

PHOTONIC COMPUTING ARCHITECTURES FOR CLASSICAL
AND QUANTUM INFORMATION PROCESSING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF APPLIED PHYSICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ben Bartlett
August 2022

© 2022 by Benjamin C Bartlett. All Rights Reserved.

Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.

<http://creativecommons.org/licenses/by-nc/3.0/us/>

This dissertation is online at: <https://purl.stanford.edu/ht812kd2819>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Shanhui Fan, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Amir Safavi-Naeini

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Olav Solgaard

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Abstract

As progress in traditional electronic computing systems approaches fundamental physical limits, we must explore alternative approaches for further growth in computing power. Photonics is a promising hardware platform for many emerging computing technologies, including optical neural networks and quantum computation. In this thesis, I will present several novel designs for light-based computing systems. First, I will discuss several advancements we have made in nanophotonic neural networks, including design and experimental realization of electro-optic nonlinear activation functions, and architectures and initialization routines for programmable linear optical devices. Next, I will present two novel schemes for quantum information processing: a programmable photonic gate array which can be dynamically reconfigured to prepare any quantum state, and an architecture for an optical quantum computer which can perform any calculation using only a single directly controllable qubit. Finally, I will discuss a design for a photonic quantum emulator capable of simulating the dynamics of a broad class of Hamiltonians in lattices with arbitrary dimensions and topologies.

Acknowledgments

The last five years at Stanford have been an exciting and enjoyable time for me, and I am deeply grateful for the support I have received throughout this journey.

First and foremost, I would like to thank my advisor, Shanhui Fan, for his continual support and guidance throughout my time at Stanford. Shanhui has both an encyclopedic knowledge of all kinds of physics and a remarkable ability to distill complex topics into something intuitive and readily understandable. I also appreciate that he has given me a great deal of intellectual freedom to explore my own (sometimes hare-brained) research ideas. Fortunately, a few of these ideas turned out alright. Shanhui cares deeply about his students' success, and he has taught me so much about how to be a creative scientist. It truly has been a privilege to have him as my advisor.

I have been fortunate to work with many brilliant grad students and postdocs throughout grad school. I am very grateful to Avik Dutt, who I have worked with closely on all things quantum photonics related. I mention MZIs 164 times in this thesis, and almost every instance has involved some sort of brainstorming with Sunil Pai. I'd also like to thank Dylan Black for many hours of watching me struggle to align lasers and for eventually always helping me out. For the first half of grad school, Tyler Hughes, Ian Williamson, and Momchil Minkov were all instrumental in solidifying my research direction in optical neural networks. Additionally, Olivia Long has been a great brainstorming companion during my last year at Stanford.

I am also indebted to several earlier mentors who helped put me on this life trajectory. My high school physics teacher, Mr. Eisele, really ignited my early interest in physics. And my undergraduate advisor, Prof. María Spiropulu, has opened so many doors for me and has been an amazing role model as physicist and as a mentor.

I would also like to thank all of my amazing friends who have supported me these last few years. [213] I am so lucky to have all of you in my life, and thank you for tolerating my long, often incoherent ramblings about physics.

I would like to thank my family for a lifetime of unwavering love and support. To my mom, Melissa, for spending a decade of her life homeschooling me and sparking an early interest in science; to my dad, Bob, for being the best example of a good work ethic; and to my brother, Jon, for always believing in me.

Finally, I owe so much of who I am to my partner and favorite person, Alex, who has been the θ to my ϕ for the last 9 years. From staying up with me through countless hours of late-night problem sets at Caltech, to listening to me nervously rehearse my PhD defense, he has been figuratively and literally by my side for every step of this journey.

Contents

Abstract	iv
Acknowledgments	v
1 Introduction	1
I Classical information processing using light	3
2 Optical neural networks	4
2.1 Introduction	4
2.2 Feedforward optical neural networks	6
2.3 Nonlinear activation function architecture	8
2.4 Performance and scalability	11
2.4.1 Power consumption	12
2.4.2 Latency	13
2.4.3 Physical footprint	14
2.4.4 Speed	15
2.5 Comparison with the Kerr effect	16
2.6 Machine learning tasks	18
2.6.1 Exclusive-OR Logic Function	18
2.6.2 Handwritten Digit Classification	20
2.7 Experimental realization of arbitrary electro-optic activation functions	23
2.7.1 Fabricated device	23
2.8 Experimental results	26
2.8.1 Direct controller	26
2.8.2 Lookup table controller	29
2.9 MNIST classification using experimental results	30
2.10 Conclusion	32

3	Optimization on universal linear optical devices	35
3.1	Introduction	35
3.2	Photonic Mesh	37
3.2.1	Photonic unitary implementation	37
3.2.2	Beamsplitter error tolerances	39
3.2.3	Phase shift tolerances	40
3.3	Haar Initialization	40
3.3.1	Cross state bias and sensitivity index	40
3.3.2	Phase shift distributions and Haar phase	41
3.3.3	Haar initialization	42
3.3.4	Tolerance dependence on N	43
3.4	Architecture Modifications	44
3.5	Simulations	46
3.5.1	Mesh initialization	46
3.5.2	Optimization problem and synthetic data	47
3.5.3	Training algorithm	48
3.5.4	Architecture comparison	48
3.6	Discussion	50
3.6.1	Haar initialization	50
3.6.2	Applications of mesh optimization	51
3.7	Conclusion	53
II	Quantum information processing using light	54
4	Photonic quantum programmable gate arrays	55
4.1	Introduction	55
4.2	Photonic quantum programmable gate arrays	57
4.2.1	Single-qubit operations	58
4.2.2	Two-photon gates	58
4.2.3	Fidelity and fault tolerance	64
4.3	Exact quantum state and operator preparation	66
4.3.1	Universality of the design	66
4.3.2	State preparation	66
4.3.3	Implementation of general quantum operators	67
4.4	Gradient-based circuit optimization	68
4.4.1	GHZ state preparation	70
4.4.2	Random state preparation	71

4.4.3	Quantum Fourier transform	72
4.4.4	Circuit compactness analysis	72
4.5	Conclusion	74
5	Photonic quantum computing	76
5.1	Introduction	76
5.2	Design	78
5.2.1	Rotation teleportation mechanism	79
5.2.2	Constructing arbitrary single-qubit gates	81
5.2.3	Two-photon gates	82
5.2.4	Arbitrary circuit compilation	83
5.2.5	Quantum state readout	84
5.3	Imperfection analysis	84
5.3.1	Deformation of photon pulses	85
5.3.2	Atomic spontaneous emission loss	85
5.3.3	Optical attenuation and insertion loss	87
5.4	Discussion	87
5.5	Conclusion	88
6	Quantum emulation	90
6.1	Introduction	90
6.2	Design	91
6.3	Device physics	91
6.4	Simulations and topological signatures	95
6.5	Error analysis	98
6.6	Conclusion	99
7	Conclusion and final remarks	100
III	Supplementary materials	102
A	Matrix optimization on programmable photonics	103
A.1	Derivation of beamsplitter errors	103
A.2	Derivation of the Haar measure	104
A.3	Unitary matrix bandsizes	107
A.4	Fabrication imperfections in a redundant mesh	108
A.5	Optical singular value decomposition simulations	109
A.6	Periodic parameters	109

A.7	Training simulation comparisons	112
A.8	An equivalent definition for α_{nl}	112
B	Quantum programmable gate arrays	116
B.1	Phase-modulated interference for photons with arbitrary spectra	116
B.2	Derivation of reflection coefficients	118
B.3	Implementations of common quantum gates	122
C	Quantum computing in a synthetic time dimension	124
C.1	Derivation of gate teleportation mechanism	124
C.2	Constructing arbitrary single-qubit rotations	127
C.3	Photonic qubit readout	130
C.4	Implementing a two-photon $c\sigma_z$ gate	131
C.5	Circuit compilation	132
C.6	Compiled example: quantum Fourier transform	133
C.7	Imperfection analysis	136
D	Programmable photonic quantum emulator	140
D.1	Deriving hopping coefficients between sites	140
D.1.1	Evaluating the commutator error terms	143
D.2	Deriving nonlinear on-site interaction potential	146
D.3	Simulation details	148
D.3.1	Tractable simulation with a Ponomarev state representation	148
D.4	Details for bandstructure computation	149
D.5	Dispersion analysis	150
E	Simulator for quantum networks and channels	151
E.1	Preliminaries	152
E.1.1	Qubits and quantum systems	152
E.1.2	Open quantum systems	153
E.1.3	Quantum gates	153
E.1.4	Quantum agents	155
E.1.5	Quantum networks and channels	155
E.2	Framework overview	156
E.2.1	Working with quantum information	157
E.2.2	Simulating quantum networks	159
E.3	Demonstrations	162
E.3.1	Quantum teleportation	162
E.3.2	Superdense coding	166

E.3.3	Man-in-the-middle attack	169
E.3.4	Quantum error correction	172
E.4	Conclusions	177
E.5	Appendix	177
E.5.1	Source code for simulator and demonstrations	177
E.5.2	Formal definition of a quantum agent	177
E.5.3	List of gates	178
F	Computing photon scattering in open quantum systems	180
F.1	Problem definition	180
F.2	Coarse-grained dynamics and the scattering operator	182
F.3	The temporal basis	182
F.4	Single waveguide: driven quantum two-level system	183
F.5	Computing photon scattering amplitudes	184
F.6	Total photon scattering probability	186
F.7	Computing second-order coherence in the scattered state	187
F.8	Pulse-wise second-order coherence	189
F.9	Multiple waveguides: spontaneous parametric downconversion	190
F.10	Two-photon scattering amplitudes	192
F.11	Multi-waveguide photon emission probability	193
	Bibliography	196

List of Tables

2.1	Summary of parameter values	11
2.2	Summary of per-layer optical neural network performance using the electro-optic activation function	12
2.3	Accuracy on the MNIST testing dataset after optimization. The phase gain, g_ϕ , of each layer was optimized during training.	23
A.1	Induction on x and y within each of the quadrants in the mesh.	114
A.2	Induction on x or y across each of the borders of $x, y = \lfloor \frac{N}{2} \rfloor$	115
B.1	A table of phase shifter parameters which implement various common single-qubit gates on the phase-modulated MZIs depicted in Figure 4.1c.	122
B.2	Construction of common multi-qubit gates by embedding single-qubit operations in a lattice of $c\sigma_z$ gates. Because the phase-modulated MZIs can implement any single-qubit operator, gate decompositions may be terminated with either with $c\sigma_z$ gates or with single-qubit gates, as the first layer of single-qubit operators of subsequent gates can implicitly include the final single-qubit operators of the previous logical gate. All quantum circuit diagrams in this paper were typeset using the <code>QCircuit</code> <code>L^AT_EX</code> package. [67]	123
E.1	A list of the built-in quantum operations included in SQUANCH as of version 1.1.0. All gates take at least one <code>Qubit</code> argument and return nothing, modifying the state of the associated <code>QSystem</code> in-place.	179

