interfaces to the control electronics. Including ground pad connections, the device requires five FFC ribbon cables to interface to the electronics. Electrical interconnections to the PIC are made through wirebonding.

Light is coupled into the chip from a tunable infrared laser using a single channel of a polarization-maintaining fiber array. In order to ensure stable optical coupling, the fiber array is permanently attached to the chip facet using index-matching epoxy. Over two years of use, we observed less than 1% variation in the optical coupling to the circuit, which was likely induced by temperature and humidity variations in the lab. No light is coupled out of the chip, as all readout is done on chip with the coherent receiver.

We measured an end-to-end loss for our system of 10 dB, including 2.5 dB fiber-to-chip coupling loss. As the depth of our system is 91 layers of optical components from input to readout, the end-to-end loss implies a per-component insertion loss of less than 0.1

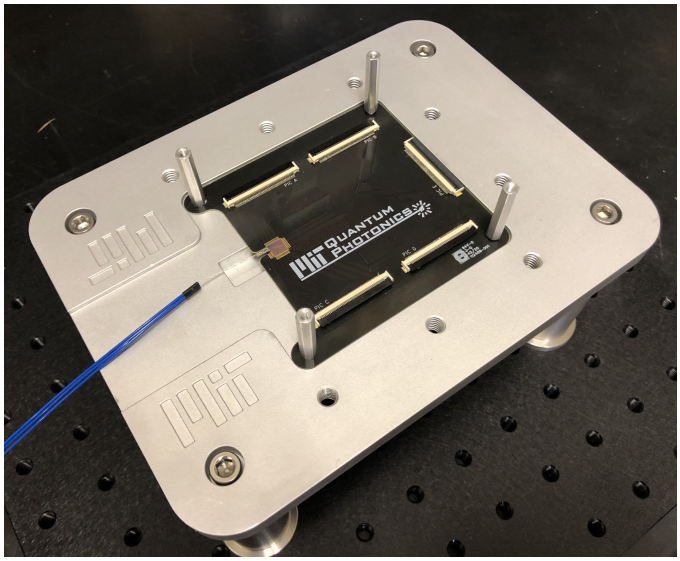

Figure 5-5: Fully-assembled evaluation board for the PIC with wirebonding, PCB, and fiber attach on mechanical chassis.

dB, enabling single-shot inference across all DNN layers without optical re-amplification.

Thermal stabilization was implemented using a custom mechanical chassis. The PIC is epoxied to a metal pad on the PCB, which is thermally shorted to a corresponding pad on the opposite side through copper vias. On the opposite end of the chassis we attached a copper thermal block that makes contact with the thermal pad on the PCB and functions as a large heatsink. A Peltier unit, connected to a feedback controller (Arroyo Instruments) and a second heatsink, is used to actively stabilize the PIC temperature to within 0.004° C.

The evaluation board with wirebonded PIC is shown in Figure 5-5.

5.4.2 Control electronics

Most of the devices on chip were electrically controlled through a 192-channel software programmable current source (Qontrol Systems Q8iv). Each channel sources up to 24 mA of current with 16 bits of precision, corresponding to approximately 0.4 mrad precision in our system.

Unfortunately, as this system is controlled through a serial connection to a digital computer, its update rate was quite slow (tens of Hertz). This is fine for most of the phase shifters on chip, which encode weights and do not change frequently. However, faster electronics are required for the transmitter and receiver on chip, which input data into the system and read out the classification result.

For faster transmission of input vectors into the DNN, we designed a custom 16-bit current driver system that used a microcontroller to buffer the training set in memory, which enabled training at the maximum DAC speed rather than the speed of the serial connection to the computer[5].

[5]As we use thermal phase shifters at the transmitter, the drivers are designed to be current sources.

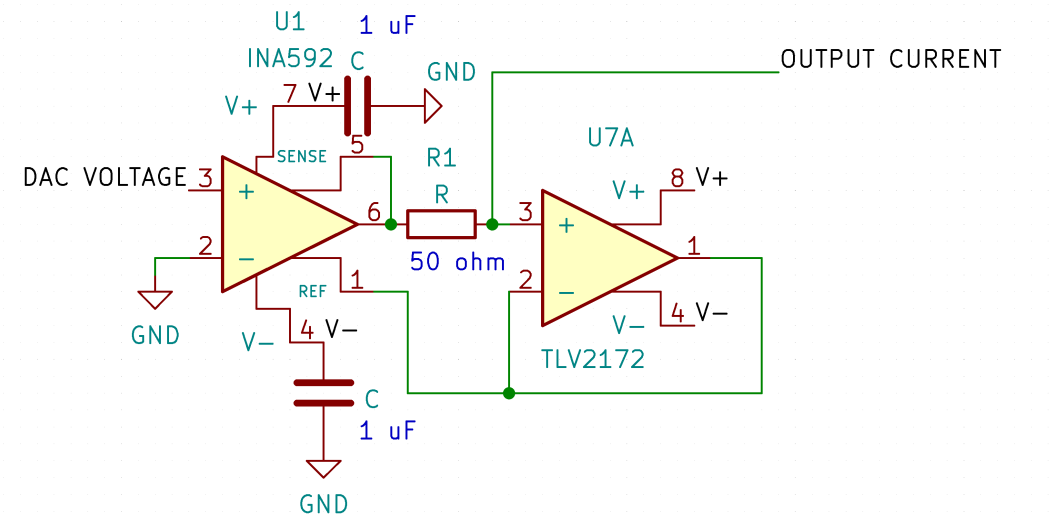

Figure 5-6: Schematic of a single channel of the transmitter board, which implements the Howland current pump architecture.

The schematic of a single channel of the current driver is shown in Figure 5-6. We use a Howland current pump architecture, where the input voltage is set by a high-precision (16-bit) digital-to-analog converter (DAC). Analog feedback in the circuit ensures a constant output current regardless of the output load. Mismatched resistances in the circuit can introduce some dependence of the output current on the load resistance; to minimize this, we use an amplifier integrated with on-chip resistors to maximize the common-mode rejection ratio.

Output signals from the coherent receiver are read out using a custom receiver board. This board amplifies output photocurrents with a transimpedance amplifier. The output signal is then digitized using a high-precision, 18-bit analog-to-digital converter (ADC).

The transmitter and receiver boards are jointly interfaced to through a single microcontroller (Teensy 4.0). Both the DAC and ADC modules communicate to the microcontroller through a serial peripheral interface (SPI), which enables interfacing to all the ADC and DAC units with a minimal number of data buses.

For the *in situ* training experiments we discuss later in the chapter, the training set is locally buffered in the microcontroller memory. Each epoch during training iterates through the training set on chip and locally stores the output signals from the coherent receiver. All of the data is then batched together and communicated to the computer running the experiment; this enables training to proceed at the maximum speed of the electronics, rather than the relatively slow serial connection between the computer and microcontroller.

In Figure 5-7, we show the test setup in the lab, with evaluation board, custom transmitter and receiver board, and the 192-channel current source used to set model parameters on chip.

Voltage sources produce significant electrical crosstalk in these systems due to parasitic resistances to the ground plane, as has been discussed extensively in ref. [51].

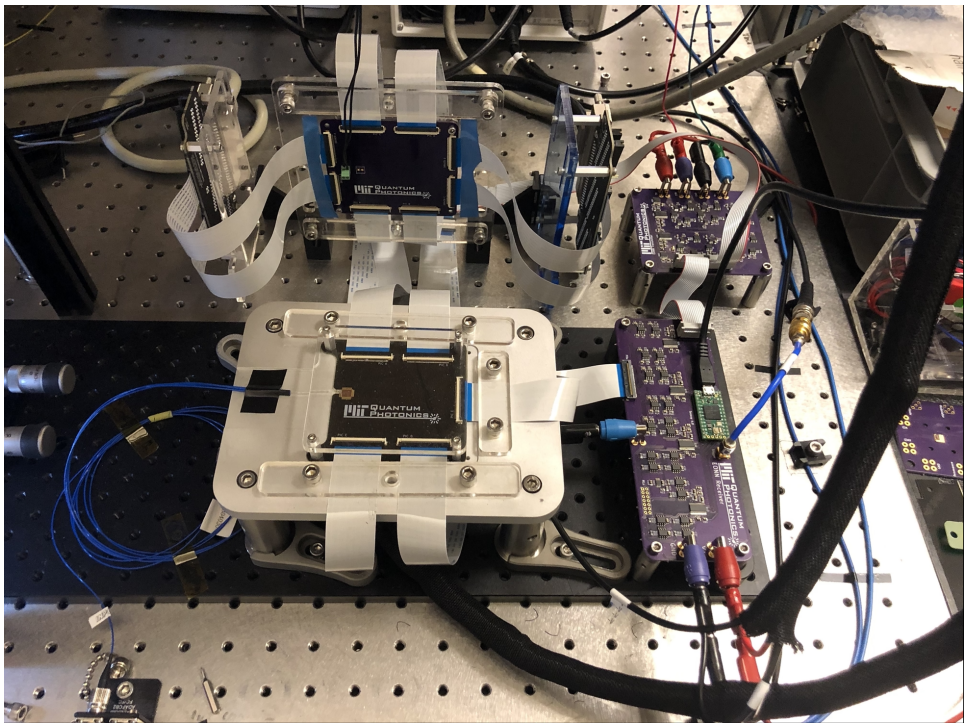

Figure 5-7: Test setup in the lab, including evaluation board, custom transmitter and receiver boards, and driver electronics.

5.4.3 Transmitter

The light coupled into the chip is first split with an MZI into a local oscillator (LO) path, which is directed to the coherent receiver, and a signal path, which is fanned out to six channels through an MMI splitting tree. Each channel of the transmitter comprises an MZI, which programs the amplitude of one element of $\mathbf{a}_{(j)}^{(1)}$, and a phase shifter on the output that encodes the phase, enabling inference on complex-valued input signals. The drop port of each channel includes an on-chip photodiode, simplifying characterization of the transmitter bank.

Characterization

Recall that a Mach-Zehnder interferometer (MZI) performs the programmable 2×2 unitary operation:

$$U(\theta_1, \theta_2) = ie^{i\theta_1/2} \begin{bmatrix} e^{i\theta_2}\sin(\theta_1/2) & e^{i\theta_2}\cos(\theta_1/2) \\ \cos(\theta_1/2) & -\sin(\theta_1/2) \end{bmatrix} \tag{5.1}$$

To characterize a single MZI, we first input light into one port of the device and measure the output transmission $T(\theta_1) = P_{\text{out}}/P_{\text{in}}$. For an ideal device, the output transmission at the bar port is

$$T_{\text{bar}}(\theta_1) = \sin^2\left(\frac{\theta_1}{2}\right) \tag{5.2}$$

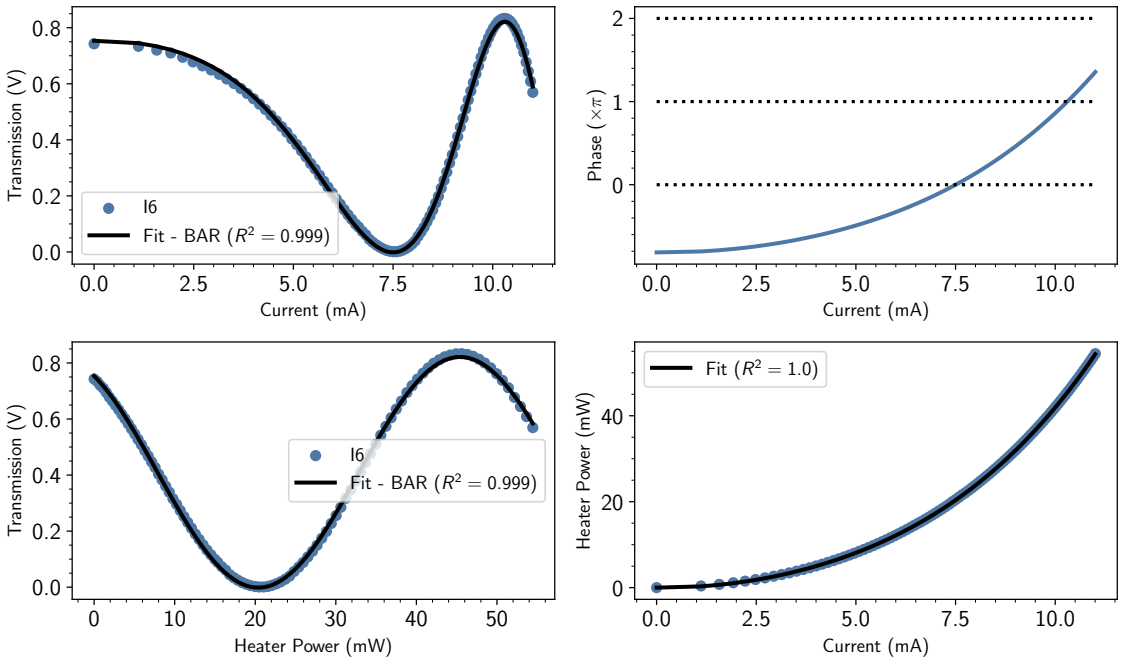

Figure 5-8: Typical fitting procedure for an MZI on the PIC.

and at the cross port:

$$T_{\text{cross}}(\theta_1) = \cos^2\left(\frac{\theta_1}{2}\right) \tag{5.3}$$

In a fabricated MZI, θ_1 is determined by the total dissipated power $I \times V(I)$, where I is the programmed current and $V(I)$ is the voltage dropped across the device. To characterize a device in the transmitter, where the cross port of each channel has a photodiode, we sweep I and measure the voltage $V(I)$ and output transmission $T(I)$. We fit the expressions:

$$V(I) = a_4 I^4 + a_3 I^3 + a_2 I^2 + a_1 I \tag{5.4}$$

$$T_{\text{cross}} = A + B\cos\left(\frac{IV(I)}{P_\pi}\pi + p_0\right) \tag{5.5}$$

where $a_4, a_3, a_2, a_1, A, B, P_\pi, p_0$ are fitting parameters. Here, P_π is the total dissipated power required to induce a π phase shift, p_0 is the static phase difference between the two interferometer arms, and $1/(A-B)$ is the interferometer extinction ratio. We found that a fourth-order polynomial was required to fit the voltage-current relationship of the heaters, which became non-Ohmic at high currents due to self-heating and velocity saturation of the carriers. If we measured $T(I)$ at the bar port, the latter expression would instead be:

$$T_{\text{bar}} = A - B\cos\left(\frac{IV(I)}{P_\pi}\pi + p_0\right) \tag{5.6}$$

This yields a mapping between the current I and the programmed phase $\theta_1(I)$ of the form:

$$\theta_1(I) = p_4 I^4 + p_3 I^3 + p_2 I^2 + p_1 I + p_0 \tag{5.7}$$

In Figure 5-8, we show the typical fitting for a single MZI. As the results show, a typical channel realizes more than 40 dB of extinction, enabling programming of input vectors with more than 13 bits of precision.