List of Figures

2.1	(a) Block diagram of a feedforward neural network of L layers. Each layer consists of a \hat{W}_i block representing a linear matrix which multiplies vector inputs x_{i-1} . The f_i block in each layer represents an element-wise nonlinear activation function operating on vectors z_i to produce outputs x_i . (b) Schematic of the optical interferometer mesh implementation of a single layer of the feedforward neural network. (c) Schematic of the proposed optical-to-optical activation function which achieves a nonlinear response by converting a small portion of the optical input, z into an electrical signal, and then intensity modulating the remaining portion of the original optical signal as it passes through an interferometer.	7
2.2	Activation function output amplitude (blue lines) and activation function transmission (green lines) as a function of input signal amplitude. The input and output are normalized to the phase gain parameter, g_ϕ . Panel pairs (a),(b) and (c),(d) correspond to a ReLU-like response, with a suppressed transmission for inputs with small amplitude and high transmission for inputs with large amplitude. Panel pairs (e),(f) and (g),(h) correspond to a clipped response, with high transmission for inputs with small amplitude and reduced transmission for inputs with larger amplitude.	10
2.3	Contours of constant activation threshold as a function of the optical-to-electrical gain and the modulator V_π of the activation function shown in Fig. 2.1(c) with a photodetector responsivity $\mathfrak{R} = 1.0$ A/W.	13
2.4	Nonlinear parameter Γ_{EO} for the electro-optic activation as a function of (a) gain, G , for $\alpha = 0.50, 0.10$, and 0.01 and (b) modulator $V_\pi L$. The nonlinear parameter associated with the optical Kerr effect, Γ_{Kerr} in a Silicon waveguide of cross sectional area $A = 0.05 \mu\text{m}^2$ corresponds to the black dotted line.	17

2.5	(a) Architecture of an L -layer ONN used to implement an N -input XOR logic function. (b) Red dots indicate the learned input-output relationship of the XOR for $N = 4$ on an 2-layer ONN. Electro-optic activation functions are configured with gain $g = 1.75\pi$ and biasing phase $\phi_b = \pi$. (c) Mean squared error (MSE) versus training epoch. (d) Final MSE after 5000 epochs averaged over 20 independent training runs vs activation function gain. Different lines correspond to the responses shown in Fig. 2.2, with $\phi_b = 1.00\pi, 0.85\pi, 0.00\pi,$ and 0.50π . Shaded regions correspond to the range (minimum and maximum) final MSE from the 20 training runs.	19
2.6	(a) Schematic of an optical image recognition setup based on an ONN. Images of handwritten numbers from the MNIST database are preprocessed by converting from real-space to k -space and selecting N Fourier coefficients associated with the smallest magnitude k -vectors. (b) Test accuracy (solid lines) and training accuracy (dashed lines) during training for a two layer ONN without activation functions (blue) and with activation functions (orange). $N = 16$ Fourier components were used as inputs to the ONN and each vector was normalized such that its L_2 norm is unity. The activation function parameters were $g_\phi = 0.05\pi$ and $\phi_b = 1.00\pi$. (c) Cross entropy loss during training. (d) Confusion matrix, specified in percentage, for the trained ONN with the electro-optic activation function.	21
2.7	(a) Schematic of the optical interferometer mesh implementation of a single layer of the feedforward neural network. (b) Schematic of the proposed optical-to-optical activation function circuit. The black and blue lines represent optical waveguides and electrical signal pathways respectively. (c) Optical image of the fabricated activation function circuit with Mach-Zehnder interferometer modulator.	24
2.8	Chip layout of the fabricated device: a 4×4 multi-layer optical neural network with multiple layers of nonlinear activation functions. Silicon nitride waveguides are shown in red, while electrical connections for the thermal phase shifters are shown in purple. Note that measured activation function transmissions in Section 2.8 used a smaller version of this chip with fewer layers.	25
2.9	The output of the MZI for various voltages applied to the phase shift for 0dBm input power to the directional coupler. The blue line shows the output power of the MZI in dBm, and the red line shows the corresponding normalized MZI output.	26
2.10	(a) A photo of the measurement setup, (b) the block diagram of the measurement setup with direct controller, (c) the block diagram of the measurement setup with the lookup table.	27

2.11	The normalized output power ($ f(z) ^2$) of nonlinear activation function circuit for various the normalized input power ($ z ^2$) at several bias point. In all of the figures, the normalized output power is relative to the maximum achievable output power of the circuit, which relates to minimum MZI attenuation.	28
2.12	(a) Target sigmoid function overlaid on the 2D power throughput map of the nonlinear activation function, (b) target modReLU function overlaid on the 2D power throughput map of the nonlinear activation function, (c) measurement result compared its target sigmoid function, (d) measurement result compared to its target modReLU function.	30
2.13	(a) Schematic of the optical neural network configuration for classifying handwritten digits from the MNIST dataset. The 28×28 pixel images are first Fourier transformed and cropped before being fed into the ONN. (b) Classification accuracy achieved for various ONN configurations. (c) Optical signal transmission through the neural network, averaged over all samples in the training dataset.	31
3.1	Mesh diagram representing the locally interacting rectangular mesh for $N = 8$. The inputs (and single-mode phase shifts at the inputs) are represented by blue triangles. Outputs are represented by purple squares. The MZI nodes are represented by red dots labelled with sensitivity index $\alpha_{n\ell}$ (e.g., $\alpha_{44} = 7$ is the most sensitive node). The nodes represent the Givens rotation U_n (in orange) at vertical layer ℓ (in green). Each photonic MZI node can be represented with 50:50 beamsplitters B (red) and phase shifters R_θ, R_ϕ (orange) with required ranges $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$	36
3.2	(a) The sensitivity index $\alpha_{n\ell}$ for $N = 64$. (b) Checkerboard plot for the average reflectivity $\langle r_{n\ell} \rangle$ in a rectangular mesh. (c) Haar-random matrix and run the decomposition in Ref. [50] to find phases approaching cross state in the middle of the mesh. (d) The Haar phase $\xi_{n\ell}$ for the rectangular mesh better displays the randomness. (e, f) Field measurements (absolute value) from propagation at input 32 in (e) Haar and (f) uniform random initialized rectangular meshes with $N = 64$	41
3.3	(a) Plot of the relationship between ξ_α and θ . (b) We show that phase shift standard deviation $\sigma_{\theta;\alpha}$ decreases as α increases. (c) A plot of $\sigma_{\theta;\alpha}$ as α increases. (d) The transmissivity of an MZI component as a function of a periodic Haar phase has a power law relationship. The periodic Haar phase $\tilde{\xi}_\alpha$ is mapped to the Haar phase by a function $\xi : \mathbb{R} \rightarrow [0, 1]$ as discussed in Appendix A.6.	43
3.4	(a) A 16×16 rectangular mesh (red). Extra tunable layers (green) may be added to significantly reduce convergence time. (b) A 16-input, 30-layer permuting rectangular mesh. The rectangular permutation layer is implemented using either waveguide crossings or cross state MZIs (gray).	45

3.5	Elementwise absolute values of unitary matrices resulting from rectangular ($U \sim \mathcal{U}_R$) and permuting rectangular ($U \sim \mathcal{U}_{PR}$) meshes where meshes are initialized with uniform-random phases.	46
3.6	A comparison of test error in <code>tensorflow</code> for $N = 128$ between rectangular (RM), permuting rectangular (PRM), and redundant rectangular (RRM) meshes for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256. Ideal = Haar random initialized $\theta_{n\ell}$ with $\epsilon = 0$. δN is the additional layers added in the redundant mesh. We stopped the $\delta N = 128$ run within 4000 iterations when it reached convergence within machine precision. Redundant meshes with 32 additional layers converge better than permuting rectangular meshes, and with just 16 additional layers, we get almost identical performance.	49
3.7	We implement six different optimizations for $N = 128$ where we vary the choice of permuting rectangular mesh (PRM) or rectangular mesh (RM); the initialization (random $\theta_{n\ell}$ or Haar-initialized $\theta_{n\ell}$); and photonic transmissivity error displacements ($\epsilon = 0$ or $\epsilon \sim \mathcal{N}(0, 0.01)$, where $\sigma_\epsilon^2 = 0.01$ is the variance of the beamsplitter errors). Conditions: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256, simulated in <code>tensorflow</code> . (a) Comparison of optimization performance (defaults are Haar initialization and $\epsilon_{n\ell} = 0$ unless otherwise indicated). Optimized error magnitude spatial map for (b) rectangular mesh shows higher off-diagonal errors and than (c) permuting rectangular. The optimized $\theta_{n\ell}$ phase shifts (see Appendix A.6) for (d) rectangular meshes are close to zero (cross state) near the center as opposed to (e) permuting rectangular meshes which have a striped pattern (likely due to initialization). NOTE: by $ \cdot $, we refer to the elementwise norm.	51

4.1	The architecture for the quantum programmable gate array shown at various levels of detail. (a) Physical layout of a four-qubit QPGA with a depth of four layers. Each logical qubit is path-encoded by a single photon in a pair of waveguides, with the parity of which waveguide represents $ 0\rangle$ and $ 1\rangle$ depending on the parity of the qubit index. (b) A quantum circuit diagram depicting the logical representation of the operator performed by the QPGA in the first panel. The “switch” symbols between two-qubit operations indicate that the connectivity of the gates can be reconfigured without changing the physical chip architecture. Solid control dots indicate $c\sigma_z$, while open dots indicate $\overline{c\sigma_z}$. (c) A single unit cell within the lattice. The ζ, ξ, θ, ϕ phase shifters are continuously variable trainable parameters, while $\eta = 0, \frac{\pi}{2}$ determines the connectivity of the $c\sigma_z$ gates between neighboring qubits. The pink dots represent quantum emitters embedded a distance a between two dichroic reflectors, depicted as blue and red rectangles, which selectively reflect light at frequencies ω and ω' , respectively. The delay lines are matched in length to ω' and terminate in reflectors. (d) Four-level energy structure of the quantum emitters embedded in the waveguides.	57
4.2	Brute-force state preparation algorithm to map $ 0\rangle^{\otimes n} \mapsto \psi\rangle$ using up to 2^n controlled rotations.	67
4.3	Fixed connectivity scheme employed in training. The $c\sigma_z$ operators in odd columns are implicitly constructed by embedding σ_x operations before and after physical $\overline{c\sigma_z}$ gates.	69
4.4	Optimization of a quantum circuit to prepare a four-qubit GHZ state. (Top) Evolution of the output state $ \tilde{\psi}\rangle$ over the course of training. The vertical axis represents the magnitude of the projection $\langle \tilde{\psi} b_j \rangle$ of the output state onto each computational basis state $ b_j\rangle$. (Bottom) Fidelity between the output state and target state over the course of training, reaching a maximum value of $\mathcal{F} \approx 99.94\%$. The shared horizontal axis indicates iterations during training.	71
4.5	Training a 20-layer QPGA to prepare an ensemble of randomly sampled four-qubit states. Fidelities between the output and target states are shown over the course of each optimization. The average fidelity at the end of training is $\overline{\mathcal{F}} = 99.2\%$	72

4.6	<p>Optimization of a 20-layer QPGA to prepare a quantum Fourier transform on the four input qubits. (Top) The operators \tilde{U}_i implemented by the QPGA after i training epochs. Each square array represents the magnitude (relative to the maximum element) and phase of the projection of the operator onto the lexicographically-ordered computational basis states, encoded in the respective size and hue of the squares. The final \tilde{U}_{50} is visually indistinguishable from \hat{U}. (Bottom) Fidelity between the implemented and target operator over the course of training. The final fidelity is $\mathcal{F} = 99.94\%$. An animated version of this figure showing the training of the implemented operator can be found in the supplementary materials.</p>	73
4.7	<p>Required circuit depths to implement a quantum Fourier transform for a range of qubit numbers using explicit decomposition (top solid line, blue) and using gradient-based decomposition (bottom solid line, orange) which achieves a fidelity above 99.9%. Relative compactness of explicit vs. gradient-based decompositions is depicted by the red dotted line. The approximate decompositions are significantly more compact than the explicitly constructed circuits.</p>	74
5.1	<p>The photonic quantum computer architecture described in this work. (a) The physical design of the device. Photonic qubits counter-propagate through a fiber storage ring and optical switches can selectively direct photons through a scattering unit to interact with an atom in a cavity which is coherently controlled by a laser. (b) The energy structure of the atom: Ω_1 is resonant with the cavity mode and photon carrier frequency, while Ω_0 is far-detuned. (c) Bloch sphere depiction of the state of a photonic qubit in the $\{ \odot\rangle, \oslash\rangle\}$ basis and an operation applied by one pass through the scattering unit. The rotations about \hat{z} by fixed angles (grey) are applied by the phase shifter and beamsplitter, while the rotation about \hat{y} by a controllable angle θ (solid red) is applied to the atom using the cavity laser. Projectively measuring the atom teleports this rotation onto the photon, but may overshoot the target angle θ by π (dotted red) depending on the measurement outcome m. This operation is a universal single-qubit primitive: by composing several of these operations and adapting subsequent rotation angles based on measurement outcomes, arbitrary single-qubit gates can be deterministically constructed. See Supplement 2 for a visualization of the gate mechanism.</p>	77
5.2	<p>Quantum gate sequence corresponding to one pass of a photon through the scattering unit. The projective measurement teleports the rotation applied to the atomic qubit onto the photonic qubit.</p>	81

5.3	Conceptual illustration of compiling a quantum circuit into an instruction sequence to be performed on the device. (a) A generic target quantum circuit. (b) Decomposition into an equivalent circuit of single-qubit and $c\sigma_z$ gates. (c) The circuit is further decomposed into a sequence of scattering interactions. This sequence can be assembled on a classical computer into an instruction set with six distinct primitives which correspond to physical actions. (d) The controllable elements of the quantum device are the optical switches, cavity laser, and atomic state readout.	83
5.4	(a) Output pulse shapes for $ g_0\rangle$ and $ g_1\rangle$ initialization when a cavity with cooperativity $C = 180$ is driven by a Gaussian input pulse. The inset highlights the behavior near maximum: the $ g_0\rangle$ output pulse is delayed and the $ g_1\rangle$ output has reduced amplitude. (b) Shape infidelity and photon leakage probability as a function of cavity cooperativity. Solid blue lines show the pulse shape infidelity when the reference pulse is delayed by $\Delta t_{01}/2$. (c) Estimated single-qubit circuit depth achievable while maintaining $> 50\%$ fidelity as a function of cavity cooperativity and photon attenuation per cycle, assuming one scattering interaction every cycle and no error correction. Dotted lines show various experimentally demonstrated cooperativity values in similar cavity systems. Lines 1-10 correspond respectively to Refs. [89], [55], [214], [152], [72], [161], [53], [137], [45], and [236].	86

6.1 Architecture for the programmable photonic quantum emulator described in this chapter. **(a)** The physical design of the device. Classical laser pulses or single-photon pulses propagate clockwise through a fiber storage ring. A programmable Mach-Zehnder interferometer connects the storage ring to a register loop which has an optical path length Δx equal to the length of a single time bin. By setting the phase shift values in the MZI, the hopping coefficients and phases κ_{mn}, α_{mn} can be programmatically adjusted. Photons have energy $\mu \equiv \hbar\omega_0$, and by using a $\chi^{(3)}$ -nonlinear fiber, a nonlinear interaction potential U can be emulated. **(b)** An example 2D grid lattice to be emulated by the device. Node labels correspond to photon pulse indices, and the device as shown in panel (a) is in the process of constructing the orange edge connecting nodes 1 and 2 with $(\kappa_{1,2}, \alpha_{1,2})$. **(c)** Illustration of a single clock cycle of the emulator constructing the interaction $(\kappa_{1,2}, \alpha_{1,2})$ in three steps. First, phase shifters are set to transfer photon 1 into the register. Second, photon 1 is interacted with photon 2 using $\theta = 2\kappa_{1,2}$ and $\phi = \alpha_{1,2}$. Third, the pulse (which now may contain a mixture of photons 1 and 2) is returned to its original time bin. **(d)** The evolution of the state of the device while emulating a tight-binding Hamiltonian over the lattice shown in panel (b). The bottom panel depicts the exact evolution of the target Hamiltonian over time, while the top panel shows the state of the emulator at each clock cycle, including register swaps and intermediate states between full iterations. A large value of $\kappa = 0.2$ was used for visual clarity, but more accurate results may be obtained by using smaller κ and running the emulation for a commensurately longer wall-clock time. 92

6.2	(a) Lattice diagram for a two-legged synthetic Hall ladder emulated with the device. By varying the inter-rung hopping phases α , an effective controllable magnetic field can be induced in the lattice. (b) Band structure of the system computed by diagonalizing the Hamiltonian for the exact case (top panels) and as emulated in the device (bottom) in the presence (left) and absence (right) of a synthetic magnetic field. Projection operators to the left and right nodes are color coded for each eigenstate. In all cases the Hamiltonian is represented in real space; for each eigenstate with eigenvalue E , we compute k with peak detection of its Fourier transform. This results in small numerical instabilities which are present in both the exact and emulated cases. Other parameters for this simulation: $\kappa = 0.1$, $\alpha = 2\pi/3$ or $\alpha = 0$, $\mu = U = 0$, number of lattice sites $D = 1000$, number of bosons $N = 1$. (c) Experimental signature for the propagation of chiral edge currents on the left leg of the ladder. A Gaussian input state is created with some initial $k = \pm 0.1$ by exciting multiple time bins with a phase difference between bins. When the gauge field is turned off ($\alpha = 0$), the pulses propagate in opposite directions, but when the field is turned on ($\alpha = 2\pi/3$), the motion in one direction is inhibited.	96
6.3	Emulated evolution of (a) a two-photon state and (b) a single-photon state in a 1D lattice as (c) time-dependent hopping phases are varied. The changing hopping phases introduce a changing gauge potential which causes the two-photon state to experience an effective electric field. The single-photon state is unaffected by this field.	97
6.4	Emulation of a tight-binding Hamiltonian over a four-dimensional tesseract. (a) Projection of the tesseract graph which defines the lattice. (b) Evolution of a two-photon state exhibiting oscillations between time bins $0 \leftrightarrow 10$ and $5 \leftrightarrow 15$. Parameters: $\kappa = 0.01$, $\alpha = \mu = U = 0$	98
A.1	Triangular mesh for $N = 8$ using (a) $2N - 3$ vertical layers ℓ showing the sensitivity index $\alpha_{n\ell}$ and (b) N diagonal layers m showing the transmissivity basis (t_n in red) and the measurement basis (x_n in purple).	105
A.2	Given $\eta = 0.001$, we compare bandsizes for rectangular ($U \sim \mathcal{U}_R(N, N)$), permuting rectangular ($U \sim \mathcal{U}_{PR}(N)$), and redundant meshes ($U \sim \mathcal{U}_R(N, 2N)$). Permuting rectangular meshes match the bandsize of Haar random matrices.	108
A.3	A comparison of test mean square error for $N = 128$ between redundant rectangular meshes with error ϵ for 256-layer mesh for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256, simulated in <code>tensorflow</code>	108

A.4	A comparison of test mean square error for $N = 64$ between SVD devices using rectangular (SVD-RM) and permuting rectangular (SVD-PRM) meshes for: 20000 iterations, Adam update, learning rate of 0.005, batch size of 128, simulated in <code>tensorflow</code> . Unless otherwise noted, the default setting is Haar random initialized θ_{nl} with $\sigma_\epsilon = 0$	110
A.5	Comparison of learned matrix errors and learned θ_{nl} weights after 20000 iterations for the Adam update at learning rate 0.0025 and batch size 256 for the simple unitary network. We consider two meshes: (1) rectangular mesh (RM), and (2) permuting rectangular mesh (PRM). We consider three conditions for each mesh: (1) ideal (with Haar random unitary initialization), (2) photonic beamsplitter error displacement $\epsilon \sim \mathcal{N}(0, 0.01)$, (3) random initialization.	111
A.6	Rectangular decomposition for even ($N = 8$) and odd ($N = 7$) meshes, showing the diagonal x, y basis. Values for α_{nl} are shown in red above each MZI, with values for $s_x[y]$ shown in blue below. The critical boundaries of $x, y = \lfloor \frac{N}{2} \rfloor$ separating the different quadrants are drawn in green. (Boundaries are offset for visual clarity.) . . .	114
B.1	Infidelities of the output state of a Mach-Zehnder interferometer for a range of spectral distributions. We assume an input wavefunction of $ \psi^{\text{in}}\rangle = \int d\omega g_{\delta t}(\omega) \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger(\omega) + \hat{a}_2^\dagger(\omega)) \emptyset\rangle$, where $g_\sigma(\omega)$ is a Gaussian with a spectral width of $\sigma = \frac{\delta\omega}{\omega_0}$ and a pulse length of $\delta t = \frac{1}{2\delta\omega}$ periods of the central frequency ω_0 . We compute the output wavefunction for an ensemble of 1000 values of ζ, ξ, θ, ϕ sampled uniformly from $[0, 2\pi)$ across 250 values of σ and plot the maximum, minimum, and average infidelity (defined as $1 - \mathcal{F}$) for each case, depicted as the bottom, middle, and top lines, respectively.	118
C.1	An annotated figure depicting the architecture described in the main text and the correspondence of physical and logical circuit elements. (a) The physical design of the device, with annotations indicating quantum operations implemented by physical circuit elements. (b) The energy structure of the atom: Ω_1 is resonant with the cavity mode and photon carrier frequency, while Ω_0 is far-detuned. (c) Gate diagram of the quantum circuit applied in a single pass of a photonic qubit through the scattering unit. The top rail denotes the state of the photonic qubit and the bottom rail denotes the atomic qubit. After the photon returns to the storage ring, $R_x(-\theta)$ is applied to the atomic qubit and a projective measurement of the atomic state is performed. The final output state $ \psi_{\text{out}}\rangle$ is $Z_{\frac{\pi}{4}} \sigma_z (-\sigma_y)^{m \oplus 1} R_y(\theta) Z_{\frac{\pi}{4}} \psi_{\text{in}}\rangle$, as described in Eq. 2 of the main text.	125