Thermal crosstalk correction

As we use thermal phase shifters for programming the transmitter, thermal crosstalk between devices will also impact the performance. To correct for this, we directly measured the 12×6 crosstalk matrix M, where M_{ij} denotes the crosstalk on channel i produced by an aggressor channel j. This quantity was measured by driving channel i and measuring the output transmission T at different current settings for channel j. For each measurement, we fit equation 5.5 to the data to extract the static phase p_0. Thermal crosstalk will cause p_0 to vary as a function of the settings in channel j due to parasitic heating; we fit a linear expression to this data to extract the crosstalk coefficient M_{ij}.

Figure 5-9a shows an example of this procedure, where we extracted the crosstalk on channel 1 produced by channel 2. Having obtained M, we can now obtain the phase settings $\mathbf{\Phi}$ for a desired programming $\mathbf{\Phi}'$ by computing:

$$\mathbf{\Phi} = M^{-1}(\mathbf{\Phi}' - \mathbf{\Phi}_0) + \mathbf{\Phi}_0 \tag{5.8}$$

where $\mathbf{\Phi}_0$ is the static phase for each channel. We neglected crosstalk on the external phase shifters of the transmitter, which program the phase of the input $\mathbf{a}^{(1)}$, as we did not have coherent detection directly at the transmitter output.

In order to benchmark the correction protocol for each transmitter channel we repeatedly attempted to program $\theta_1 = \pi/2$ while setting all other channels to a random phase setting. Shown in Figure 5-9b is the phase setting actually implemented by the transmitter channel for 500 such experiments, which we extracted by measuring the transmission T and computing $2 \arccos \sqrt{T}$. Thermal crosstalk correction greatly improves both the accuracy and repeatability of each channel; for example, the measured phase on channel 2 improves from 0.493 ± 0.015 to 0.501 ± 0.003 following correction.

5.4.4 Coherent matrix multiplication unit

Linear transformations on chip are computed with the coherent matrix multiplication unit (CMXU), which is comprised of a programmable photonic mesh [29] of 15 MZIs connected in the Clements configuration [45]. This device implements an arbitrary 6×6 unitary operation $U^{(1)}$ on the optical fields $\mathbf{a}^{(1)}$. Unitary weighting, which redistributes light between optical modes but does not attenuate it, minimizes optical losses and enables single-shot DNN inference without re-amplification or readout between layers. Training unitary layers has also been shown to avoid the vanishing gradient problem, improving optimization of deep and recurrent neural networks [150].

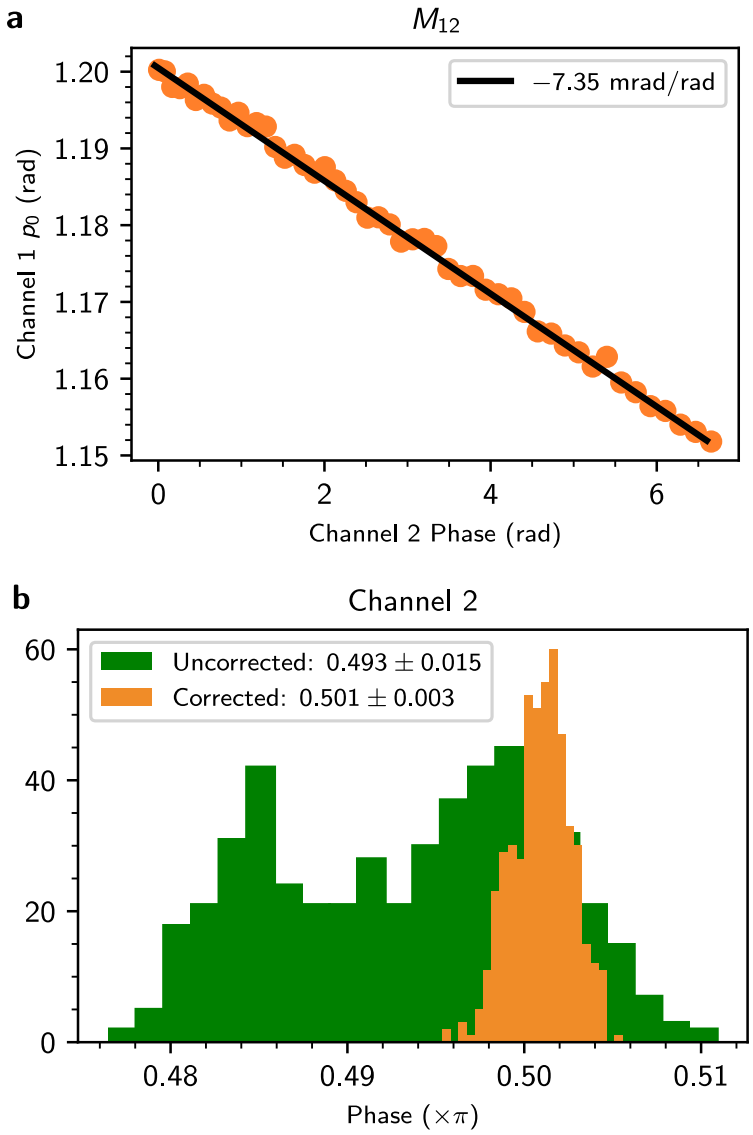

Figure 5-9: a) To determine the elements of the thermal crosstalk matrix M, we drive an aggressor channel j while characterizing the static phase p_0 of channel i. As an example, here we characterize M_{12} by plotting the static phase of channel 1 as a function of the phase setting of channel 2. We fit a linear function to this data to find a crosstalk coefficient of $M_{12} = -0.00735$. b) We benchmark the effectiveness of thermal crosstalk correction by repeatedly trying to program a channel to $\theta_1 = \pi/2$, while setting all other channels to random values. We then determine the actual phase implemented by measuring the output transmission T and computing $2\arccos\sqrt{T}$. As an example, here we show the results for channel 2, where over 500 random experiments thermal crosstalk correction greatly improves the repeatability of programming a channel to a desired phase.

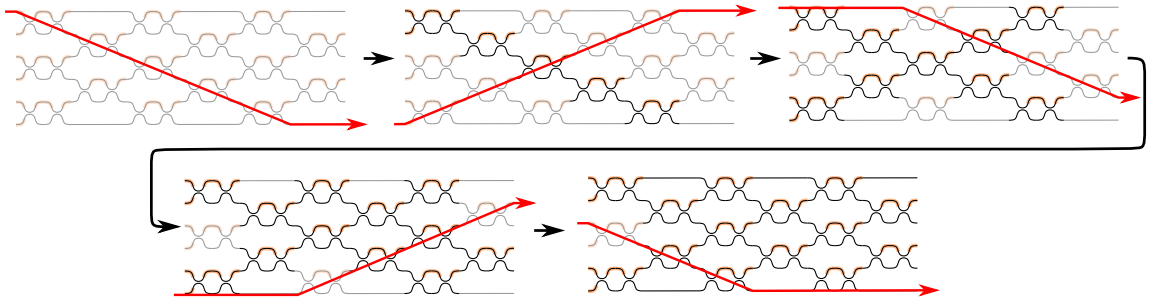

Figure 5-10: Calibration procedure for internal phase shifters in the CMXU. The devices along the main diagonal and antidiagonal are calibrated first. Once these devices are characterized, the remainder of the phase shifters can be calibrated by programming devices along the main diagonal.

Characterization of internal phase shifters

The procedure for characterizing the CMXU is sketched in Figure 5-10[6]. We use the photodiodes at the output of each matrix processor; for the first two layers, we use the photodiode at each nonlinear optical function unit (NOFU), while the last layer is calibrated with the system's receiver.

In an uncalibrated mesh, light will scatter randomly through the circuit as p_0 is random for each device. To characterize the circuit, we first input light into the top input (input 1) of the mesh and measure the transmission at the bottom output (output 6). We then optimize the internal phase shifters along the main diagonal in a round robin fashion to maximize the signal at output 6. This procedure deterministically initializes the main diagonal to the cross state ($\theta_1 = 0$), as there is only one possible path between input 1 and output 6. Having initialized the diagonal, we can then calibrate each device along it by sweeping the phase shifter θ_1, measuring T at output 6, and fitting equation 5.5 to the data. The antidiagonal, connecting input 6 to output 1, is calibrated in the same way.

Having characterized the main diagonals, the remainder of the devices can be calibrated in a similar fashion. For instance, inputting light into mode 1 and setting the top left MZI in the circuit, which is already calibrated, to the bar state provides access to the first subdiagonal. The uncalibrated devices can then be characterized with the same procedure as was used for the main diagonal. We show the full calibration sequence in Figure 5-10.

Characterization of external phase shifters

The protocol above calibrates all internal phase shifters θ_1 in a matrix processor. The external phase shifters θ_2 are calibrated using "meta-MZIs," as shown in Figure 5-11. A "meta-MZI" consists of two MZIs in columns $i-1, i+1$ that are programmed to implement a 50-50 beamsplitter ($\theta_1 = \pi/2$). This subcircuit now functions as an effective MZI, where the relative phase difference between two external phase shifters $\theta_{2,a}, \theta_{2,b}$ is equivalent to the setting of the internal phase shifter in a discrete device.

[6] I'd like to thank Dr. Ryan Hamerly, who I collaborated with on photonic mesh error correction techniques, for many useful discussions on calibrating these systems.

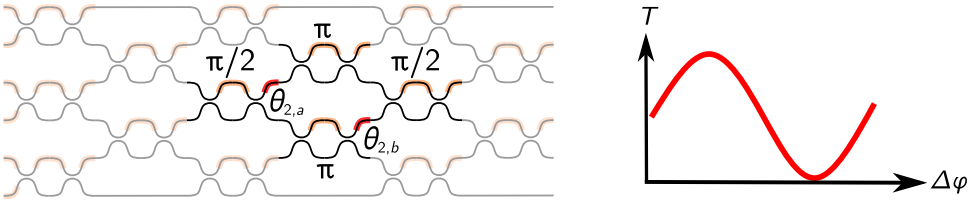

Figure 5-11: "Meta-MZI" for calibrating external phase shifters. Two phase shifters in columns $i-1, i+1$ are set to implement a 50-50 beamsplitter. The output transmission of this meta-interferometer, which functions exactly like a discrete MZI, is dependent on the phase difference between the external phase shifters $\Delta\phi = \theta_{2,b} - \theta_{2,a}$.

We fix one of the two external phase shifters to $I = 0$, sweep the current programmed into the other, and measure the output transmission T. Fitting the data to equations 5.5, 5.6, depending on the port T is measured out of, calibrates the static phase difference $\Delta\phi(I = 0) = \theta_{2,b}(I = 0) - \theta_{2,a}(I = 0)$. Repeating this procedure for all devices produces a linear system of equations that can be inverted to find the static phase p_0 for each external heater. More details on this procedure can be found in [51].

Thermal crosstalk correction

Correcting for thermal crosstalk in the CMXU is more challenging. As the CMXU is a mesh of interferometers, changing the programming of aggressor channels can introduce phases and redirect light through the circuit in unexpected ways. These effects are challenging to disentangle from pure thermal crosstalk when the circuit also has other component errors, such as beamsplitter imperfections and device loss, making it difficult to directly measure M.

To address this, we instead developed a digital model of the hardware, which modeled in software the response of a device with known beamsplitter errors, waveguide losses, and thermal crosstalk. As the effects of all of these imperfections are known *a priori* for Mach-Zehnder interferometer meshes [58, 54], we can fit a software model, where these imperfections are initially unknown model parameters, to data taken on the real device. If the software model can accurately reproduce measurements from the hardware, the parameters found to describe the device imperfections can be used to deterministically correct errors on the real hardware.

As a note, our approach is not a "black-box" or neural network model of the device. Our model is based on the physics of how Mach-Zehnder interferometer meshes behave, and thus the parameters we find are realistic and correspond to true physical attributes of the device, such as the error for a particular directional coupler. Similar to the earlier work on hardware error correction, our approach here efficiently corrects for component errors, as no real-time optimization is done on the hardware. However, fitting the device response to a software model eliminates the need to calibrate component errors one at a time.

We fit the model to a dataset obtained by programming 300 random unitary matrices into the chip and measuring the response to 100 randomly selected input vectors. Our

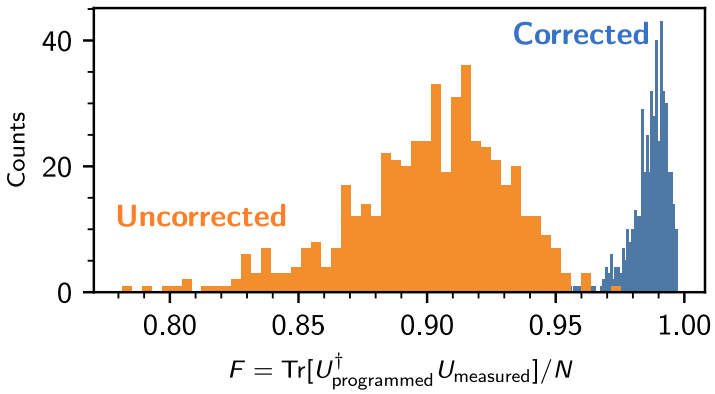

Figure 5-12: Measured fidelity of 500 arbitrary unitary matrices implemented on a single layer using a "direct" approach (orange) and an approach that takes into account hardware errors and thermal crosstalk (blue).

software model, which is written in `JAX` for auto-differentiability, is fit to the measured data using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm. We found that our software model was able to predict hardware outputs with an average fidelity $F = \mathrm{Tr}[U^{\dagger}_{\mathrm{measured}} U_{\mathrm{software}}]/N$ of 0.969 ± 0.023.