C.2	Construction of a SWAP gate from three scattering interactions. The top rail denotes the photonic qubit and the bottom rail denotes the atom. The $BZ_{\frac{\pi}{2}}B$ operations correspond to a return trip of the photon from the scattering site to the ring and back, passing through the beamsplitter and phase shifter twice.	131
C.3	Construction of a $c\sigma_z$ gate with three scattering interactions using a measurement-based approach. After measurement, the left and right circuits are equivalent. The single-qubit gates on either side of $c\sigma_z$ can be removed by absorbing them into the preceding/subsequent single-qubit gates as described above.	132
C.4	Graphical depiction of the circuit compilation process. (a) The target quantum circuit we wish to implement in the device, in this case a three-qubit quantum Fourier transform. (b) The first step of the compilation process is to decompose complex circuit elements into single-qubit and $c\sigma_z$ gates. The subcircuit depicted here implements the first controlled- $\phi_{\frac{\pi}{2}}$ gate between photonic qubits q_1 and q_2 . (c, d) The second step is to decompose each single-qubit gate (c) via Euler angles into a sequence of rotations which can be teleported from the atom to the photonic qubits, and to decompose each $c\sigma_z$ gate (d) using the scattering sequence shown in Figure C.3. (e, f) Programmatic representation of the instructions sent to the device to implement subroutines (c,d), respectively. The full code for implementing the target quantum circuit depicted in (a) is shown in Program C.1.	133
D.1	An example Fock state $ \psi\rangle = 0, 0, 2, 0, 1, 0, 0, 1\rangle$, which can be re-indexed by boson number as $ \psi'\rangle = 8, 5, 3, 3\rangle$. This maps to the Ponomarev basis state $ n_\psi\rangle = 112\rangle$	148
D.2	Construction of an annihilation operator in the bosonic lattice representation. Python pseudocode showing the process for constructing a QuTiP object from the Ponomarev indexing functions is included in the upper right.	149
D.3	Computing the k values for a lattice eigenstate using peak detection of the Fourier transform.	150
E.1	Circuit diagram for applying the single-qubit gate U to qubit q_k . Wires implicitly denote tensor products with identity.	154
E.2	Circuit diagram representing a quantum channel \mathcal{N} . Frequently, the coupling of the channel to the environment is assumed and not explicitly drawn, as in Figure E.11.	155

E.3	Schematic overview of the modules available in the SQUANCH framework, with an illustration in the lower left showing how quantum streams and states are represented. The <code>QSystem</code> is the most fundamental class, representing a multi-particle quantum state as a density matrix. Ensembles of quantum systems are efficiently handled by <code>QStreams</code> , and each <code>QSystem</code> has references to its constituent <code>Qubits</code> . Functions in the <code>Gates</code> module can be used to manipulate the state of a quantum system. <code>Agents</code> are generalized quantum-mechanical “actors” which are initialized from a <code>QStream</code> instance and can alter the state of the quantum systems in their stream object, typically by interacting directly with <code>Qubits</code> . <code>Agents</code> run in parallel from separate processes and are connected by quantum and classical <code>Channels</code> , which apply customizable <code>Error</code> models to the transmitted information and synchronize agent clocks.	156
E.4	Circuit implementation of a quantum Fourier transform. ϕ_m denotes the phase gate $\begin{pmatrix} 1 & 0 & \omega_m \end{pmatrix}$, where $\omega_m \equiv e^{\frac{2\pi i}{2^m}}$ is a primitive root of unity.	158
E.5	Circuit diagram for the two-party quantum teleportation simulation described above.	163
E.6	Observed and expected fractional populations of measurements resulting in $ \psi\rangle \rightarrow 1\rangle$ for the two-party quantum teleportation simulation with an ensemble size of 250.	165
E.7	Circuit diagram for the three-party quantum superdense coding experiment described above.	166
E.8	Results of Program E.6, showing Alice’s original image (left), and Bob’s received image (right). Bob applies no correction for attenuation errors, replacing dropped qubits with bit pairs of (0,0).	169
E.9	Circuit diagram demonstrating a naive interception attack on the superdense networking protocol demonstrated in Figure E.7. Eve’s meter represents measurement of a qubit and transmission of a new qubit in the observed state.	169
E.10	The results of the simulated interception attack simulated by Program E.7. Since Eve only has access to half of the entangled $ AB\rangle$ state, she recovers only random noise, and Bob’s image is half-corrupted, alerting him to the presence of an eavesdropper.	171
E.11	Circuit diagram of encoding and decoding qubits using the Shor code. E represents a quantum channel with an error model which can corrupt a single physical qubit by applying a random unitary operator.	172
E.12	A screenshot of the output of Program E.9 running in a Jupyter notebook. The original message can be recovered if a quantum error correcting protocol is used. Progress bars are automatically updated from <code>Agent.qstream.__iter__</code> when running programs with <code>Simulation(...).run()</code> and are shown in notebook or terminal environments.	177

F.1 The general problem is to compute the field scattered into unidirectional waveguide(s) from an energy-nonconserving system Hamiltonian with a unique ground state. This class of Hamiltonian is often used to represent coherent laser pulses scattering off quantum-optical systems such as a two-level system, Jaynes-Cummings system, or entangled photon pair source. 181

Chapter 1

Introduction

Technological progress in this century has been inextricably linked to the continual advancement of computing power. For the last five decades, computing power has roughly doubled every two years, largely enabled by shrinking transistor sizes. This sustained growth in compute capacity has enabled incredible technological developments, and today computers routinely play complex games, generate art from written descriptions, and drive cars.

However, this progress is unsustainable, and traditional electronic computing systems are approaching fundamental physical limits. Transistors are now shrinking to sizes comparable to silicon atomic lattice spacings. If we continue to shrink transistors further, quantum tunneling will become an exponentially dominating source of errors. Additionally, clock frequencies in processors have stagnated in the last decade due to thermal constraints, which are primarily a function of switching energy. This, too, will eventually hit a fundamental limit, as there is a thermodynamic cost for the change in entropy of irreversibly processing one bit of information. [129]

As progress in traditional electronic computing systems approaches these fundamental physical limits, we will need to explore alternative approaches for further growth in computing power.

Photonics is a promising hardware platform for many emerging computing technologies, including neural network accelerators and quantum information processing. The physics of light is fundamentally linear, which makes photonics ideal for performing linear algebra tasks for machine learning applications. Additionally, optical computing systems can perform unitary, fully reversible computations which preserve entropy, so in principle they can be done for no energy cost while producing no heat. Optics is also very appealing for quantum computers, as photonic qubits have very long coherence times, can operate at room temperature, and are optimal information carriers between distant nodes in a quantum network.

In this thesis, we will primarily explore two emerging technologies which use light to process information: optical neural networks and quantum computers.

In Part I, we present several advancements we have made for optical neural networks, which use

lattices of programmable interferometers to physically process optically encoded information. Chapter 2 theoretically describes and experimentally demonstrates an electro-optic nonlinear activation function for optical neural networks. Chapter 3 presents error-tolerant initialization routines and architectures for the programmable multi-mode interferometers which can be combined to form an optical neural network.

In Part II, we present three novel schemes for quantum computation and emulation. Chapter 4 details a design for a quantum programmable gate array – an integrated photonic circuit which can be dynamically reconfigured without hardware modifications to prepare any quantum state or operator. Chapter 5 presents an architecture for a photonic quantum computer which can perform any computation using only a single coherently controlled atomic qubit. Finally, Chapter 6 describes a design for a programmable quantum emulator capable of simulating the dynamics of a broad class of quantum systems in arbitrary dimensions and topologies.

Part I

Classical information processing using light

Chapter 2

Optical neural networks

In this Chapter, we introduce an electro-optic hardware platform for nonlinear activation functions in optical neural networks (ONNs) [257]. The optical-to-optical nonlinearity operates by converting a small portion of the input optical signal into an analog electric signal, which is used to modulate the original optical signal with no reduction in processing speed. Our scheme allows for complete nonlinear on-off contrast in transmission at relatively low optical power thresholds and eliminates the requirement of having additional optical sources between each layer of the network. Moreover, the activation function is reconfigurable via electrical bias, allowing it to be programmed or trained to synthesize a variety of nonlinear responses. This activation function significantly improves the performance of optical neural networks on a variety of machine learning tasks.

Additionally, we experimentally demonstrate this electro-optic activation function using a fabricated silicon-nitride chip [193]. Electrical signal processing allows the integrated photonic circuit to realize any optical-to-optical nonlinearity that does not require amplification. The demonstrated activation circuit allows for the realization of arbitrary nonlinearities with far lower optical signal attenuation, paving the way for much deeper ONNs.

2.1 Introduction

In recent years, there has been significant interest in alternative computing platforms specialized for high performance and efficiency on machine learning tasks. GPUs have demonstrated peak performance with trillions of floating point operations per second when performing matrix multiplication, which is several orders of magnitude larger than general-purpose digital processors such as CPUs [179]. However, dedicated matrix multiplication hardware promises even further computational advantages, and analog computing has been explored for this task because it is not limited by the bottlenecks of sequential instruction execution and memory access [219, 220, 51, 46, 52].

Optical hardware platforms are particularly appealing for computing and signal processing due

to their ultra-large signal bandwidths, low latencies, and reconfigurability [40, 148, 76]. They have also gathered significant interest in machine learning applications, such as artificial neural networks (ANNs). Nearly three decades ago, the first optical neural networks (ONNs) were proposed based on free-space optical lens and holography setups [3, 196]. More recently, ONNs have been implemented in chip-integrated photonic platforms [226] using programmable waveguide interferometer meshes which perform matrix-vector multiplications [157]. In theory, the performance of such systems is competitive with digital computing platforms because they may perform matrix-vector multiplications in constant time with respect to the matrix dimension. In contrast, matrix-vector multiplication has a quadratic time complexity on a digital processor. Other approaches to performing matrix-vector multiplications in chip-integrated ONNs, such as microring weight banks and photodiodes, have also been proposed [237].