Benchmarking

We benchmarked the matrix accuracy of the CMXU by programming 500 random 6×6 unitary matrices sampled from the Haar measure into the device and measuring the fidelity $F = \mathrm{Tr}[U^{\dagger}_{\mathrm{programmed}} U_{\mathrm{measured}}]/6$. To measure the fidelity on chip, we sequentially transmit the columns of $U^{\dagger}_{\mathrm{programmed}}$ to compute the metric $F = \mathrm{Tr}[U^{\dagger}_{\mathrm{programmed}} U_{\mathrm{hardware}}]/N$. As the inverse of a unitary matrix is its adjoint, for a perfect hardware implementation of $U_{\mathrm{programmed}}$ the quantity F should equal 1.

In the histogram in Figure 5-12, we show the measured fidelity obtained with a "direct" programming, where we algorithmically decompose the phase shifter settings as outlined in [45], and using a modified programming that corrects for hardware errors, losses, and thermal crosstalk [58, 54, 55]. While a direct programming only achieves a matrix fidelity of $\langle F \rangle = 0.900 \pm 0.031$, correcting for hardware non-idealities improves this value to $\langle F \rangle = 0.987 \pm 0.007$ for the CMXU.

5.5 Nonlinear optical function unit

It is the combination of linear and nonlinear transformations that make DNNs universal function approximators[7]. Therefore, a key requirement for a coherent optical DNN processor is realizing fast, energy-efficient nonlinearities that can be integrated into photonic circuits.

[7]Several linear layers cascaded in series is equivalent to one linear transformation!

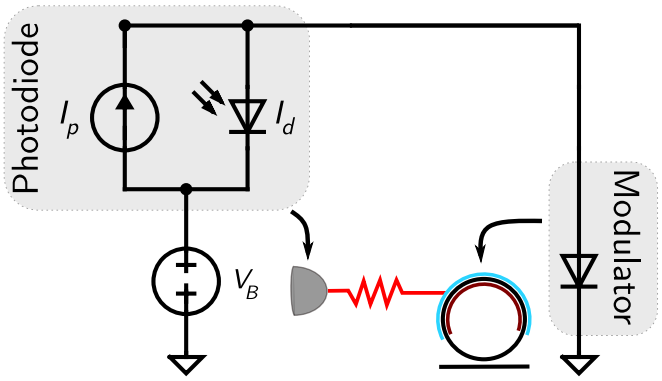

Figure 5-13: Circuit diagram of resonant EO nonlinearity. The photocurrent I_p directly drives a *pn*-doped resonant modulator. No amplifier stage is required between the two and the devices are directly connected on chip. By adjusting the bias voltage V_B, the nonlinearity can be operated in forward or reverse bias.

This is an ongoing problem in the photonic computing community. Optical material nonlinearities, such as the second-order susceptibility, are notoriously weak and require watts of optical power to activate. The early demonstration by Shen and Harris [15] proposed realizing an optical nonlinearity using saturable absorbtion. There are two obstacles to using such a nonlinearity in realistic system, however: (1) they are difficult to monolithically integrate; and (2) response is linear at low optical powers and only exhibits nonlinearity at high incident powers. Since optical power is progressively lost through the circuit, it will become increasingly challenging to trigger the nonlinearity for very deep networks. Preferably, we should have a nonlinearity that triggers at low optical powers[8].

A more viable approach is to make use of an intermediate electrical conversion, as electrical devices can realize extremely strong nonlinearities. Here, the idea is to photodetect part or all of the optical signal, convert the photocurrent to a voltage through a transimpedance conversion, and use the generated voltage to drive an optical modulator that re-encodes onto a new signal. This strategy, often referred to as an optical-electrical-optical (or OEO) nonlinearity, has been explored previously by other groups [151, 78, 142]. However, these devices have been relatively inefficient, either requiring absorbing all of the optical signal and converting it to the electrical domain, or integrating with the system an off-chip, high-gain electronic amplifier to boost the modulation voltage.

Neither of these strategies emulate a true optical nonlinearity, which would entail the optical signal *coherently* modulating itself. Although not strictly necessary, such a device would also be programmable to realize different types of nonlinear functions[9].

In order to implement such a function, we developed the resonant electro-optical nonlinearity shown schematically in Figure 5-2iii)[10]. This device directs a fraction β of the incident optical power $|b|^2$ into a photodiode by programming the phase shift θ in an MZI.

[8] Such a nonlinearity could approximate a rectified linear unit, or ReLU function, which is an extremely popular activation function for today's DNNs.

[9] As we discovered later, this also enabled the exciting possibility of training the nonlinearity.

[10] Many of the characterization experiments presented in this section on the nonlinear unit were performed in collaboration with Dr. Alexander Sludds.

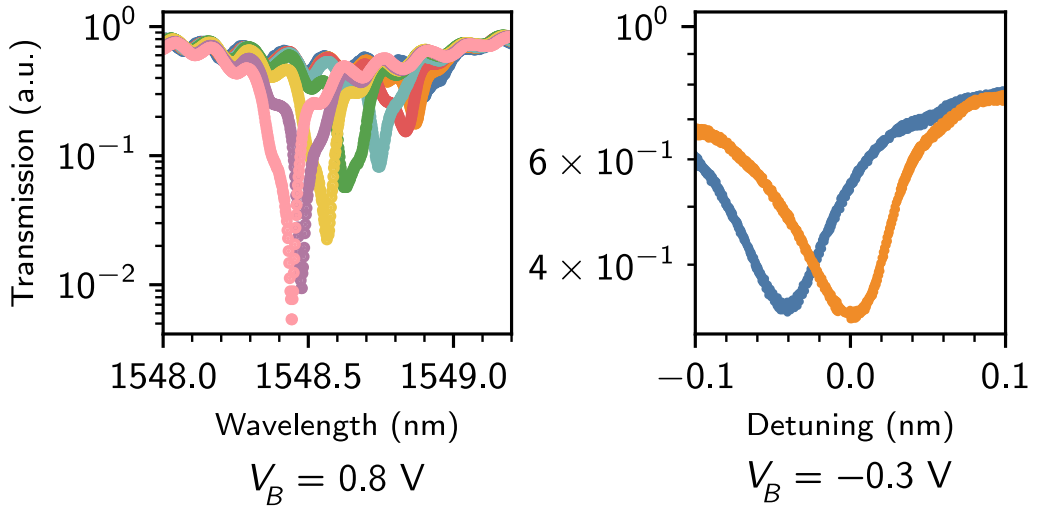

Figure 5-14: Left: Detuning of the cavity resonance at various incident optical powers when operated in carrier injection mode ($V_B > 0$). Right: Cavity detuning in carrier depletion mode ($V_B < 0$). Our system realizes close to a linewidth detuning without the use of any amplifier, improving energy consumption and latency of the nonlinearity. A full linewidth detuning can be realized by further engineering the cavity finesse.

The photodiode is electrically connected to a *pn*-doped resonant microring modulator, and the resultant photocurrent (or photovoltage) detunes the resonance by either injecting (or depleting) carriers from the waveguide. The remainder of the incident signal field passes into the microring resonator; the nonlinear modulation of the electric field b by the cavity, which is dependent on the incident optical power $|b|^2$, results in a coherent nonlinear optical function for DNNs. Setting the detuning of the cavity and the fraction of optical power tapped off to the photodiode determines the implemented function.

The electrical circuit for the NOFU is shown in Figure 5-13. Incident light generates a reverse current in the photodiode; depending on the bias voltage V_B, this either injects carriers into the modulator or generates a photovoltage that depletes the modulator of carriers. Figure 5-14 shows the device response in injection (left) and depletion modes (right). In injection mode, optical power modulates both the loss and phase of the resonator, producing a strong nonlinear response to the incident field b. In depletion mode, we observe nearly a linewidth detuning when the incident light is switched on vs. off, which is induced by the voltage produced by the photodiode.

The phase response and round-trip attenuation a as a function of the photocurrent I are shown in Figure 5-15. Assuming a photodiode responsivity of ~ 1 A/W, we find that about 75 μW is sufficient to detune the NOFU by a linewidth. As we bias the device to 0.8 V in our experiment, the power consumption during operation is therefore ~ 60 μW.

Compared to prior approaches [78, 142], the NOFU directly drives the modulator through the photodiode and eliminates the amplifier stage between them. This greatly improves the latency and energy efficiency of the device, as high speed transimpedance amplifiers can consume up to hundreds of milliwatts of power [152]. For our device, incorporating such an amplifier would have increased the power consumption by about

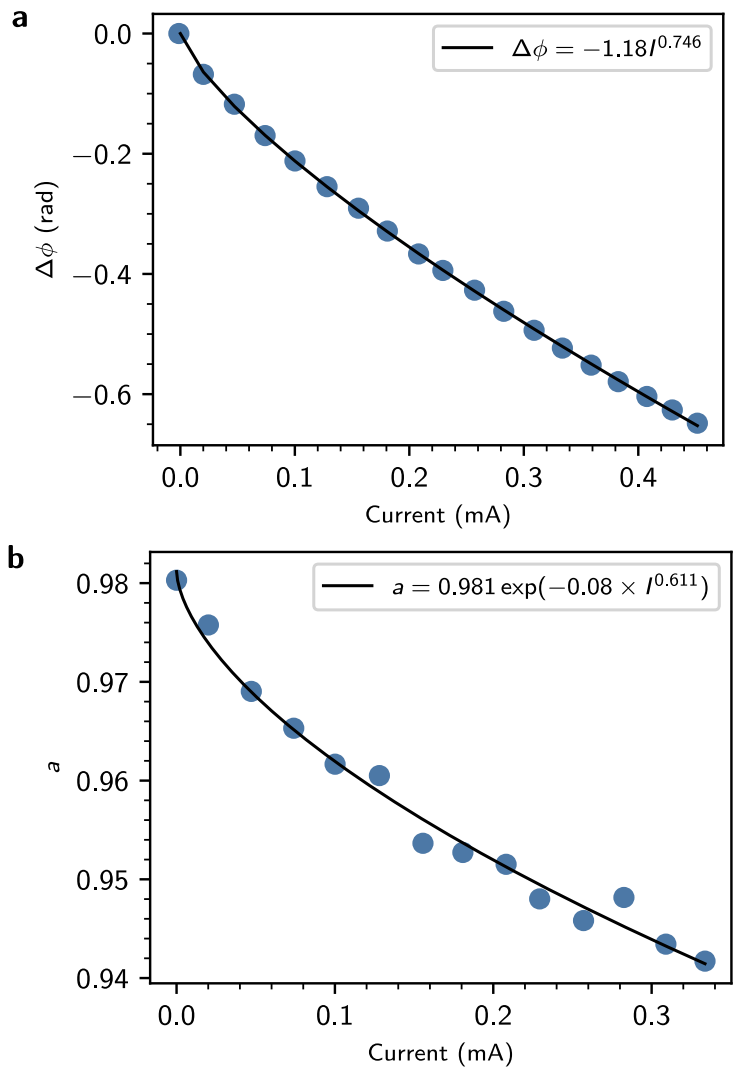

Figure 5-15: a) Phase shift $\Delta\phi$ in cavity vs. incident photocurrent. b) Round-trip amplitude loss a as a function of incident photocurrent. As photocurrent increases more carriers are injected into the waveguide, increasing the loss of the optical signal inside the resonator.

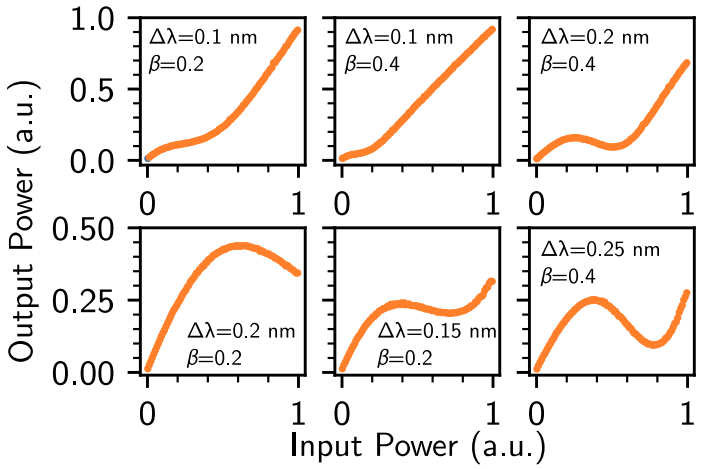

Figure 5-16: Activation functions measured on chip. Programmable function shapes can be realized by adjusting the cavity detuning $\Delta\lambda$ and fraction of light β tapped off to the photodiode.

two orders of magnitude. Our design, which eliminates intermediate amplifier circuitry and is therefore "receiverless" [3], is not only more energy-efficient, but also eliminates the latency introduced by the amplifier.

In Figure 5-16, we show several of the activation functions measured on chip. The programmability of the device enables a wide range of nonlinear optical functions to be realized. By tuning the fraction of power tapped off to the photodiode and the relative detuning of the cavity, we can not only program the form of the nonlinear function, but also train it during model optimization.

5.6 Optically accelerating training

A central challenge in machine learning is the efficient training of model parameters. In particular, a critical bottleneck for model training is forward inference, as it requires many evaluations of the model on a large training set to optimize weight parameters. *In situ* training on photonic hardware can take advantage of near-instantaneous DNN inference, lowering the latency and power consumption of model training. Moreover, learning weights in real time can benefit applications that natively process optical data, such as LiDAR systems [144], optical transceivers [148], and federated learning for edge devices [153, 154].