Nonlinear activation functions play a key role in ANNs by enabling them to learn complex mappings between their inputs and outputs. Whereas digital processors have the expressiveness to trivially apply nonlinearities such as the widely-used `sigmoid`, `ReLU`, and `tanh` functions, the realization of nonlinearities in optical hardware platforms is more challenging. One reason for this is that optical nonlinearities are relatively weak, necessitating a combination of large interaction lengths and high signal powers, which impose lower bounds on the physical footprint and the energy consumption, respectively. Although it is possible to resonantly enhance optical nonlinearities, this comes with an unavoidable trade-off in reducing the operating bandwidth, thereby limiting the information processing capacity of an ONN. Additionally, maintaining uniform resonant responses across many elements of an optical circuit necessitates additional control circuitry for calibrating each element [198].

A more fundamental limitation of optical nonlinearities is that their responses tend to be fixed during device fabrication. This limited tunability of the nonlinear optical response prevents an ONN from being reprogrammed to realize different forms of nonlinear activation functions, which may be important for tailoring ONNs for different machine learning tasks. Similarly, a fixed nonlinear response may also limit the performance of very deep ONNs with many layers of activation functions since the optical signal power drops below the activation threshold, where nonlinearity is strongest, in later layers due to loss in previous layers. For example, with optical saturable absorption from 2D materials in waveguides, the activation threshold is on the order of 1-10 mW [17, 180, 105], meaning that the strength of the nonlinearity in each subsequent layer will be successively weaker as the transmitted power falls below the threshold.

In light of these challenges, the ONN demonstrated in Ref. [226] implemented its activation functions by detecting each optical signal, feeding them through a conventional digital computer to apply the nonlinearity, and then modulating new optical signals for the subsequent layer. Although this approach benefits from the flexibility of digital signal processing, conventional processors have a limited number of input and output channels, which make it challenging to scale this approach

to very large matrix dimensions, which corresponds to a large number of optical inputs. Moreover, digitally applied nonlinearities add latency from the analog-to-digital conversion process and constrain the computational speed of the neural network to the same GHz-scale clock rates which ONNs seek to overcome. Thus, a hardware nonlinear optical activation, which doesn't require repeated bidirectional optical-electronic signal conversion, is of fundamental interest for making integrated ONNs a viable machine learning platform.

In this chapter, we propose an electro-optic architecture for synthesizing optical-to-optical nonlinearities which alleviates the issues discussed above. Our architecture features complete *on-off* contrast in signal transmission, a variety of nonlinear response curves, and a low activation threshold. Rather than using traditional optical nonlinearities, our scheme operates by measuring a small portion of the incoming optical signal power and using electro-optic modulators to modulate the original optical signal, without any reduction in operating bandwidth or computational speed. Additionally, our scheme allows for the possibility of performing additional nonlinear transformations on the signal using analog electrical components. Related electro-optical architectures for generating optical nonlinearities have been previously considered [136, 145, 238]. In this chapter, we focus on the application of our architecture as an element-wise activation in a feedforward ONN, but the synthesis of low-threshold optical nonlinearities could be of broader interest to optical computing and information processing.

2.2 Feedforward optical neural networks

In this section, we briefly review the basics of feedforward artificial neural networks (ANNs) and describe their implementation in a reconfigurable optical circuit, as proposed in Ref. [226]. As outlined in Fig. 2.1(a), an ANN is a function which accepts an input vector, x_0 and returns an output vector, x_L . This is accomplished in a layer-by-layer fashion, with each layer consisting of a linear matrix-vector multiplication followed by the application of an element-wise nonlinear function, or *activation*, on the result. For a layer with index i , containing a weight matrix \hat{W}_i and activation function $f_i(\cdot)$, its operation is described mathematically as

$$x_i = f_i\left(\hat{W}_i \cdot x_{i-1}\right) \quad (2.1)$$

for i from 1 to L .

Before they are able to perform a given machine learning task, ANNs must be trained. The training process is typically accomplished by minimizing the prediction error of the ANN on a set of training examples, which come in the form of input and target output pairs. For a given ANN, a loss function is defined to quantify the difference between the target output and output predicted by the network. During training, this loss function is minimized with respect to tunable degrees of freedom, namely the elements of the weight matrix \hat{W}_i within each layer. In general, although less

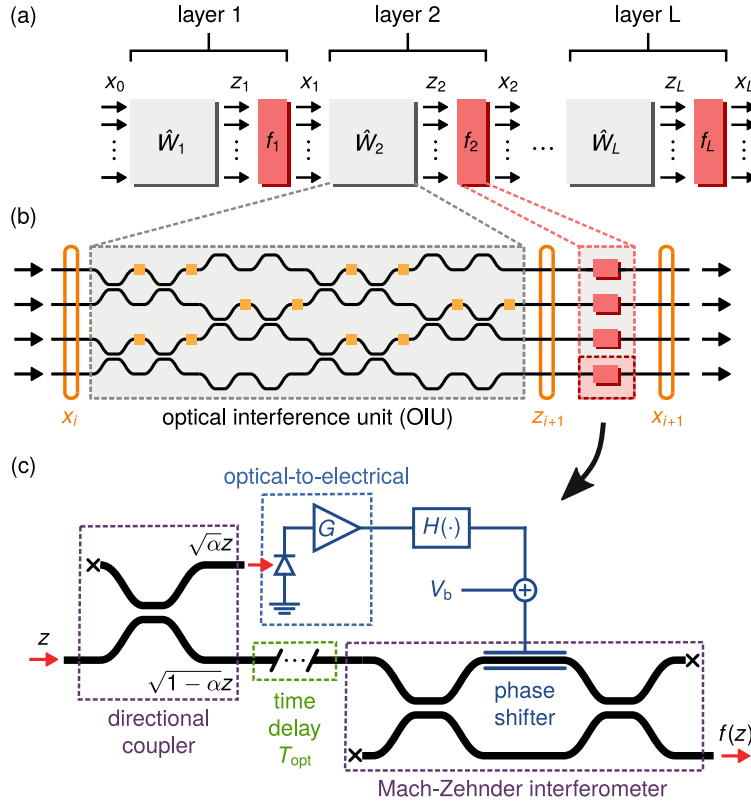


Figure 2.1: (a) Block diagram of a feedforward neural network of L layers. Each layer consists of a \hat{W}_i block representing a linear matrix which multiplies vector inputs x_{i-1} . The f_i block in each layer represents an element-wise nonlinear activation function operating on vectors z_i to produce outputs x_i . (b) Schematic of the optical interferometer mesh implementation of a single layer of the feedforward neural network. (c) Schematic of the proposed optical-to-optical activation function which achieves a nonlinear response by converting a small portion of the optical input, z into an electrical signal, and then intensity modulating the remaining portion of the original optical signal as it passes through an interferometer.

common, it is also possible to train the parameters of the activation functions [246].

Optical hardware implementations of ANNs have been proposed in various forms over the past few decades. In this chapter, we focus on a recent demonstration in which the linear operations are implemented using an integrated optical circuit [226]. In this scheme, the information being processed by the network, x_i , is encoded into the modal amplitudes of the waveguides feeding the device and the matrix-vector multiplications are accomplished using meshes of integrated optical interferometers. In this case, training the network requires finding the optimal settings for the integrated optical phase shifters controlling the interferometers, which may be found using an analytical model of the chip, or using *in-situ* backpropagation techniques [101].

In the next section, we present an approach for realizing the activation function, $f_i(\cdot)$, on-chip

with a hybrid electro-optic circuit feeding an inteferometer. In Fig. 2.1(b), we show how this activation scheme fits into a single layer of an ONN and show the specific form of the activation in Fig. 2.1(c). We also give the specific mathematical form of this activation and analyze its performance in practical operation.

2.3 Nonlinear activation function architecture

In this section, we describe our proposed nonlinear activation function architecture for optical neural networks, which implements an optical-to-optical nonlinearity by converting a small portion of the optical input power into an electrical voltage. The remaining portion of the original optical signal is phase- and amplitude-modulated by this voltage as it passes through an interferometer. For an input signal with amplitude z , the resulting nonlinear optical activation function, $f(z)$, is a result of the responses of the interferometer under modulation as well as the components in the electrical signal pathway.

A schematic of the architecture is shown in Fig. 2.1(c), where black and blue lines represent optical waveguides and electrical signal pathways, respectively. The input signal first enters a directional coupler which routes a portion, α , of the input optical power to a photodetector. The photodetector is the first element of an optical-to-electrical conversion circuit, which is a standard component of high-speed optical receivers for converting an optical intensity into a voltage. Here we assume a normalization of the optical signal such that the total power in the input signal is given by $|z|^2$. The optical-to-electrical conversion process consists of the photodetector producing an electrical current, $I_{\text{pd}} = \mathfrak{R} \cdot \alpha |z|^2$, where \mathfrak{R} is the photodetector responsivity, and a transimpedance amplifying stage, characterized by a gain G , converting this current into a voltage $V_G = G \cdot \mathfrak{R} \cdot \alpha |z|^2$. The output voltage of the optical-to-electrical conversion circuit then passes through a nonlinear signal conditioner with a transfer function, $H(\cdot)$. This component allows for the application of additional nonlinear functions to transform the voltage signal. Finally, the conditioned voltage signal, $H(V_G)$ is combined with a static bias voltage, V_b to induce a phase shift of

$$\Delta\phi = \frac{\pi}{V_\pi} [V_b + H(G\mathfrak{R}\alpha|z|^2)] \quad (2.2)$$

for the optical signal routed through the lower port of the directional coupler. The parameter V_π represents the voltage required to induce a phase shift of π in the phase modulator. This phase shift, defined by Eq. 2.2, is a nonlinear self-phase modulation because it depends on the input signal intensity.

An optical delay line between the directional coupler and the Mach-Zehnder interferometer (MZI) is used to match the signal propagation delays in the optical and electrical pathways. This ensures that the nonlinear self-phase modulation defined by Eq. 2.2 is applied at the same time that the optical signal which generated it passes through the phase modulator. For the circuit shown in Fig.

2.1(c), the optical delay is $\tau_{\text{opt}} = \tau_{\text{oe}} + \tau_{\text{nl}} + \tau_{\text{rc}}$, accounting for the contributions from the group delay of the optical-to-electrical conversion stage (τ_{oe}), the delay associated with the nonlinear signal conditioner (τ_{nl}), and the RC time constant of the phase modulator (τ_{rc}).