Previous work on *in situ* training has focused on developing optical implementations of "backpropagation," which is the standard for training electronic DNNs [155, 156]. However, these approaches train only the linear layers of a photonic system and require evaluating gradients of activation functions on a digital system, thereby limiting the optical acceleration obtained by computing a multi-layer DNN in a single shot. Alternatively, genetic algorithms have been used to optimize weights on chip [157], but they are challenging to scale to large model sizes and require many generations to converge.

We trained the model parameters of the FICONN *in situ*, including those of the activation functions, by evaluating the derivatives of those parameters directly on the

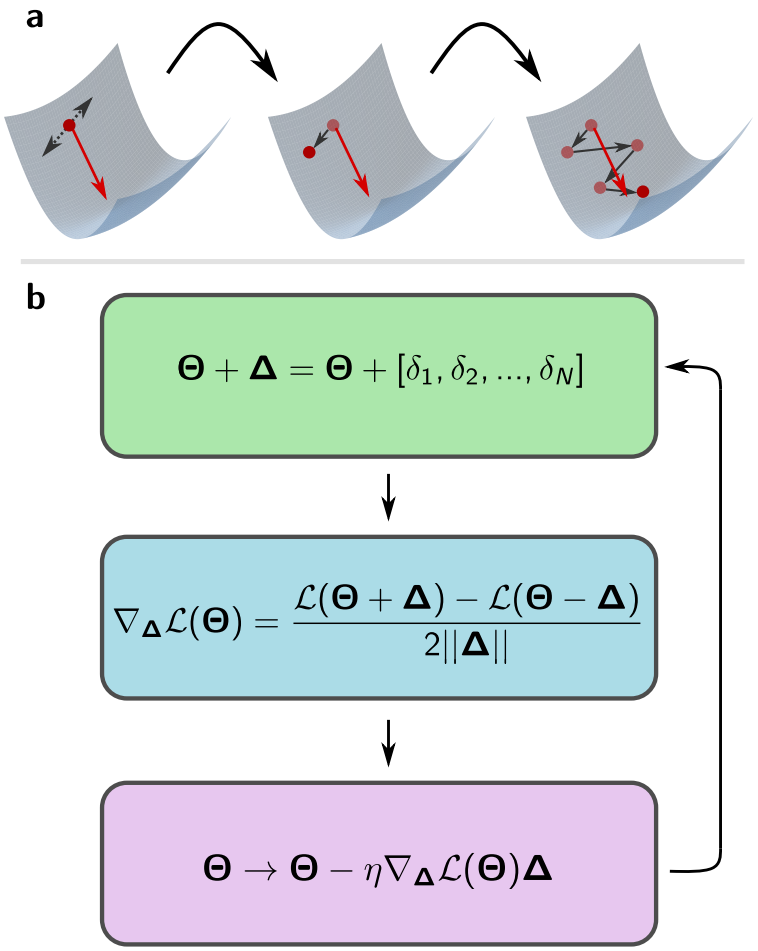

Figure 5-17: a) A multivariate cost function $\mathcal{L}(\boldsymbol{\Theta})$ can be minimized by computing the directional derivative of the function along a random direction (black). This directs the optimization along the component of the gradient (red) parallel to the search direction. Over multiple iterations, the steps taken along random directions average to follow the direction of steepest descent to the minimum. b) *In situ* training procedure. At every iteration, the directional derivative of the cost function $\mathcal{L}(\boldsymbol{\Theta})$ is computed in hardware along a randomly chosen direction $\boldsymbol{\Delta}$ in the search space. $\boldsymbol{\Delta}$ is chosen from a Bernoulli distribution to be $\pm\delta$. The weights $\boldsymbol{\Theta}$ are then updated by the measured derivative following a learning rate η chosen as a hyperparameter of the optimization.

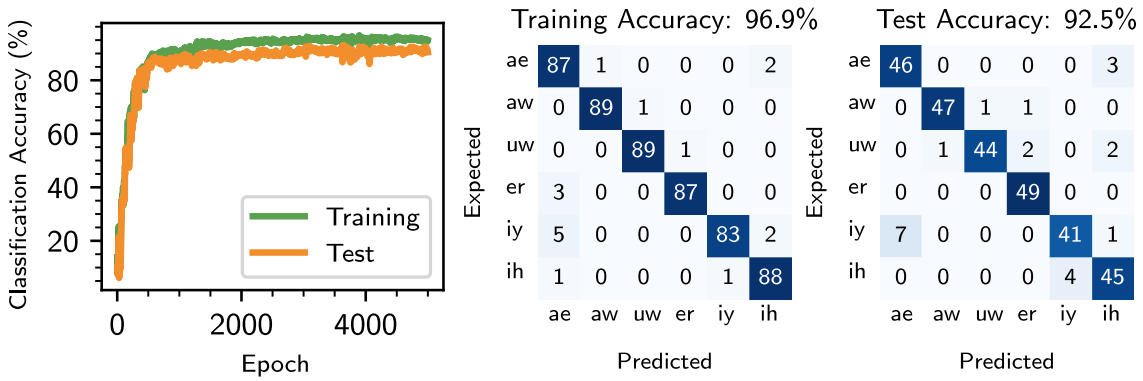

Figure 5-18: *In situ* training of a photonic DNN for vowel classification. We obtain 92.5% accuracy on a test set, which is comparable to the performance (92.5%) obtained on a digital model with the same number of weights. Despite not having direct access to gradients, our approach produces a training curve similar to those produced by standard gradient descent algorithms.

hardware[11]. Our approach, which is based on prior work on *in situ* optimization of analog VLSI neural networks [158, 159], is robust to noise, performs gradient descent on average, and is guaranteed to converge to a local minimum. Moreover, it is not limited to our specific system, but can be generalized to any hardware architecture for photonic DNNs.

A direct approach to computing the gradient on hardware would be to perturb the model parameters $\mathbf{\Theta} = [\Theta_1, \Theta_2, ..., \Theta_N]$ one weight at a time and repeatedly batch the training set through the system [15]. This procedure produces a forward difference estimate of the loss gradient $\nabla\mathcal{L}(\mathbf{\Theta})$ with respect to all weights. Moreover, since the derivatives are evaluated directly on chip, this procedure extends to other hardware parameters, such as the detuning and fraction of power tapped off in the NOFU. The drawback to this approach is that for N parameters, it requires batching the training set through the hardware $2N$ times.

Our approach varies all model parameters $\mathbf{\Theta}$ simultaneously. Figure 5-17 sketches the optimization procedure. Instead of perturbing the parameters one weight at a time, during training the system perturbs all parameters towards a random direction $\mathbf{\Delta}$ in search space, i.e. $\mathbf{\Theta} \rightarrow \mathbf{\Theta} + \mathbf{\Delta} = \mathbf{\Theta} + [\delta_1, \delta_2, ..., \delta_N]$. At each iteration the system then computes the directional derivative:

$$\nabla_{\mathbf{\Delta}}\mathcal{L}(\mathbf{\Theta}) = \frac{\mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}) - \mathcal{L}(\mathbf{\Theta} - \mathbf{\Delta})}{2||\mathbf{\Delta}||} \tag{5.9}$$

As in standard gradient descent, the weights $\mathbf{\Theta}$ are then updated to $\mathbf{\Theta} \rightarrow \mathbf{\Theta} - \eta\nabla_{\mathbf{\Delta}}\mathcal{L}(\mathbf{\Theta})\mathbf{\Delta}$, where η is a learning rate chosen as a hyperparameter of the system.

Compared to the forward difference approach outlined earlier, our approach requires batching the training set through the hardware only twice per iteration. Moreover, we obtain true estimates of the cost function $\mathcal{L}$ and the derivative $\nabla_{\mathbf{\Delta}}\mathcal{L}(\mathbf{W})$, ensuring that component errors or errors in calibration do not affect the accuracy of training. Unlike

[11] I would like to thank Prof. Stefan Krastanov for many useful discussions on the training experiments.

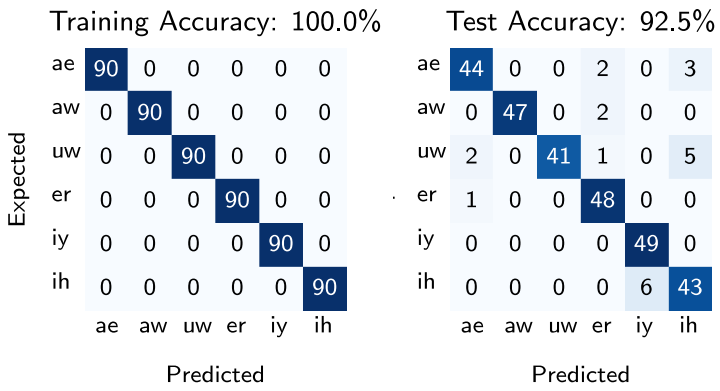

Figure 5-19: Performance of the digital model on the vowel classification task. The model overfits the training set, achieving 100% accuracy, but performance on the test set is comparable to the accuracy achieved by our system (92.5% on the digital model vs. 92.5% on the FICONN).

other derivative-free optimization methods, our approach will always track the direction of steepest descent, as errors in the gradient direction average out to zero over multiple epochs [158, 159].

We implemented *in situ* training of $\boldsymbol{\Theta}$, which includes weights and nonlinear function parameters, for a standard vowel classification task (dataset available at [160]). At each epoch, we batched a training set of 540 samples into the system and implemented the optimization loop described in Figure 5-17b with a learning rate $\eta = 0.002$. We reserved part of the data ($N = 294$) to evaluate the trained model on inputs it had not seen before.

The top plot of Figure 5-18 shows the classification accuracy of both datasets during training. Our system achieves over 96% accuracy on the training set, and over 92% accuracy on the test set. When training a digital system, we found it also obtained similar accuracy on the test set. Each epoch batches the training set only three times through the system; two times to evaluate the derivative $\nabla_{\boldsymbol{\Delta}}\mathcal{L}(\boldsymbol{\Theta})$ and once more to evaluate $\mathcal{L}(\boldsymbol{\Theta})$ at the current parameter set $\boldsymbol{\Theta}$. We observed that the system quickly trained to an accuracy exceeding 80%, and then slowly asymptoted to a training accuracy of 96%. This behavior resembles the optimization trajectories of other first-order methods for training DNNs in electronics, such as stochastic gradient descent. Moreover, our system successfully trains using only 16-bit accuracies for the weights. Lower precision weights reduce memory requirements for training; however, digital systems are challenging to train with fewer than 32 bits due to numerical errors in gradients accumulating during backpropagation [161].

We also trained a digital model on the vowel classification task to benchmark the performance of *in situ* training on our system. The two models had the same number of neurons ($3 \times 6^2 = 108$), but the weights of the digital model, unlike those of our system, were unconstrained and could be arbitrary real matrices. We trained the system with a tanh nonlinearity, as we obtained very poor performance on the test set using a ReLU function. When training, we normalized the output with a softmax function and used the categorical cross-entropy loss function, as we did in the *in situ* training experiment.

The performance of the digital model is shown in Figure 5-19. The performance on the test set is similar to that obtained by our system. However, the digital model is significantly overfit, achieving perfect (100%) accuracy on the training set. One possible explanation for why our system does not overfit as much is the presence of analog noise, which has been suggested to function as regularization during DNN training [162].

5.7 Why does training work?

It may seem a bit surprising that our training algorithm, which is stochastically searching the parameter space, converges so efficiently to a high accuracy. Unlike other derivative-free optimizers, however, the advantage of the stochastic optimization approach we use is that it performs gradient descent on average. Here we illustrate this by adapting the proof provided in [158].

Suppose we are optimizing a DNN with model parameters $\boldsymbol{\Theta}$ and error functional $\mathcal{L}(\boldsymbol{\Theta})$. Gradient descent iteratively optimizes the parameters with the update rule:

$$\Delta\boldsymbol{\Theta} = -\eta\frac{\partial\mathcal{L}}{\partial\boldsymbol{\Theta}} \tag{5.10}$$

Assuming $\eta > 0$ and is sufficiently small, this update rule will converge to a local minimum of $\mathcal{L}(\boldsymbol{\Theta})$. Finite difference methods for analog hardware attempt to compute $\partial\mathcal{L}/\partial\boldsymbol{\Theta}$ by perturbing one parameter at a time in the system. Each epoch therefore requires $2N$ evaluations of the model on the training set for N model parameters.

Alternatively, one could perturb all model parameters at once by a random vector $\boldsymbol{\Pi} = [\pi_1, \pi_2, ..., \pi_N]$, where the elements of $\boldsymbol{\Pi}$ are randomly and independently chosen from an N-dimensional hypercube. The update rule here is:

$$\Delta\boldsymbol{\Theta} = -\mu\frac{\mathcal{L}(\boldsymbol{\Theta}+\boldsymbol{\Pi}) - \mathcal{L}(\boldsymbol{\Theta}-\boldsymbol{\Pi})}{2||\boldsymbol{\Pi}||}\boldsymbol{\Pi} \tag{5.11}$$

$$= -\frac{\mu}{2|\pi|\sqrt{N}}[\mathcal{L}(\boldsymbol{\Theta}+\boldsymbol{\Pi}) - \mathcal{L}(\boldsymbol{\Theta}-\boldsymbol{\Pi})]\boldsymbol{\Pi} \tag{5.12}$$

where μ is the learning rate. Assuming that the elements of $\boldsymbol{\Pi}$ are independently drawn from a Bernoulli distribution as $\pm\pi$, we can substitute $||\boldsymbol{\Pi}||$ as $|\pi|\sqrt{N}$. We note here that $\mu \neq \eta$, and in practice μ can be much larger than η while preserving stable convergence.