The nonlinear self-phase modulation achieved by the electric circuit is converted into a nonlinear amplitude response by the MZI, which has a transmission depending on $\Delta\phi$ as

$$t_{\text{MZI}} = j \exp\left(-j\frac{\Delta\phi}{2}\right) \cos\left(\frac{\Delta\phi}{2}\right). \quad (2.3)$$

Depending on the configuration of the bias, V_b , a larger input optical signal amplitude causes either more or less power to be diverted away from the output port, resulting in a nonlinear self-intensity modulation. Combining the expression for the nonlinear self-phase modulation, given by Eq. 2.2, with the MZI transmission, given by Eq. 2.3, the mathematical form of the activation function can be written explicitly as

$$f(z) = j\sqrt{1-\alpha} \exp\left(-j\frac{1}{2}\left[\phi_b + \pi\frac{H(G\Re\alpha|z|^2)}{V_\pi}\right]\right) \cos\left(\frac{1}{2}\left[\phi_b + \pi\frac{H(G\Re\alpha|z|^2)}{V_\pi}\right]\right) z, \quad (2.4)$$

where the contribution to the phase shift from the bias voltage is

$$\phi_b = \pi\frac{V_b}{V_\pi}. \quad (2.5)$$

For the remainder of this chapter, we focus on the case where no nonlinear signal conditioning is applied to the electrical signal pathway, i.e. $H(V_G) = V_G$. However, even with this simplification the activation function still exhibits a highly nonlinear response. We also neglect saturating effects in the OE conversion stage which can occur in either the photodetector or the amplifier. However, in practice, the nonlinear optical-to-optical transfer function could take advantage of these saturating effects.

With the above simplifications, a more compact expression for the activation function response is

$$f(z) = j\sqrt{1-\alpha} \exp\left(-j\left[\frac{g_\phi|z|^2}{2} + \frac{\phi_b}{2}\right]\right) \cos\left(\frac{g_\phi|z|^2}{2} + \frac{\phi_b}{2}\right) z, \quad (2.6)$$

where the phase gain parameter is defined as

$$g_\phi = \pi\frac{\alpha G\Re}{V_\pi}. \quad (2.7)$$

Equation 2.7 indicates that the amount of phase shift per unit input signal power can be increased via the gain and photodiode responsivity, or by converting a larger fraction of the optical power to the electrical domain. However, tapping out a larger fraction optical power also results in a larger linear loss, which is undesirable.

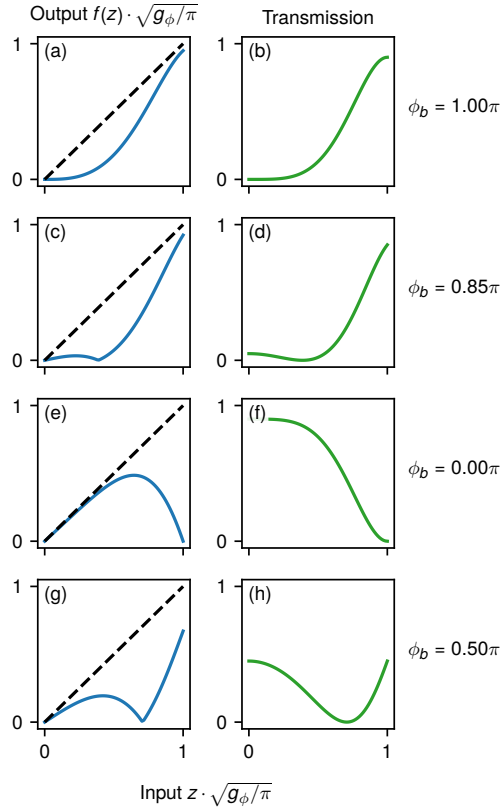


Figure 2.2: Activation function output amplitude (blue lines) and activation function transmission (green lines) as a function of input signal amplitude. The input and output are normalized to the phase gain parameter, g_ϕ . Panel pairs (a),(b) and (c),(d) correspond to a ReLU-like response, with a suppressed transmission for inputs with small amplitude and high transmission for inputs with large amplitude. Panel pairs (e),(f) and (g),(h) correspond to a clipped response, with high transmission for inputs with small amplitude and reduced transmission for inputs with larger amplitude.

The electrical biasing of the activation phase shifter, represented by V_b , is an important degree of freedom for determining its nonlinear response. We consider a representative selection, consisting of four different responses, in Fig. 2.2. The left column of Fig. 2.2 plots the output signal amplitude as a function of the input signal amplitude i.e. $|f(z)|$ in Eq. 2.6, while the right column plots the transmission coefficient i.e. $|f(z)|^2/|z|^2$, a quantity which is more commonly used in optics than machine learning. The first two rows of Fig. 2.2, corresponding to $\phi_b = 1.00\pi$ and 0.85π , exhibit a response which is comparable to the ReLU activation function: transmission is low for small input values and high for large input values. For the bias of $\phi_b = 0.85\pi$, transmission at low input values is slightly increased with respect to the response where $\phi_b = 1.00\pi$. Unlike the ideal ReLU response, the activation at $\phi_b = 0.85\pi$ is not entirely monotonic because transmission first goes to zero before increasing. On the other hand, the responses shown in the bottom two rows of Fig.

Parameter	Value
Modulator and detector rate	10 GHz
Photodetector responsivity (\mathfrak{R})	1 A/W
Optical-to-electrical circuit power consumption	100 mW
Optical-to-electrical circuit group delay (τ_{eo})	100 ps
Phase modulator RC delay (τ_{rc})	20 ps
Mesh MZI length (D_{MZI})	100 μm
Mesh MZI height (H_{MZI})	60 μm
Waveguide effective index (n_{eff})	3.5

Table 2.1: Summary of parameter values

2.2, corresponding to $\phi_b = 0.0\pi$ and 0.50π , are quite different. These configurations demonstrate a saturating response in which the output is suppressed for higher input values but enhanced for lower input values. For all of the responses shown in Fig. 2.2, we have assumed $\alpha = 0.1$ which limits the maximum transmission to $1 - \alpha = 0.9$.

A benefit of having electrical control over the activation response is that, in principle, its electrical bias can be connected to the same control circuitry which programs the linear interferometer meshes. In doing so, a single ONN hardware unit can then be reprogrammed to synthesize many different activation function responses. This opens up the possibility of heuristically selecting an activation function response, or directly optimizing the the activation bias using a training algorithm. This realization of a flexible optical-to-optical nonlinearity can allow ONNs to be applied to much broader classes of machine learning tasks.

We note that Fig. 2.2 shows only the amplitude response of the activation function. In fact, all of these responses also introduce a nonlinear self-phase modulation to the output signal. If desired, this nonlinear self-phase modulation can be suppressed using a push-pull interferometer configuration in which the generated phase shift, $\Delta\phi$, is divided and applied with opposite sign to the top and bottom arms.

2.4 Performance and scalability

In this section, we discuss the performance of an integrated ONN which uses meshes of integrated optical interferometers to perform matrix-vector multiplications and the electro-optic activation function, as shown in Fig. 2.1(b),(c). Here, we focus on characterizing how the power consumption, computational latency, physical footprint, and computational speed of the ONN scale with respect to the number of network layers, L and the dimension of the input vector, N , assuming square matrices. The system parameters used for this analysis are summarized in Table 2.1 and the figures of merit are summarized in Table 2.2.

	Scaling		Per-layer figures of merit		
	Mesh	Activation	$N = 4$	$N = 10$	$N = 100$
Power consumption*		LN	0.4 W	1 W	10 W
Latency	LN	L	125 ps	132 ps	237 ps
Footprint	LN^2	LN	2.5 mm ²	6.6 mm ²	120.0 mm ²
Speed	LN^2		1.6×10^{11} MAC/s	1×10^{12} MAC/s	1×10^{14} MAC/s
Efficiency*		N^{-1}	2.5 pJ/MAC	1 pJ/MAC	100 fJ/MAC

*Assuming no power consumption in the interferometer mesh phase shifters

Table 2.2: Summary of per-layer optical neural network performance using the electro-optic activation function

2.4.1 Power consumption

The power consumption of the ONN, as shown in Fig. 2.1(b), consists of contributions from (1) the programmable phase shifters inside the interferometer mesh, (2) the optical source supplying the input vectors, x_0 , and (3) the active components of the activation function such as the amplifier and photodetector. In principle, the contribution from (1) can be made negligible by using phase change materials or ultra-low power MEMS phase shifters. Therefore, in this section we focus only on contributions (2) and (3) which pertain to the activation function.

To quantify the power consumption, we first consider the minimum input optical power to a *single* activation that triggers a nonlinear response. We refer to this as the activation function threshold, which is mathematically defined as

$$P_{\text{th}} = \frac{\Delta\phi|_{\delta T=0.5}}{g_\phi} = \frac{V_\pi}{\pi\alpha G\mathfrak{R}} \cdot \Delta\phi|_{\delta T=0.5}, \quad (2.8)$$

where $\Delta\phi|_{\delta T=0.5}$ is the phase shift necessary to generate a 50% change in the power transmission with respect to the transmission with null input for a given ϕ_b . This threshold corresponds to $z\sqrt{g_\phi/\pi} = 0.73$ in Fig. 2.2(b), to $z\sqrt{g_\phi/\pi} = 0.85$ in Fig. 2.2(d), to $z\sqrt{g_\phi/\pi} = 0.73$ in Fig. 2.2(f), and to $z\sqrt{g_\phi/\pi} = 0.70$ in Fig. 2.2(h). In general, a lower activation threshold will result in a lower optical power required at the ONN input, $|x_0|^2$. According to Eq. 2.8, the activation threshold can be reduced via a small V_π and a large optical-to-electrical conversion gain, $G\mathfrak{R} \sim 1.0$ V/mW. The relationship between G and V_π for activation thresholds of 0.1 mW, 1.0 mW, and 10.0 mW is shown in Fig. 2.3 for a fixed $\mathfrak{R} = 1$ A/W. Additionally, in Fig. 2.3 we conservatively assume $\phi_b = \pi$ which has the highest threshold of the activation function biases shown in Fig. 2.2.

If we take the lowest activation threshold of 0.1 mW in Fig. 2.3, the optical source to the ONN would then need to supply $N \cdot 0.1$ mW of optical power. The power consumption of integrated optical receiver amplifiers varies considerably, ranging from as low as 10 mW to as high as 150 mW [6, 218, 169], depending on a variety of factors which are beyond the scope of this article. Therefore,

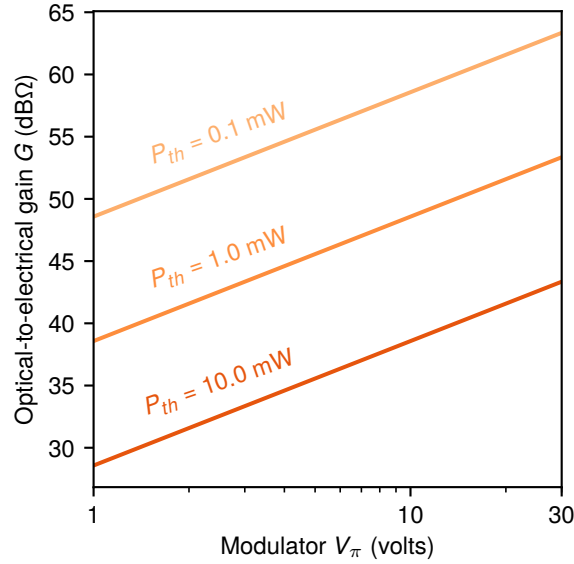


Figure 2.3: Contours of constant activation threshold as a function of the optical-to-electrical gain and the modulator V_π of the activation function shown in Fig. 2.1(c) with a photodetector responsivity $\mathfrak{R} = 1.0$ A/W.

a conservative estimate of the power consumption from the optical-to-electrical conversion circuits in all activations is $L \cdot N \cdot 100$ mW. For an ONN with $N = 100$, the power consumption per layer from the activation function would be 10 W and would require a total optical input power of $N \cdot P_{th} = 100 \cdot 0.1$ mW = 10 mW. Thus, the total power consumption of the ONN is dominated by the activation function electronics.

2.4.2 Latency

For the feedforward neural network architecture shown in Fig. 2.1(a), the latency is defined by the elapsed time between supplying an input vector, x_0 and detecting its corresponding prediction vector, x_L . In an integrated ONN, as implemented in Fig. 2.1(b), this delay is simply the travel time for an optical pulse through all L -layers. Following Ref. [226], the propagation distance in a square interferometer mesh is $D_W = N \cdot D_{MZI}$, where D_{MZI} is the length of each MZI within the mesh. In the nonlinear activation layer, the propagation length will be dominated by the delay line required to match the optical and electrical delays, and is given by

$$D_f = (\tau_{oe} + \tau_{nl} + \tau_{rc}) \cdot v_g, \quad (2.9)$$

where the group velocity $v_g = c_0/n_{\text{eff}}$ is the speed of optical pulses in the waveguide. Therefore,

$$\text{latency} = \underbrace{L \cdot N \cdot D_{\text{MZI}} \cdot v_g^{-1}}_{\text{Interferometer mesh}} + \underbrace{L \cdot (\tau_{\text{oe}} + \tau_{\text{nl}} + \tau_{\text{rc}})}_{\text{Activation function}}. \quad (2.10)$$

Equation 2.10 indicates that the latency contribution from the interferometer mesh scales with the product LN , which is the same scaling as predicted in Ref. [226]. On the other hand, the activation function adds to the latency independently of N because each activation circuit is applied in parallel to all N -vector elements.

For concreteness, we assume $D_{\text{MZI}} = 100 \mu\text{m}$ and $n_{\text{eff}} = 3.5$. Following our assumption in the previous section of using no nonlinear electrical signal conditioner in the activation function, $\tau_{\text{nl}} = 0$ ps. Typical group delays for integrated transimpedance amplifiers used in optical receivers can range from $\tau_{\text{oe}} \approx 10$ to 100 ps. Moreover, assuming an RC-limited phase modulator speed of 50 GHz yields $\tau_{\text{rc}} \approx 20$ ps. Therefore, if we assume a conservative value of $\tau_{\text{oe}} = 100$ ps, a network dimension of $N \approx 100$ would have a latency of 237 ps per layer, with equal contributions from the mesh and the activation function. For a ten layer network ($L = 10$) the total latency would be approximately 2.4 ns, still orders of magnitude lower than the latency typically associated with GPUs.

2.4.3 Physical footprint

The physical footprint of the ONN consists of the space taken up by both the linear interferometer mesh and the optical and electrical components of the activation function. Neglecting the electrical control lines for tuning each MZI, the total footprint of the ONN is

$$A = \underbrace{L \cdot N^2 \cdot A_{\text{MZI}}}_{\text{Interferometer mesh}} + \underbrace{L \cdot N \cdot A_{\text{f}}}_{\text{Activation function}}, \quad (2.11)$$

where $A_{\text{MZI}} = D_{\text{MZI}} \cdot H_{\text{MZI}}$ is the area of a single MZI element in the mesh and $A_{\text{f}} = D_{\text{f}} \cdot H_{\text{f}}$ is the area of a single activation function.

In the direction of propagation, D_{f} is dominated by the waveguide optical delay line required to match the delay of the electrical signal pathway. Based on the previous discussion of the activation function's latency, $\tau_{\text{opt}} = 120$ ps corresponds to a total waveguide length of $D_{\text{f}} \approx 1$ cm. For simplicity, we assume this delay is achieved using a straight waveguide, which results in a large footprint but with optical losses that can be very low. For example, in silicon waveguides losses below 0.5 dB/cm have been experimentally demonstrated [217]. In principle, incorporating waveguide bends or resonant optical elements could significantly reduce the activation function's footprint. For example, coupled micro ring arrays have experimentally achieved group delays of 135 ps over a bandwidth of 10 GHz in a $0.03 \text{ mm} \times 0.25 \text{ mm}$ footprint [41].

Transverse to the direction of propagation, the activation function footprint will be dominated by the electronic components of the optical-to-electrical conversion circuit. In principle, compact

waveguide photodetectors and modulators can be utilized. However, the components of the transimpedance amplifier may be challenging to integrate in the area available between neighboring output waveguides of the interferometer mesh. One possibility towards achieving a fully integrated opto-electronic ONN would be to use so-called *amplifier-free* optical receivers [169], where ultra-low capacitance detectors provide high-speed opto-electronic conversion. Similarly to the experimental demonstration in Ref. [170], the amplifier-free receiver could be integrated directly with a high efficiency (e.g. effectively a low V_π) electro-optic modulator. Compact electro-absorption modulators could also be utilized. In addition to achieving a compact footprint, operating without an amplifier would also result in an order of magnitude reduction in both power consumption and latency, with the later reducing the required length of the optical delay line and thus the footprint.

For the purposes of our analysis, we assume no integration of the electronic transimpedance amplifier and, therefore, that the on-chip components of the activation function fit within the height of each interferometer mesh row, $D_f \leq D_{\text{MZI}} = 60 \mu\text{m}$. Under this assumption and following the scaling in Eq. 2.11, the total footprint of a single ONN layer of dimension $N = 10$ would be $11.0 \text{ mm} \times 0.6 \text{ mm}$. Interestingly, following the latency discussion in the previous section, a single ONN layer of dimension $N = 100$ would have a footprint of $20.0 \text{ mm} \times 6.0 \text{ mm}$, with equal contribution from the activation function and from the mesh.

2.4.4 Speed

The speed, or computational capacity, of the ONN, as shown in Fig. 2.1(a), is determined by the number of input vectors, x_0 that can be processed per unit time. Here, we argue that although our activation function is not fully optical, it results in no speed degradation compared to a linear ONN consisting of only interferometer meshes.

The reason for this is that a fully integrated ONN would also include high-speed modulators and detectors on-chip to perform fast modulation and detection of sequences of x_0 vectors and x_L vectors, respectively. We therefore argue that the same high-speed detector and modulator elements could also be integrated between the linear network layers to provide the optical-electrical and electrical-optical transduction for the activation function. State of the art integrated transimpedance amplifiers can already operate at speeds comparable to the optical modulator and detector rates, which are on the order of 50 - 100 GHz [263, 6], and thus would not be a limiting factor in the speed of our architecture.

To perform a matrix-vector multiplication on a conventional CPU requires N^2 multiply-accumulate (MAC) operations, each consisting of a single multiplication and a single addition. Therefore, assuming a photodetector and modulator rate of 10 GHz means that an ONN can effectively perform $N^2 \cdot L \cdot 10^{10}$ MAC/sec. This means that one layer of an ONN with dimension $N = 10$ would effectively perform 10^{12} MAC/sec. Increasing the input dimension to $N = 100$ would then scale the performance of the ONN to 10^{14} MAC/sec per layer. This is two orders of magnitude greater

than the peak performance obtainable with modern GPUs, which typically have performance on the order of 10^{12} floating point operations/sec (FLOPS). Because the power consumption of the ONN scales as LN (assuming passive phase shifters in the mesh) and the speed scales as LN^2 , the energy per operation is minimized for large N (Table 2.2). Thus, for large ONNs the power consumption associated with the electro-optic conversion in the activation function can be amortized over the parallelized operation of the linear mesh.

We note that the activation function circuit shown in Fig. 2.1(c) can be modified to remove the matched optical delay line by using very long optical pulses. This modification may be advantageous for reducing the footprint of the activation and would result in $\tau_{\text{opt}} \ll \tau_{\text{ele}}$. However, this results in a reduction of the ONN speed, which would then be limited by the combined activation delay of all L nonlinear layers in the network, $\sim (L \cdot \tau_{\text{ele}})^{-1}$.

2.5 Comparison with the Kerr effect

All-optical nonlinearities such as bistability and saturable absorption have been previously considered as potential activation functions in ONNs [3, 227]. An alternative implementation of the activation function in Fig. 2.1(c) could consist of a nonlinear MZI, with one of its arms having a material with Kerr nonlinear optical response. The Kerr effect is a third-order optical nonlinearity which generates a change in the refractive index, and thus a nonlinear phase shift, which is proportional to the input pulse intensity. In this section we compare the electro-optic activation function introduced in the previous section [Fig. 2.1(c)] to such an alternative all-optical activation function using the Kerr effect, highlighting how the electro-optic activation can achieve a lower activation threshold.

Unlike the electro-optic activation function, the Kerr effect is lossless and has no latency because it arises from a nonlinear material response, rather than a feedforward circuit. A standard figure of merit for quantifying the strength of the Kerr effect in a waveguide is through the amount of nonlinear phase shift generated per unit input power per unit waveguide length. This is given mathematically by the expression

$$\Gamma_{\text{Kerr}} = \frac{2\pi n_2}{\lambda_0 A}, \quad (2.12)$$

where n_2 is the nonlinear refractive index of the material and A is the effective mode area. Γ_{Kerr} ranges from $100 \text{ (W}\cdot\text{m)}^{-1}$ in chalcogenide to $350 \text{ (W}\cdot\text{m)}^{-1}$ in silicon [126]. An equivalent figure of merit for the electro-optic feedforward scheme can be mathematically defined as

$$\Gamma_{\text{EO}} = \pi \frac{\alpha \mathcal{R}G}{V_\pi L}, \quad (2.13)$$

where $V_\pi L$ is the phase modulator figure of merit. The figures of merit described in Eqs. 2.12-2.13 can be represented as an activation threshold (Eq. 2.8) via the relationship $P_{\text{th}} = \frac{\Delta\phi|_{\delta T=0.5}}{\Gamma L}$, for a given waveguide length, L where the electro-optic phase shift or nonlinear Kerr effect take place.