We Taylor expand the expression $[\mathcal{L}(\boldsymbol{\Theta}+\boldsymbol{\Pi}) - \mathcal{L}(\boldsymbol{\Theta}-\boldsymbol{\Pi})]$ as:

$$2\sum_i \frac{\partial\mathcal{L}}{\partial\theta_i}\pi_i \tag{5.13}$$

Substituting this into the update rule, we get:

$$\Delta\boldsymbol{\Theta} = -\frac{\mu}{|\pi|\sqrt{N}}\left(\sum_i \frac{\partial\mathcal{L}}{\partial\theta_i}\pi_i\right)\boldsymbol{\Pi} \tag{5.14}$$

$$= -\frac{\mu}{|\pi|\sqrt{N}} \left(\sum_i \frac{\partial \mathcal{L}}{\partial \theta_i} \pi_i \right) [\pi_1, \pi_2, ..., \pi_N] \tag{5.15}$$

Since the π_i are independently chosen, $\mathsf{E}[\pi_i \pi_j] = 0$ if $i \neq j$. Therefore, the expected parameter update $\mathsf{E}[\Delta \boldsymbol{\Theta}]$ is:

$$\mathsf{E}[\Delta \boldsymbol{\Theta}] = -\frac{\mu}{|\pi|\sqrt{N}} \left(\sum_i \frac{\partial \mathcal{L}}{\partial \theta_i} \mathsf{E}[\pi_i^2] \hat{x}_i \right) \tag{5.16}$$

$$= -\frac{\mu}{|\pi|\sqrt{N}} \left(\sum_i \frac{\partial \mathcal{L}}{\partial \theta_i} |\pi|^2 \hat{x}_i \right) \tag{5.17}$$

$$= -\frac{\mu |\pi|}{\sqrt{N}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}} \tag{5.18}$$

We therefore find that, on average, this procedure performs gradient descent with an effective learning rate $\eta = \mu|\pi|/\sqrt{N}$.

5.8 Discussion

An important DNN metric is the latency τ_{latency} of inference, i.e. the time delay between input of a vector and the DNN output. For the FICONN, τ_{latency} is dominated by the optical propagation delay, which we can estimate from the propagation length on chip to be $\sim$435 ps.

The FICONN's power consumption is dominated by the thermal phase shifters, which require $\sim$25 mW of electrical power to produce a π phase shift. Replacing these devices with low-power quasi-static phase shifters, and integrating high-speed modulators [163] at the transmitter, could push total energy consumption to $\sim$ 10 fJ/OP for large systems, while maintaining ns latencies and throughputs of thousands of TOPS. As a point of comparison, electronic systolic arrays such as the tensor processing unit (TPU) require at minimum $N + 1$ clock cycles for a single $N \times N$ matrix-vector multiplication. A three-layer DNN with 256 neurons would therefore require $\sim$1 μs to compute at a 700 MHz clock speed [13], which is more than two orders of magnitude longer than in a photonic processor.

Low inference latency in the FICONN could be used to improve the speed of model training, which consumes significant power [164] and has motivated work on more efficient scheduling algorithms [165]. *In situ* training could also ultimately improve the generalization of DNN models, as training with noise has been suggested to regularize models, preventing overfitting [162] and improving adversarial robustness [166] to small perturbations in the input. Training *in situ* can implement this regularization automatically by leveraging quantum noise in hardware. We observed this effect in our own experiments; while both the FICONN and a digital system obtained similar performance on the test set for the classification task studied, the digital system overfit the model, achieving perfect accuracy on the training set. Lastly, our implementation of *in situ* training, which does not require a digital system for computing gradients, is compatible with feedback-based "self-learning" photonic devices [16, 167], enabling fast, autonomous training of models

without any required external input.

The FICONN architecture, which was implemented in a foundry-fabricated photonic integrated circuit, could be scaled to larger sizes with current-day technologies. Silicon photonic foundries have already produced functional systems of up to tens of thousands of components [168]. Spectral multiplexing, for instance through integration of microcomb sources with silicon photonics [169], can enable classification of data simultaneously across many wavelength channels, further reducing energy consumption and increasing throughput. The system's energy consumption would further improve by optimization of the NOFU; while our implementation makes use of microring resonators, photonic crystal modulators [170], microdisks [163], or hybrid integration of lithium niobate [171, 172] can further reduce the optical power required to trigger the nonlinearity. While our implementation of the FICONN makes use of feedforward unitary circuits, which implement fully-connected layers in a DNN, this architecture can also be generalized to other types of neural networks. For example, temporal or frequency data may be classified using recirculating waveguide meshes [69], which can implement feedback and resonant filters. Such a system, where phase shifter settings are trained *in situ* [69, 173], could be used for intelligent processing of microwave signals in the optical domain.

5.9 Scaling

The FICONN architecture with N modes and M layers performs $2MN^2 + 2(M-1)N$ operations per inference, where the first term accounts for linear matrix operations and the second term refers to the nonlinear activation function. For large N the first term dominates and we approximate the total number of operations as $2MN^2$.

The total energy consumption per operation of the system for a single inference can therefore be approximated as $\tau_{\text{latency}} P_{\text{total}}/(2MN^2)$, where P_{total} is the total power consumption of the photonics, drivers, and readout electronics and τ_{latency} is the time required for a single inference. The system requires MN^2 phase shifters, N transmitters, N receivers, and MN nonlinear optical function units, making the total power consumption $P_{\text{total}} = MN^2 P_{\text{PS}} + MNP_{\text{NOFU}} + N(P_{\text{TX}} + P_{\text{ICR}})$. Dividing the FICONN's energy consumed during τ_{latency} by N_{OPS} upper-bounds the energy-per-operation as

$$E_{\text{OP}} \approx \frac{\tau_{\text{latency}}}{2}\left[P_{\text{PS}} + \frac{P_{\text{NOFU}}}{N} + \frac{P_{\text{TX}} + P_{\text{ICR}}}{MN}\right], \tag{5.19}$$

Our device performs $2MN^2 + 2(M-1)N = 240$ operations per inference, where $M = 3$ and $N = 6$. The phase shifters require about 25 mW per π phase shift; as the internal phase shifters only require up to π phase shift, while the external phase shifters require up to 2π, we assume the average power consumption per phase shifter is 18.75 mW. The phase shifter contribution to the energy per operation is therefore $144 \times (18.75\text{ mW}) \times (435\text{ ps})/(240\text{ OPs}) = 4.9\text{ pJ/OP}$, where we include phase shifters for both model parameters and the transmitter. This energy requirement would reduce substantially with the use of undercut thermal phase shifters [39], which reduce power dissipation by an order of magnitude, or MEMS-actuated devices [92, 174], both of which

are available in silicon photonic foundries. The nonlinear optical function unit consumes 60 μW of power, which contributes about $12 \times (60\ \mu\text{W}) \times (435\ \text{ps})/(240\ \text{OPs}) = 1.3$ fJ/OP to this total.

In principle, P_{PS} could be zero through the use of nonvolatile phase shifters, such as phase change materials, or even through ultra-low power phase shifters such as MEMS devices, which exhibit static power dissipations on the order of fW [174]. As a result, larger system sizes would have an asymptoptic decrease in the energy/OP as $1/N$.

This is fundamentally a different paradigm from digital systems, which have a fixed energy cost of 100-1000 fJ/OP regardless of the model size. Here, the energy cost for $O(N^2)$ operations scales as $O(N)$; thus, very large system sizes could potentially realize energy consumptions at the attojoule/OP level.

5.10 Conclusion

In this chapter, we discussed the demonstration of a coherent optical DNN on a single chip that performs both inference and *in situ* training. The FICONN system introduces inline nonlinear activation functions based on modulators driven by "receiverless" photodection, eliminating the latency and power consumption introduced by optical-to-electrical conversion between DNN layers and preserving phase information for optical data to be processed coherently. The system fabrication relied entirely on commercial foundry photolithography, potentially enabling scaling to wafer-level systems.

Moreover, we have demonstrated *in situ* training of DNNs by estimating derivatives of model parameters directly on hardware. Our approach is also generalizable to other photonic DNN hardware being currently studied. *In situ* training, which takes advantage of the optically-accelerated forward pass enabled by receiverless hardware, opens the path to a new generation of devices that learn in real time for sensing, autonomous driving, and telecommunications.

6 The road ahead

6.1 Introduction

Programmable silicon photonic systems show promise to be a flexible, scalable platform for accelerating tasks in computation. In this thesis, we have focused on several key challenges for building up these systems, including addressing analog hardware errors, realizing scalable packaging, and control, calibration, and algorithms for realizing end-to-end, photonic processing of deep neural networks. However, many challenges remain; in this section, I conclude by discussing key research questions to be addressed in the future.

6.2 Scalable, error-corrected photonic meshes

In Chapter 3, we reported the development of the first deterministic, gate-by-gate error correction algorithm for programmable photonic meshes. Our algorithm, which is generalizable to any programmable architecture of Mach-Zehnder interferometers, enables scalable, accurate optical computation in photonic systems of up to hundreds of channels. As we move forward in scaling these systems, future questions to consider include:

- Our work focused primarily on coherent errors introduced by beamsplitter imperfections. While we found device loss to have little effect on accuracy once error correction is applied, this may not hold true for extremely large circuits. Are there algorithms that can correct for the incoherent errors introduced by device loss?

- Following our work on local error correction, we developed self-configuration algorithms that can progressively error correct a rectangular mesh using feedback only from the circuit output. Are there ways to apply these algorithms to generic circuit architectures?

- As these systems scale up, increasing numbers of devices will not be programmable to the correct setting due to the Haar distribution. Are there ways to prune these devices or replace them with passive components to improve computation accuracy?

- Are there neural network architectures that are particularly robust to error in these systems? Importantly, will they exhibit performances comparable to state-of-the-art DNN architectures today?

6.3 Large-scale, multi-chip photonic modules

Chapter 4 reported the development of a novel, "alignment-free" paradigm for chip-to-chip photonic interconnects. This approach greatly relaxes the alignment tolerances required for photonic assembly, potentially enabling the use of high-volume, low-precision packaging tools. Here, future questions to consider include:

- The main tradeoff we make to realize alignment tolerance is the increased demands on fabrication tolerance. Are there strategies to relax this tolerance, perhaps by making use of ideas from adiabatic coupling?

- While we developed an analytical formalism, we found in practice when designing experimental devices that it still required tedious, finite-difference time-domain simulations. Can we further extend our analytical theory, perhaps using ideas from eigenmode expansion (EME), to simplify device design?

- Efficient coupling between two different material platforms requires the interconnect waveguides have the same effective index. This may be challenging for material systems that have dramatically different refractive indices. Are there ways to simplify the mode-matching process for these cases?

- There is a tradeoff between angular tolerance and insertion loss, as a sudden, non-adiabatic intersection between the waveguides produces radiation loss. Are there strategies to relax this tolerance, perhaps by adiabatically varying the intersection angle?

6.4 Energy-efficient, high-speed photonic nonlinearities

A key element of the demonstration in Chapter 5 was the realization of an in-line, coherent optical nonlinearity that enabled programmable activation functions for DNNs. Future work in developing these systems may consider:

- We realize extremely efficient optical nonlinearities by making use of carrier injection. The main drawback here is that it is difficult to realize bandwidths exceeding 1 GHz due to recombination lifetimes. Are there alternative ways to realize higher speeds?

- Can we further reduce activation energies for these devices, perhaps with closer integration (thereby reducing device capacitances) or more efficient resonators (such as photonic crystal devices)?

- The nonlinear function unit we report is dependent on the instantaneous optical signal incident on the device. Are there ways to incorporate hysteresis or memory into such a device? Such a device would have applications for emerging classes of AI models, such as recurrent neural networks (RNNs) and transformer models used in natural language processing.

- Our results in Chapter 5 show that we can realize functions that emulate nonlinearities used in DNN training, such as the rectified linear unit (ReLU) function. Are there ways to realize sharper nonlinear functions or true bistabilities? For example, is it possible to realize a true, all-optical comparator that can implement a ReLU function?

6.5 Optically accelerated neural network training

This work reported in this thesis culminated with the demonstration, in Chapter 5, of end-to-end photonic training of deep neural networks. We obtained accuracies comparable to digital system by evaluating gradients of model parameters directly on the hardware. Future research directions in this area might include:

- We used a "parallel perturbation" scheme that can be shown to estimate gradient descent. However, there is an algorithmic slowdown relative to true gradient descent. How does this slowdown scale, and are there more efficient algorithms for training in optics that do not suffer from this drawback?

- While inference in our system was clockless, our training still operated on an effective "clock," as each cycle required: 1) perturbing the weights; 2) computing the error function; and 3) updating the model parameters. Are there ways to make use of fast analog feedback to realize truly clockless, "speed-of-light-limited" training of neural networks?

- Backpropagation is the *de facto* standard for digital training of DNNs. While some work has been done in realizing these systems in optics, it has several drawbacks, including: 1) low precision, making scaling to deep networks challenging; 2) requiring evaluation of activation gradients on digital hardware; and 3) requiring multiple evaluations of the training set on the hardware per epoch. Are there ways to implement effective gradient descent without these limitations? Recent work on frequency-multiplexed gradient descent in analog hardware suggests this is possible.

- Do we need to backpropagate for efficient training? Recent work on direct feedback alignment (DFA) suggests this may not be needed. Are there ways to map such an algorithm to a single-shot, deep photonic neural network?

Bibliography

[1] Dennard, R., Gaensslen, F., Yu, H.-N., Rideout, V., Bassous, E. & LeBlanc, A. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits* **9**, 256–268 (1974).

[2] Rupp, K. Microprocessor Trend Data (2023). URL `https://github.com/karlrupp/microprocessor-trend-data`.

[3] Miller, D. A. B. Attojoule Optoelectronics for Low-Energy Information Processing and Communications. *Journal of Lightwave Technology* **35** (2017).

[4] Esmaeilzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K. & Burger, D. Dark silicon and the end of multicore scaling. *SIGARCH Comput. Archit. News* **39**, 365–376 (2011).

[5] Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (2012).

[6] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (2016).

[7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. & Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 1877–1901 (2020).

[8] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss,

M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C. & Silver, D. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).

[9] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. & Hassabis, D. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).

[10] Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Le, Q. V., Laudon, J., Ho, R., Carpenter, R. & Dean, J. A graph placement methodology for fast chip design. *Nature* **594**, 207–212 (2021).

[11] Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Networks* **2**, 183–192 (1989).

[12] Papers with Code - ImageNet Benchmark (Image Classification). URL `https://paperswithcode.com/sota/image-classification-on-imagenet`.

[13] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E. & Yoon, D. H. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ISCA '17, 1–12 (Association for Computing Machinery, New York, NY, USA, 2017).

[14] Shao, Y. S., Clemons, J., Venkatesan, R., Zimmer, B., Fojtik, M., Jiang, N., Keller, B., Klinefelter, A., Pinckney, N., Raina, P., Tell, S. G., Zhang, Y., Dally, W. J., Emer, J., Gray, C. T., Khailany, B. & Keckler, S. W. Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 14–27 (ACM, 2019).

[15] Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D. & Soljačić, M. Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11**, 441–446 (2017).

[16] Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. & Pernice, W. H. P. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569**, 208–214 (2019).

[17] Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A. S., Liu, J., Wright, C. D., Sebastian, A., Kippenberg, T. J., Pernice, W. H. P. & Bhaskaran, H. Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58 (2021).

[18] Xu, X., Tan, M., Corcoran, B., Wu, J., Boes, A., Nguyen, T. G., Chu, S. T., Little, B. E., Hicks, D. G., Morandotti, R., Mitchell, A. & Moss, D. J. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589**, 44–51 (2021).

[19] Poberaj, G., Hu, H., Sohler, W. & Gunter, P. Lithium niobate on insulator (LNOI) for micro-photonic devices. *Lasers and Photonics Reviews* **6**, 488–503 (2012).

[20] Dietrich, C. P., Fiore, A., Thompson, M. G., Kamp, M. & Hofling, S. GaAs integrated quantum photonics: Towards compact and multi-functional quantum photonic integrated circuits. *Lasers and Photonics Reviews* **10**, 870–94 (2016).

[21] Smit, M. K. InP photonic integrated circuits. In *The 15th Annual Meeting of the IEEE Lasers and Electro-Optics Society*, vol. 2, 843–844 vol.2 (2002).

[22] Lu, T.-J., Fanto, M., Choi, H., Thomas, P., Steidle, J., Mouradian, S., Kong, W., Zhu, D., Moon, H., Berggren, K., Kim, J., Soltani, M., Preble, S. & Englund, D. Aluminum nitride integrated photonics platform for the ultraviolet to visible spectrum. *Opt. Express* **26**, 11147–11160 (2018).

[23] Fujisawa, T., Makino, S., Sato, T. & Saitoh, K. Low-loss, compact, and fabrication-tolerant Si-wire 90 degree waveguide bend using clothoid and normal curves for large scale photonic integrated circuits. *Opt. Express* **25**, 9150–9159 (2017).

[24] Harris, N. C., Steinbrecher, G. R., Prabhu, M., Lahini, Y., Mower, J., Bunandar, D., Chen, C., Wong, F. N. C., Baehr-Jones, T., Hochberg, M., Lloyd, S. & Englund, D. Quantum transport simulations in a programmable nanophotonic processor. *Nature Photonics* **11**, 447–452 (2017).

[25] Sparrow, C., Martín-López, E., Maraviglia, N., Neville, A., Harrold, C., Carolan, J., Joglekar, Y. N., Hashimoto, T., Matsuda, N., O'Brien, J. L., Tew, D. P. & Laing, A. Simulating the vibrational quantum dynamics of molecules using photonics. *Nature* **557**, 660–667 (2018).

[26] Qiang, X., Zhou, X., Wang, J., Wilkes, C. M., Loke, T., O'Gara, S., Kling, L., Marshall, G. D., Santagati, R., Ralph, T. C., Wang, J. B., O'Brien, J. L., Thompson, M. G. & Matthews, J. C. F. Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. *Nature Photonics* **12**, 534–539 (2018).

[27] Novack, A., Streshinsky, M., Huynh, T., Galfsky, T., Guan, H., Liu, Y., Ma, Y., Shi, R., Horth, A., Chen, Y., Hanjani, A., Roman, J., Dziashko, Y., Ding, R., Fathololoumi, S., Lim, A. E.-J., Padmaraju, K., Sukkar, R., Younce, R., Rohde, H., Palmer, R., Saathoff, G., Wuth, T., Bohn, M., Ahmed, A., Ahmed, M., Williams, C., Lim, D., Elmoznine, A., Rylyakov, A., Baehr-Jones, T., Magill, P., Scordo, D. & Hochberg, M. A Silicon Photonic Transceiver and Hybrid Tunable Laser for 64 Gbaud Coherent Communication. In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 1–3 (2018).

[28] Zhuang, L., Roeloffzen, C. G. H., Hoekman, M., Boller, K.-J. & Lowery, A. J. Programmable photonic signal processor chip for radiofrequency applications. *Optica* **2**, 854 (2015).

[29] Bogaerts, W., Pérez, D., Capmany, J., Miller, D. A. B., Poon, J., Englund, D., Morichetti, F. & Melloni, A. Programmable photonic circuits. *Nature* **586**, 207–216 (2020).

[30] Chrostowski, L. & Hochberg, M. *Silicon Photonics Design: From Devices to Systems* (Cambridge University Press, 2015).

[31] Joannopoulos, J. D., Johnson, S. G., Winn, J. N. & Meade, R. D. *Photonic Crystals: Molding the Flow of Light* (Princeton University Press, 2008).

[32] Garcia, V. M., Vidal, A., Boria, V. E. & Vidal, A. M. Efficient and accurate waveguide mode computation using BI-RME and Lanczos methods. *International Journal for Numerical Methods in Engineering* **65**, 1773–88 (2005).

[33] Giewont, K., Hu, S., Peng, B., Rakowski, M., Rauch, S., Rosenberg, J. C., Sahin, A., Stobert, I., Stricker, A., Nummy, K., Anderson, F. A., Ayala, J., Barwicz, T., Bian, Y., Dezfulian, K. K., Gill, D. M. & Houghton, T. 300-mm Monolithic Silicon Photonics Foundry Technology. *IEEE Journal of Selected Topics in Quantum Electronics* **25**, 1–11 (2019).

[34] Teng, M., Niu, B., Han, K., Kim, S., Xuan, Y., Lee, Y. J. & Qi, M. Trident Shape SOI Metamaterial Fiber-to-Chip Edge Coupler. In *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 1–3 (2019).

[35] Papes, M., Cheben, P., Benedikovic, D., Schmid, J. H., Pond, J., Halir, R., Ortega-Moñux, A., Wangüemert-Pérez, G., Ye, W. N., Xu, D.-X., Janz, S., Dado, M. & Vašinek, V. Fiber-chip edge coupler with large mode size for silicon photonic wire waveguides. *Optics Express* **24**, 5026–5038 (2016).

[36] Sun, X., Liu, H.-C. & Yariv, A. Adiabaticity criterion and the shortest adiabatic mode transformer in a coupled-waveguide system. *Optics Letters* **34**, 280 (2009).

[37] Bogaerts, W., Heyn, P. D., Vaerenbergh, T. V., Vos, K. D., Selvaraja, S. K., Claes, T., Dumon, P., Bienstman, P., Thourhout, D. V. & Baets, R. Silicon microring resonators. *Lasers and Photonics Reviews* **6**, 47–73 (2012).

[38] Harris, N. C., Ma, Y., Mower, J., Baehr-Jones, T., Englund, D., Hochberg, M. & Galland, C. Efficient, compact and low loss thermo-optic phase shifter in silicon. *Optics Express* **22**, 10487 (2014).

[39] Dong, P., Qian, W., Liang, H., Shafiiha, R., Feng, D., Li, G., Cunningham, J. E., Krishnamoorthy, A. V. & Asghari, M. Thermally tunable silicon racetrack resonators with ultralow tuning power. *Optics Express* **18**, 20298 (2010).

[40] Soref, R. & Bennett, B. Electrooptical effects in silicon. *IEEE Journal of Quantum Electronics* **23**, 123–129 (1987).

[41] Xu, Q., Manipatruni, S., Schmidt, B., Shakya, J. & Lipson, M. 12.5 Gbit/s carrier-injection-based silicon micro-ring silicon modulators. *Optics Express* **15**, 430 (2007).

[42] Streshinsky, M., Ding, R., Liu, Y., Novack, A., Yang, Y., Ma, Y., Tu, X., Chee, E. K. S., Lim, A. E.-J., Lo, P. G.-Q., Baehr-Jones, T. & Hochberg, M. Low power 50 Gb/s silicon traveling wave Mach-Zehnder modulator near 1300 nm. *Opt. Express* **21**, 30350–30357 (2013).

[43] Lischke, S., Peczek, A., Morgan, J. S., Sun, K., Steckler, D., Yamamoto, Y., Korndörfer, F., Mai, C., Marschmeyer, S., Fraschke, M., Krüger, A., Beling, A. & Zimmermann, L. Ultra-fast germanium photodiode with 3-dB bandwidth of 265 GHz. *Nature Photonics* **15**, 925–931 (2021).

[44] Byrd, M. J., Timurdogan, E., Su, Z., Poulton, C. V., Fahrenkopf, N. M., Leake, G., Coolbaugh, D. D. & Watts, M. R. Mode-evolution-based coupler for high saturation power Ge-on-Si photodetectors. *Optics Letters* **42**, 851 (2017).

[45] Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S. & Walsmley, I. A. Optimal design for universal multiport interferometers. *Optica* **3**, 1460 (2016).

[46] Reck, M., Zeilinger, A., Bernstein, H. J. & Bertani, P. Experimental realization of any discrete unitary operator. *Physical Review Letters* **73**, 58–61 (1994).

[47] Carolan, J., Harrold, C., Sparrow, C., Martin-Lopez, E., Russell, N. J., Silverstone, J. W., Shadbolt, P. J., Matsuda, N., Oguma, M., Itoh, M., Marshall, G. D., Thompson, M. G., Matthews, J. C. F., Hashimoto, T., O'Brien, J. L. & Laing, A. Universal linear optics. *Science* **349**, 711–716 (2015).

[48] Annoni, A., Guglielmi, E., Carminati, M., Ferrari, G., Sampietro, M., Miller, D. A., Melloni, A. & Morichetti, F. Unscrambling light—automatically undoing strong mixing between modes. *Light: Science & Applications* **6**, e17110 (2017).

[49] Wang, J., Paesani, S., Ding, Y., Santagati, R., Skrzypczyk, P., Salavrakos, A., Tura, J., Augusiak, R., Mančinska, L., Bacco, D., Bonneau, D., Silverstone, J. W., Gong, Q., Acín, A., Rottwitt, K., Oxenløwe, L. K., O'Brien, J. L., Laing, A. & Thompson, M. G. Multidimensional quantum entanglement with large-scale integrated optics. *Science* **360**, 285–291 (2018).

[50] Milanizadeh, M., Borga, P., Morichetti, F., Miller, D. & Melloni, A. Manipulating Free-space Optical Beams with a Silicon Photonic Mesh. In *2019 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, 1–2 (2019).

[51] Prabhu, M., Roques-Carmes, C., Shen, Y., Harris, N., Jing, L., Carolan, J., Hamerly, R., Baehr-Jones, T., Hochberg, M., Čeperić, V., Joannopoulos, J. D., Englund, D. R. & Soljačić, M. Accelerating recurrent Ising machines in photonic integrated circuits. *Optica* **7**, 551 (2020).

[52] Bandyopadhyay, S., Hamerly, R. & Englund, D. R. Error Correction for Programmable Photonics, US Patent Application 17/556,033 (2021).

[53] Hamerly, R., Bandyopadhyay, S. & Englund, D. R. Self-Configuration and Error Correction in Linear Photonic Circuits, US Patent Application 17/711,640 (2022).

[54] Hamerly, R., Bandyopadhyay, S. & Englund, D. Stability of self-configuring large multiport interferometers. *Phys. Rev. Applied* **18**, 024018 (2022).

[55] Hamerly, R., Bandyopadhyay, S. & Englund, D. Accurate self-configuration of rectangular multiport interferometers. *Phys. Rev. Applied* **18**, 024019 (2022).

[56] Basani, J. R., Vadlamani, S. K., Bandyopadhyay, S., Englund, D. R. & Hamerly, R. A self-similar sine-cosine fractal architecture for multiport interferometers. *Nanophotonics* **12**, 975–984 (2023).

[57] Hamerly, R., Bandyopadhyay, S. & Englund, D. Asymptotically-fault tolerant programmable photonics. *Nature Communications* **13**, 6831 (2022).

[58] Bandyopadhyay, S., Hamerly, R. & Englund, D. Hardware error correction for programmable photonics. *Optica* **8**, 1247–1255 (2021).

[59] Miller, D. A. B. Self-configuring universal linear optical component. *Photonics Research* **1**, 1 (2013).

[60] Mikkelsen, J. C., Sacher, W. D. & Poon, J. K. S. Dimensional variation tolerant silicon-on-insulator directional couplers. *Optics Express* **22**, 3145 (2014).

[61] Fang, M. Y.-S., Manipatruni, S., Wierzynski, C., Khosrowshahi, A. & DeWeese, M. R. Design of optical neural networks with component imprecisions. *Optics Express* **27**, 14009 (2019).

[62] Pérez, D., Gasulla, I. & Capmany, J. Field-programmable photonic arrays. *Optics Express* **26**, 27265 (2018).

[63] Pérez, D., Gasulla, I., Capmany, J. & Soref, R. A. Reconfigurable lattice mesh designs for programmable photonic processors. *Optics Express* **24**, 12093 (2016).

[64] Zand, I. & Bogaerts, W. Effects of coupling and phase imperfections in programmable photonic hexagonal waveguide meshes. *Photonics Research* **8**, 211 (2020).

[65] Lu, Z., Jhoja, J., Klein, J., Wang, X., Liu, A., Flueckiger, J., Pond, J. & Chrostowski, L. Performance prediction for silicon photonics integrated circuits with layout-dependent correlated manufacturing variability. *Optics Express* **25**, 9712 (2017).

[66] Burgwal, R., Clements, W. R., Smith, D. H., Gates, J. C., Kolthammer, W. S., Renema, J. J. & Walmsley, I. A. Using an imperfect photonic network to implement random unitaries. *Optics Express* **25**, 28236 (2017).

[67] López, A., Pérez, D., DasMahapatra, P. & Capmany, J. Auto-routing algorithm for field-programmable photonic gate arrays. *Optics Express* **28**, 737 (2020).

[68] López, D. P. Programmable Integrated Silicon Photonics Waveguide Meshes: Optimized Designs and Control Algorithms. *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1–12 (2020).

[69] Pérez-López, D., López, A., DasMahapatra, P. & Capmany, J. Multipurpose self-configuration of programmable photonic circuits. *Nature Communications* **11**, 6359 (2020).

[70] Mower, J., Harris, N. C., Steinbrecher, G. R., Lahini, Y. & Englund, D. High-fidelity quantum state evolution in imperfect photonic integrated circuits. *Physical Review A* **92**, 032322 (2015).

[71] Pai, S., Bartlett, B., Solgaard, O. & Miller, D. A. B. Matrix Optimization on Universal Unitary Photonic Devices. *Physical Review Applied* **11**, 064044 (2019).

[72] Miller, D. A. B. Setting up meshes of interferometers – reversed local light interference method. *Optics Express* **25**, 29233 (2017).

[73] Pai, S., Williamson, I. A. D., Hughes, T. W., Minkov, M., Solgaard, O., Fan, S. & Miller, D. A. B. Parallel Programming of an Arbitrary Feedforward Photonic Network. *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1–13 (2020).

[74] Miller, D. A. B. Self-aligning universal beam coupler. *Optics Express* **21**, 6360 (2013).

[75] Pérez-López, D., Gutierrez, A. M., Sánchez, E., DasMahapatra, P. & Capmany, J. Integrated photonic tunable basic units using dual-drive directional couplers. *Optics Express* **27**, 38071 (2019).

[76] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

[77] Russell, N. J., Chakhmakhchyan, L., O'Brien, J. L. & Laing, A. Direct dialling of Haar random unitary matrices. *New Journal of Physics* **19**, 033007 (2017).

[78] Williamson, I. A. D., Hughes, T. W., Minkov, M., Bartlett, B., Pai, S. & Fan, S. Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks. *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1–12 (2020).

[79] Arjovsky, M., Shah, A. & Bengio, Y. Unitary Evolution Recurrent Neural Networks. In *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48 of *Proceedings of Machine Learning Research*, 1120–1128 (PMLR, New York, New York, USA, 2016).

[80] Streshinsky, M., Novack, A., Ding, R., Liu, Y., Lim, A. E., Lo, P. G., Baehr-Jones, T. & Hochberg, M. Silicon Parallel Single Mode 48 $\times$ 50 Gb/s Modulator and Photodetector Array. *Journal of Lightwave Technology* **32**, 4370–4377 (2014).

[81] Watts, M. R., Zortman, W. A., Trotter, D. C., Young, R. W. & Lentine, A. L. Low-Voltage, Compact, Depletion-Mode, Silicon Mach–Zehnder Modulator. *IEEE Journal of Selected Topics in Quantum Electronics* **16**, 159–164 (2010).

[82] Zhang, Y., Yang, S., Yang, Y., Gould, M., Ophir, N., Lim, A. E.-J., Lo, G.-Q., Magill, P., Bergman, K., Baehr-Jones, T. & Hochberg, M. A high-responsivity photodetector absent metal-germanium direct contact. *Optics Express* **22**, 11367 (2014).

[83] Madsen, C. K. & Lenz, G. Optical all-pass filters for phase response design with applications for dispersion compensation. *IEEE Photonics Technology Letters* **10**, 994–996 (1998).

[84] Mower, J., Zhang, Z., Desjardins, P., Lee, C., Shapiro, J. H. & Englund, D. High-dimensional quantum key distribution using dispersive optics. *Physical Review A* **87**, 062322 (2013).

[85] Notaros, J., Mower, J., Heuck, M., Lupo, C., Harris, N. C., Steinbrecher, G. R., Bunandar, D., Baehr-Jones, T., Hochberg, M., Lloyd, S. & Englund, D. Programmable dispersion on a photonic integrated circuit for classical and quantum applications. *Optics Express* **25**, 21275 (2017).

[86] Mason, S. J. Feedback Theory-Some Properties of Signal Flow Graphs. *Proceedings of the IRE* **41**, 1144–1156 (1953).

[87] Mason, S. J. Feedback Theory-Further Properties of Signal Flow Graphs. *Proceedings of the IRE* **44**, 920–926 (1956).

[88] Powell, M. J. D. Direct search algorithms for optimization calculations. *Acta Numerica* **7**, 287–336 (1998).

[89] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**, 261–272 (2020).

[90] Yang, Y., Ma, Y., Guan, H., Liu, Y., Danziger, S., Ocheltree, S., Bergman, K., Baehr-Jones, T. & Hochberg, M. Phase coherence length in silicon photonic platform. *Optics Express* **23**, 16890 (2015).

[91] Jacques, M., Samani, A., El-Fiky, E., Patel, D., Xing, Z. & Plant, D. V. Optimization of thermo-optic phase-shifter design and mitigation of thermal crosstalk on the SOI platform. *Optics Express* **27**, 10456 (2019).

[92] Baghdadi, R., Gould, M., Gupta, S., Tymchenko, M., Bunandar, D., Ramey, C. & Harris, N. C. Dual slot-mode NOEM phase shifter. *Optics Express* **29**, 19113 (2021).

[93] Wilmart, Q., Brision, S., Hartmann, J.-M., Myko, A., Ribaud, K., Petit-Etienne, C., Youssef, L., Fowler, D., Charbonnier, B., Sciancalepore, C., Pargon, E., Bernabé, S. & Szelag, B. A Complete Si Photonics Platform Embedding Ultra-Low Loss Waveguides for O- and C-Band. *Journal of Lightwave Technology* **39**, 532–538 (2021).

[94] Bell, B. A. & Walmsley, I. A. Further Compactifying Linear Optical Unitaries. *arXiv:2104.07561* (2021).

[95] Zhang, Y. & Ashe, J. Designing a High Performance TEC Controller. Tech. Rep., Analog Devices (2002).

[96] Komma, J., Schwarz, C., Hofmann, G., Heinert, D. & Nawrodt, R. Thermo-optic coefficient of silicon at 1550 nm and cryogenic temperatures. *Applied Physics Letters* **101**, 041905 (2012).

[97] Guan, H., Ma, Y., Shi, R., Zhu, X., Younce, R., Chen, Y., Roman, J., Ophir, N., Liu, Y., Ding, R., Baehr-Jones, T., Bergman, K. & Hochberg, M. Compact and low loss 90° optical hybrid on a silicon-on-insulator platform. *Optics Express* **25**, 28957 (2017).

[98] Suzuki, K., Cong, G., Tanizawa, K., Kim, S.-H., Ikeda, K., Namiki, S. & Kawashima, H. Ultra-high-extinction-ratio 2 $\times$ 2 silicon optical switch with variable splitter. *Optics Express* **23**, 9086 (2015).

[99] Wang, M., Ribero, A., Xing, Y. & Bogaerts, W. Tolerant, broadband tunable 2 $\times$ 2 coupler circuit. *Optics Express* **28**, 5555 (2020).

[100] Bandyopadhyay, S. & Englund, D. Alignment-free photonic interconnects. *arXiv:2110.12851* (2021).

[101] Barwicz, T., Lichoulas, T. W., Taira, Y., Martin, Y., Takenobu, S., Janta-Polczynski, A., Numata, H., Kimbrell, E. L., Nah, J.-W., Peng, B., Childers, D., Leidy, R., Khater, M., Kamlapurkar, S., Cyr, E., Engelmann, S., Fortier, P. & Boyer, N. Automated, high-throughput photonic packaging. *Optical Fiber Technology* **44**, 24–35 (2018).

[102] Live Blog: Intel Innovation Day 1. URL https://www.intel.com/content/www/us/en/newsroom/news/2022-intel-innovation-live-blog-day-1.html.

[103] Davenport, M. L., Skendžić, S., Volet, N., Hulme, J. C., Heck, M. J. R. & Bowers, J. E. Heterogeneous Silicon/III–V Semiconductor Optical Amplifiers. *IEEE Journal of Selected Topics in Quantum Electronics* **22**, 78–88 (2016).

[104] Corato-Zanarella, M., Gil-Molina, A., Ji, X., Shin, M. C., Mohanty, A. & Lipson, M. Widely tunable and narrow-linewidth chip-scale lasers from near-ultraviolet to near-infrared wavelengths. *Nature Photonics* **17**, 157–164 (2023).

[105] Wan, N. H., Lu, T.-J., Chen, K. C., Walsh, M. P., Trusheim, M. E., De Santis, L., Bersin, E. A., Harris, I. B., Mouradian, S. L., Christen, I. R., Bielejec, E. S. & Englund, D. Large-scale integration of artificial atoms in hybrid photonic circuits. *Nature* **583**, 226–231 (2020).

[106] He, M., Xu, M., Ren, Y., Jian, J., Ruan, Z., Xu, Y., Gao, S., Sun, S., Wen, X., Zhou, L., Liu, L., Guo, C., Chen, H., Yu, S., Liu, L. & Cai, X. High-performance hybrid silicon and lithium niobate Mach-Zehnder modulators for 100 Gbit/s and beyond. *Nature Photonics* **13**, 359–364 (2019).

[107] Dangel, R., La Porta, A., Jubin, D., Horst, F., Meier, N., Seifried, M. & Offrein, B. J. Polymer Waveguides Enabling Scalable Low-Loss Adiabatic Optical Coupling for Silicon Photonics. *IEEE Journal of Selected Topics in Quantum Electronics* **24**, 1–11 (2018).

[108] Soganci, I. M., La Porta, A. & Offrein, B. J. Flip-chip optical couplers with scalable I/O count for silicon photonics. *Optics Express* **21**, 16075 (2013).

[109] Tiecke, T. G., Nayak, K. P., Thompson, J. D., Peyronel, T., de Leon, N. P., Vuletić, V. & Lukin, M. D. Efficient fiber-optical interface for nanophotonic devices. *Optica* **2**, 70 (2015).

[110] Lindenmann, N., Dottermusch, S., Goedecke, M. L., Hoose, T., Billah, M. R., Onanuga, T. P., Hofmann, A., Freude, W. & Koos, C. Connecting Silicon Photonic Circuits to Multicore Fibers by Photonic Wire Bonding. *Journal of Lightwave Technology* **33**, 755–760 (2015).

[111] Lindenmann, N., Balthasar, G., Hillerkuss, D., Schmogrow, R., Jordan, M., Leuthold, J., Freude, W. & Koos, C. Photonic wire bonding: a novel concept for chip-scale interconnects. *Optics Express* **20**, 17667 (2012).

[112] Dietrich, P.-I., Blaicher, M., Reuter, I., Billah, M., Hoose, T., Hofmann, A., Caer, C., Dangel, R., Offrein, B., Troppenz, U., Moehrle, M., Freude, W. & Koos, C. In situ 3D nanoprinting of free-form coupling elements for hybrid photonic integration. *Nature Photonics* **12**, 241–247 (2018).

[113] Scarcella, C., Gradkowski, K., Carroll, L., Lee, J.-S., Duperron, M., Fowler, D. & O'Brien, P. Pluggable Single-Mode Fiber-Array-to-PIC Coupling Using Micro-Lenses. *IEEE Photonics Technology Letters* **29**, 1943–1946 (2017).

[114] Mangal, N., Missinne, J., Campenhout, J. V., Snyder, B. & Steenberge, G. V. Ball Lens Embedded Through-Package Via To Enable Backside Coupling Between Silicon Photonics Interposer and Board-Level Interconnects. *Journal of Lightwave Technology* **38**, 2360–2369 (2020).

[115] Hwang, H. Y., Morrissey, P., Lee, J. S., O'Brien, P., Henriksson, J., Wu, M. C. & Seok, T. J. 128 $\times$ 128 silicon photonic MEMS switch package using glass interposer and pitch reducing fibre array. In *2017 IEEE 19th Electronics Packaging Technology Conference (EPTC)*, 1–4 (2017).

[116] Pashkova, T. & O'Brien, P. Development of Silicon Grating-to-Grating coupling technology and Demonstration of Fan-In/Fan-Out for Multi-Core Fiber applications. In *2019 IEEE 21st Electronics Packaging Technology Conference (EPTC)*, 582–585 (2019).

[117] Ogunsola, O. O., Thacker, H. D., Bachim, B. L., Bakir, M. S., Pikarsky, J., Gaylord, T. K. & Meindl, J. D. Chip-level waveguide-mirror-pillar optical interconnect structure. *IEEE Photonics Technology Letters* **18**, 1672–1674 (2006).

[118] Lin, X., Hosseini, A., Dou, X., Subbaraman, H. & Chen, R. T. Low-cost board-to-board optical interconnects using molded polymer waveguide with 45 degree mirrors and inkjet-printed micro-lenses as proximity vertical coupler. *Optics Express* **21**, 60 (2013).

[119] Yu, S., Zuo, H., Sun, X., Liu, J., Gu, T. & Hu, J. Optical Free-Form Couplers for High-density Integrated Photonics (OFFCHIP): A Universal Optical Interface. *Journal of Lightwave Technology* **38**, 3358–3365 (2020).

[120] Almeida, V. R., Panepucci, R. R. & Lipson, M. Nanotaper for compact mode conversion. *Optics Letters* **28**, 1302–1304 (2003).

[121] Waldhäusl, R., Schnabel, B., Dannberg, P., Kley, E.-B., Bräuer, A. & Karthe, W. Efficient coupling into polymer waveguides by gratings. *Applied Optics* **36**, 9383–9390 (1997).

[122] Taillaert, D., Bogaerts, W., Bienstman, P., Krauss, T., Van Daele, P., Moerman, I., Verstuyft, S., De Mesel, K. & Baets, R. An out-of-plane grating coupler for efficient butt-coupling between compact planar waveguides and single-mode fibers. *IEEE Journal of Quantum Electronics* **38**, 949–955 (2002).

[123] Theurer, M., Moehrle, M., Sigmund, A., Velthaus, K.-O., Oldenbeuving, R. M., Wevers, L., Postma, F. M., Mateman, R., Schreuder, F., Geskus, D., Wörhoff, K., Dekker, R., Heideman, R. G. & Schell, M. Flip-Chip Integration of InP and SiN. *IEEE Photonics Technology Letters* **31**, 273–276 (2019).

[124] Op de Beeck, C., Haq, B., Elsinger, L., Gocalinska, A., Pelucchi, E., Corbett, B., Roelkens, G. & Kuyken, B. Heterogeneous III-V on silicon nitride amplifiers and lasers via microtransfer printing. *Optica* **7**, 386 (2020).

[125] Zhang, J., Haq, B., O'Callaghan, J., Gocalinska, A., Pelucchi, E., Trindade, A. J., Corbett, B., Morthier, G. & Roelkens, G. Transfer-printing-based integration of a III-V-on-silicon distributed feedback laser. *Optics Express* **26**, 8821–8830 (2018).

[126] De Groote, A., Cardile, P., Subramanian, A. Z., Fecioru, A. M., Bower, C., Delbeke, D., Baets, R. & Roelkens, G. Transfer-printing-based integration of single-mode waveguide-coupled III-V-on-silicon broadband light emitters. *Optics Express* **24**, 13754 (2016).

[127] Piels, M., Bauters, J. F., Davenport, M. L., Heck, M. J. R. & Bowers, J. E. Low-Loss Silicon Nitride AWG Demultiplexer Heterogeneously Integrated With Hybrid III–V/Silicon Photodetectors. *Journal of Lightwave Technology* **32**, 817–823 (2014).

[128] Shen, Y., Feng, S., Xie, X., Zang, J., Li, S., Su, T., Shang, K., Lai, W., Liu, G., Ben Yoo, S. J. & Campbell, J. C. Hybrid integration of modified uni-traveling carrier photodiodes on a multi-layer silicon nitride platform using total reflection mirrors. *Optics Express* **25**, 9521 (2017).

[129] Marom, E., Ramer, O. & Ruschin, S. Relation between normal-mode and coupled-mode analyses of parallel waveguides. *IEEE Journal of Quantum Electronics* **20**, 1311–1319 (1984).

[130] Blumenthal, D. J., Heideman, R., Geuzebroek, D., Leinse, A. & Roeloffzen, C. Silicon Nitride in Silicon Photonics. *Proceedings of the IEEE* **106**, 2209–2231 (2018).

[131] Rabiei, P., Steier, W., Zhang, C. & Dalton, L. Polymer micro-ring filters and modulators. *Journal of Lightwave Technology* **20**, 1968–1975 (2002).

[132] Lacraz, A., Polis, M., Theodosiou, A., Koutsides, C. & Kalli, K. Femtosecond Laser Inscribed Bragg Gratings in Low Loss CYTOP Polymer Optical Fiber. *IEEE Photonics Technology Letters* **27**, 693–696 (2015).

[133] Elshaari, A. W., Zadeh, I. E., Jöns, K. D. & Zwiller, V. Thermo-Optic Characterization of Silicon Nitride Resonators for Cryogenic Photonic Circuits. *IEEE Photonics Journal* **8**, 1–9 (2016).

[134] Xia, F., Sekaric, L. & Vlasov, Y. A. Mode conversion losses in silicon-on-insulator photonic wire based racetrack resonators. *Optics Express* **14**, 3872 (2006).

[135] Spillane, S. M., Kippenberg, T. J., Painter, O. J. & Vahala, K. J. Ideality in a Fiber-Taper-Coupled Microresonator System for Application to Cavity Quantum Electrodynamics. *Physical Review Letters* **91**, 043902 (2003).

[136] Saleh, B. E. A. & Teich, M. C. *Fundamentals of Photonics* (Wiley, New York, NY, 2007).

[137] Ramadan, T. A., Scarmozzino, R. & Osgood, R. M. Adiabatic couplers: design rules and optimization. *Journal of Lightwave Technology* **16**, 277–283 (1998).

[138] Bandyopadhyay, S., Sludds, A., Krastanov, S., Hamerly, R., Harris, N., Bunandar, D., Streshinsky, M., Hochberg, M. & Englund, D. Single chip photonic deep neural network with accelerated training. *arXiv:2208.01623* (2022).

[139] Xia, Q. & Yang, J. J. Memristive crossbar arrays for brain-inspired computing. *Nature Materials* **18**, 309–323 (2019).

[140] Wetzstein, G., Ozcan, A., Gigan, S., Fan, S., Englund, D., Soljačić, M., Denz, C., Miller, D. A. B. & Psaltis, D. Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39–47 (2020).

[141] Bernstein, L., Sludds, A., Panuski, C., Trajtenberg-Mills, S., Hamerly, R. & Englund, D. Single-Shot Optical Neural Network. *arXiv:2205.09103 [cs.ET]* (2022).

[142] Ashtiani, F., Geers, A. J. & Aflatouni, F. An on-chip photonic deep neural network for image classification. *Nature* **606**, 501–506 (2022).

[143] Wang, T., Ma, S.-Y., Wright, L. G., Onodera, T., Richard, B. C. & McMahon, P. L. An optical neural network using less than 1 photon per multiplication. *Nature Communications* **13**, 123 (2022).

[144] Liu, Z., Amini, A., Zhu, S., Karaman, S., Han, S. & Rus, D. L. Efficient and Robust LiDAR-Based End-to-End Navigation. 13247–13254 (2021).

[145] Messick, C., Blackburn, K., Brady, P., Brockill, P., Cannon, K., Cariou, R., Caudill, S., Chamberlin, S. J., Creighton, J. D., Everett, R., Hanna, C., Keppel, D., Lang, R. N., Li, T. G., Meacher, D., Nielsen, A., Pankow, C., Privitera, S., Qi, H., Sachdev, S., Sadeghian, L., Singer, L., Thomas, E. G., Wade, L., Wade, M., Weinstein, A. & Wiesner, K. Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data. *Physical Review D* **95**, 042001 (2017).

[146] Huerta, E. A., Allen, G., Andreoni, I., Antelis, J. M., Bachelet, E., Berriman, G. B., Bianco, F. B., Biswas, R., Carrasco Kind, M., Chard, K., Cho, M., Cowperthwaite, P. S., Etienne, Z. B., Fishbach, M., Forster, F., George, D., Gibbs, T., Graham, M., Gropp, W., Gruendl, R., Gupta, A., Haas, R., Habib, S., Jennings, E., Johnson, M. W. G., Katsavounidis, E., Katz, D. S., Khan, A., Kindratenko, V., Kramer, W. T. C., Liu, X., Mahabal, A., Marka, Z., McHenry, K., Miller, J. M., Moreno, C., Neubauer, M. S., Oberlin, S., Olivas, A. R., Petravick, D., Rebei, A., Rosofsky, S., Ruiz, M., Saxton, A., Schutz, B. F., Schwing, A., Seidel, E., Shapiro, S. L., Shen, H., Shen, Y., Singer, L. P., Sipocz, B. M., Sun, L., Towns, J., Tsokaros, A., Wei, W., Wells, J., Williams, T. J., Xiong, J. & Zhao, Z. Enabling real-time multi-messenger astrophysics discoveries with deep learning. *Nature Reviews Physics* **1**, 600–608 (2019).

[147] Coelho, C. N., Kuusela, A., Li, S., Zhuang, H., Ngadiuba, J., Aarrestad, T. K., Loncar, V., Pierini, M., Pol, A. A. & Summers, S. Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. *Nature Machine Intelligence* **3**, 675–686 (2021).

[148] Zibar, D., Piels, M., Jones, R. & Schaeffer, C. G. Machine Learning Techniques in Optical Communication. *Journal of Lightwave Technology* **34**, 1442–1452 (2016).

[149] Zhang, H., Gu, M., Jiang, X. D., Thompson, J., Cai, H., Paesani, S., Santagati, R., Laing, A., Zhang, Y., Yung, M. H., Shi, Y. Z., Muhammad, F. K., Lo, G. Q., Luo, X. S., Dong, B., Kwong, D. L., Kwek, L. C. & Liu, A. Q. An optical neural chip for implementing complex-valued neural network. *Nature Communications* **12**, 457 (2021).

[150] Jing, L., Shen, Y., Dubcek, T., Peurifoy, J., Skirlo, S., LeCun, Y., Tegmark, M. & Soljačić, M. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. vol. 70 of *Proceedings of Machine Learning Research*, 1733–1741 (PMLR, 2017).

[151] Tait, A. N., Nahmias, M. A., Shastri, B. J. & Prucnal, P. R. Broadcast and Weight: An Integrated Network For Scalable Photonic Spike Processing. *Journal of Lightwave Technology* **32**, 4029–4041 (2014).

[152] Ahmed, M. G., Huynh, T. N., Williams, C., Wang, Y., Shringarpure, R., Yousefi, R., Roman, J., Ophir, N. & Rylyakov, A. A 34Gbaud Linear Transimpedance Amplifier with Automatic Gain Control for 200Gb/s DP-16QAM Optical Coherent Receivers. In *2018 Optical Fiber Communications Conference and Exposition (OFC)* (2018).

[153] Konečný, J., McMahan, B. & Ramage, D. Federated Optimization: Distributed Optimization Beyond the Datacenter. *arXiv:1511.03575* (2015).

[154] Sludds, A., Bandyopadhyay, S., Chen, Z., Zhong, Z., Cochrane, J., Bernstein, L., Bunandar, D., Dixon, P. B., Hamilton, S. A., Streshinsky, M., Novack, A., Baehr-Jones, T., Hochberg, M., Ghobadi, M., Hamerly, R. & Englund, D. Delocalized photonic deep learning on the internet's edge. *Science* **378**, 270–276 (2022).

[155] Hughes, T. W., Minkov, M., Shi, Y. & Fan, S. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* **5**, 864 (2018).

[156] Pai, S., Sun, Z., Hughes, T. W., Park, T., Bartlett, B., Williamson, I. A. D., Minkov, M., Milanizadeh, M., Abebe, N., Morichetti, F., Melloni, A., Fan, S., Solgaard, O. & Miller, D. A. B. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science* **380**, 398–404 (2023).

[157] Zhang, H., Thompson, J., Gu, M., Jiang, X. D., Cai, H., Liu, P. Y., Shi, Y., Zhang, Y., Karim, M. F., Lo, G. Q., Luo, X., Dong, B., Kwek, L. C. & Liu, A. Q. Efficient On-Chip Training of Optical Neural Networks Using Genetic Algorithm. *ACS Photonics* **8**, 1662–1672 (2021).

[158] Cauwenberghs, G. A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization. In Hanson, S., Cowan, J. & Giles, C. (eds.) *Advances in Neural Information Processing Systems*, vol. 5 (Morgan-Kaufmann, 1992).

[159] Spall, J. C. An Overview of the Simultaneous Perturbation Method for Efficient Optimization. Tech. Rep. 19 (4), Johns Hopkins Applied Physics Laboratory (1998).

[160] Hillenbrand, J. M. (1995). URL `https://homepages.wmich.edu/~hillenbr/voweldata.html`.

[161] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G. & Wu, H. Mixed precision training. In *International Conference on Learning Representations* (2018).

[162] Camuto, A., Willetts, M., Simsekli, U., Roberts, S. J. & Holmes, C. C. Explicit Regularisation in Gaussian Noise Injections. In *Advances in Neural Information Processing Systems*, vol. 33, 16603–16614 (2020).

[163] Timurdogan, E., Sorace-Agaskar, C. M., Sun, J., Shah Hosseini, E., Biberman, A. & Watts, M. R. An ultralow power athermal silicon modulator. *Nature Communications* **5**, 4008 (2014).

[164] Strubell, E., Ganesh, A. & McCallum, A. Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 13693–13696 (2020).

[165] You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J. & Keutzer, K. ImageNet Training in Minutes. In *Proceedings of the 47th International Conference on Parallel Processing*, ICPP 2018 (Association for Computing Machinery, New York, NY, USA, 2018).

[166] Liu, X., Cheng, M., Zhang, H. & Hsieh, C.-J. Towards Robust Neural Networks via Random Self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018).

[167] Lopez-Pastor, V. & Marquardt, F. Self-learning Machines based on Hamiltonian Echo Backpropagation. *arXiv:2103.04992* (2021).

[168] Sun, J., Timurdogan, E., Yaacobi, A., Hosseini, E. S. & Watts, M. R. Large-scale nanophotonic phased array. *Nature* **493**, 195–199 (2013).

[169] Shu, H., Chang, L., Tao, Y., Shen, B., Xie, W., Jin, M., Netherton, A., Tao, Z., Zhang, X., Chen, R., Bai, B., Qin, J., Yu, S., Wang, X. & Bowers, J. E. Microcomb-driven silicon photonic systems. *Nature* **605**, 457–463 (2022).

[170] Nozaki, K., Matsuo, S., Fujii, T., Takeda, K., Shinya, A., Kuramochi, E. & Notomi, M. Femtofarad optoelectronic integration demonstrating energy-saving signal conversion and nonlinear functions. *Nature Photonics* **13**, 454–459 (2019).

[171] Li, G. H., Sekine, R., Nehra, R., Gray, R. M., Ledezma, L., Guo, Q. & Marandi, A. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics* (2022).

[172] Wang, C., Zhang, M., Chen, X., Bertrand, M., Shams-Ansari, A., Chandrasekhar, S., Winzer, P. & Lončar, M. Integrated lithium niobate electro-optic modulators operating at CMOS-compatible voltages. *Nature* **562**, 101–104 (2018).

[173] Mak, J. C. C., Xue, T., Yong, Z. & Poon, J. K. S. Wavelength Tunable Matched-Pair Vernier Multi-Ring Filters Using Derivative-Free Optimization Algorithms. *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1–12 (2020).

[174] Gyger, S., Zichi, J., Schweickert, L., Elshaari, A. W., Steinhauer, S., Covre da Silva, S. F., Rastelli, A., Zwiller, V., Jöns, K. D. & Errando-Herranz, C. Reconfigurable photonics with on-chip single-photon detectors. *Nature Communications* **12**, 1408 (2021).