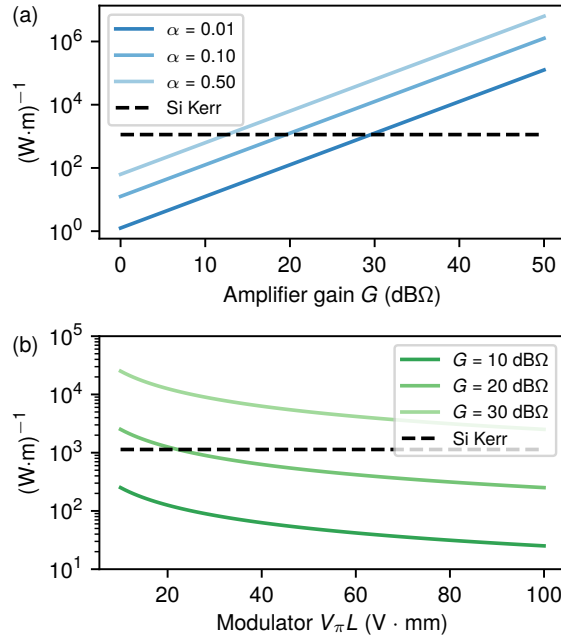


Figure 2.4: Nonlinear parameter Γ_{EO} for the electro-optic activation as a function of (a) gain, G , for $\alpha = 0.50, 0.10$, and 0.01 and (b) modulator $V_\pi L$. The nonlinear parameter associated with the optical Kerr effect, Γ_{KERR} in a Silicon waveguide of cross sectional area $A = 0.05 \mu m^2$ corresponds to the black dotted line.

A comparison of Eq. 2.12 and Eq. 2.13 indicates that while the strength of the Kerr effect is largely fixed by waveguide design and material choice, the electro-optic scheme has several degrees of freedom which allow it to potentially achieve a stronger nonlinear response. The first design parameter is the amount of power tapped off to the photodetector, which can be increased to generate a larger voltage at the phase modulator. However, increasing α also increases the *linear* signal loss through the activation which does not contribute to the nonlinear mapping between the input and output of the ONN. Therefore, α should be minimized as long as the optical power routed to the photodetector is large enough to be above the noise equivalent power level.

On the other hand, the product $\mathfrak{R}G$ determines the conversion efficiency of the detected optical power into an electrical voltage. Fig. 2.4(a) compares the nonlinearity strength of the electro-optic activation (blue lines) to that of an implementation using the Kerr effect in silicon (black dashed line) for several values of α , as a function of G . The responsivity is fixed at $\mathfrak{R} = 1.0$ A/W. We observe that tapping out 10% of the optical power requires a gain of 20 $dB\Omega$ to achieve a nonlinear phase shift equivalent threshold to that of a silicon waveguide where $A = 0.05 \mu m^2$ for the same amount of input optical power. Tapping out only 1% of the optical power requires an additional 10 $dB\Omega$ of gain to maintain this equivalence. We note that the gain range considered in Fig. 2.4(a) is well within the regime of what has been demonstrated in integrated transimpedance amplifiers

for optical receivers [6, 218, 169]. In fact, many of these systems have demonstrated much higher gain. In Fig. 2.4(a), the phase modulator $V_\pi L$ was fixed at 20 V·mm. However, because a lower $V_\pi L$ translates into an increased phase shift for a given applied voltage, this parameter can also be used to enhance the nonlinearity. Fig. 2.4(b) demonstrates the effect of changing the $V_\pi L$ for several values of G , again, with a fixed responsivity $\mathfrak{R} = 1.0$ A/W. This demonstrates that with a reasonable level of gain and phase modulator performance, the electro-optic activation function can trade off an increase in latency for a significantly lower optical activation threshold than the Kerr effect.

2.6 Machine learning tasks

In this section, we apply the electro-optic activation function introduced above to several machine learning tasks. In Sec. 2.6.1, we simulate training an ONN to implement an exclusive-OR (XOR) logical operation. The network is modeled using `neuroptica` [22], a custom ONN simulator written in Python, which trains the simulated networks only from physically measurable field quantities using the on-chip backpropagation algorithm introduced in Ref. [101]. In Sec. 2.6.2, we consider the more complex task of using an ONN to classify handwritten digits from the Modified NIST (MNIST) dataset, which we model using the `neurophox` [176] package and `tensorflow` [1], which computes gradients using automatic differentiation. In both cases, we model the values in the network as complex-valued quantities and represent the interferometer meshes as unitary matrices parameterized by phase shifters.

2.6.1 Exclusive-OR Logic Function

An exclusive-OR (XOR) is a logic function which takes two inputs and produces a single output. The output is *high* if only one of the two inputs is *high*, and *low* for all other possible input combinations. In this example, we consider a multi-input XOR which takes N input values, given by $x_1 \dots x_N$, and produces a single output value, y . The input-output relationship of the multi-input XOR function is a generalization of the two-input XOR. For example, defining logical *high* and *low* values as 1 and 0, respectively, a four-input XOR has an output table indicated the desired values in Fig. 2.5(b). We select this task for the ONN to learn because it requires a non-trivial level of nonlinearity, meaning that it could not be implemented in an ONN consisting of only linear interferometer meshes.

The architecture of the ONN used to learn the XOR is shown schematically in Fig. 2.5(a). The network consists of L layers, with each layer constructed from an $N \times N$ unitary interferometer mesh followed by an array of N parallel electro-optic activation functions, with each element corresponding to the circuit in Fig. 2.1(c). After the final layer, the lower $N - 1$ outputs are dropped to produce a single output value which corresponds to y . Unlike the ideal XOR input-output relationship described above, for the XOR task learned by the ONN we normalize the input vectors such that

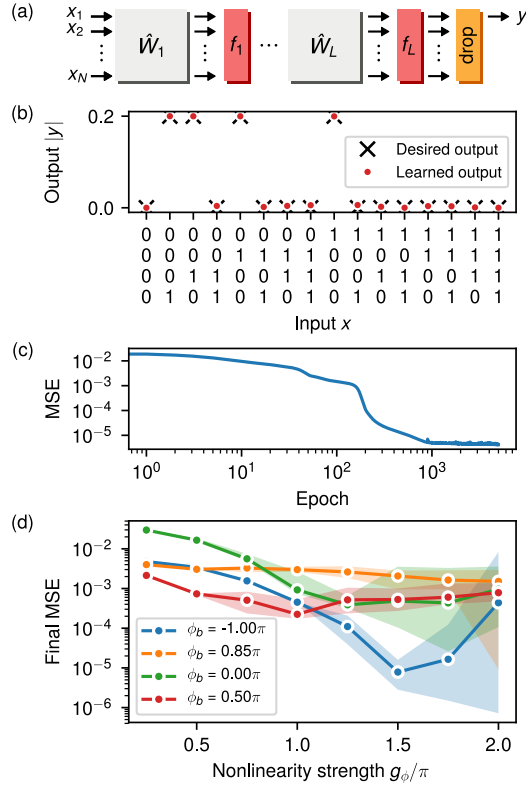


Figure 2.5: (a) Architecture of an L -layer ONN used to implement an N -input XOR logic function. (b) Red dots indicate the learned input-output relationship of the XOR for $N = 4$ on a 2-layer ONN. Electro-optic activation functions are configured with gain $g = 1.75\pi$ and biasing phase $\phi_b = \pi$. (c) Mean squared error (MSE) versus training epoch. (d) Final MSE after 5000 epochs averaged over 20 independent training runs vs activation function gain. Different lines correspond to the responses shown in Fig. 2.2, with $\phi_b = 1.00\pi$, 0.85π , 0.00π , and 0.50π . Shaded regions correspond to the range (minimum and maximum) final MSE from the 20 training runs.

they always have an L_2 norm of 1. This constraint is equivalent to enforcing a constant input power to the network. Additionally, because the activation function causes the optical power level to be attenuated at each layer, we take the *high* output state to be a value of 0.2, as shown in Fig. 2.1(b). The *low* output remains at a value of 0.0. An alternative to using a smaller amplitude for the output high state would be to add additional ports with fixed power biases to increase the total input power to the network, similarly to the XOR demonstrated in Ref. [101].

In Fig. 2.5(b) we show the four-input XOR input-output relationship which was learned by a two-layer ONN. The electro-optic activation functions were configured to have a gain of $g = 1.75\pi$ and biasing phase of $\phi_b = \pi$. This biasing phase configuration corresponds to the ReLU-like response shown in Fig. 2.2(a). The black markers indicate the desired output values while the red circles indicate the output learned by the two-layer ONN. Fig. 2.5(b) indicates excellent agreement between

the learned output and the desired output. The evolution of the mean squared error (MSE) between the ONN output and the desired output during training confirms this agreement, as shown in Fig. 2.5(c), with a final MSE below 10^{-5} .

To train the ONN, a total of $2^N = 16$ training examples were used, corresponding to all possible binary input combinations along the x-axis of Fig. 2.5(b). All 16 training examples were fed through the network in a batch to calculate the mean squared error (MSE) loss function. The gradient of the loss function with respect to each phase shifter was computed by backpropagating the error signal through the network to calculate the loss sensitivity at each phase shifter [101]. The above steps were repeated until the MSE converged, as shown in Fig. 2.5(c). Only the phase shifter parameters were optimized by the training algorithm, while all parameters of the activation function were unchanged.

To demonstrate that the nonlinearity provided by the electro-optic activation function is essential for the ONN to successfully learn the XOR, in Fig. 2.5(d) we plot the final MSE after 5000 training epochs, averaged over 20 independent training runs, as a function of the activation function gain, g_ϕ . The shaded regions indicate the minimum and maximum range of the final MSE over the 20 training runs. The four lines shown in Fig. 2.5(d) correspond to the four activation function bias configurations shown in Fig. 2.2.

For the blue curve in Fig. 2.5(d), which corresponds to the ReLU-like activation, we observe a clear improvement in the final MSE with an increase in the nonlinearity strength. We also observe that for very high nonlinearity, above $g_\phi = 1.5\pi$, the range between the minimum and maximum final MSE broadens and the mean final MSE increases. However, the best case (minimum) final MSE continues to decrease, as indicated by the lower border of the shaded blue region. This trend indicates that although increasing nonlinearity improves the ONN's ability to learn the XOR function, very high levels of nonlinearity may also prevent the training algorithm from converging.

A trend of decreasing MSE with increasing nonlinearity is also observed for the activation corresponding to the green curve in Fig. 2.5(d). However, the range of MSE values begins to broaden at a lower value of $g_\phi = 1.0\pi$. Such broadening may be a result of the changing slope in the activation function output, as shown in Fig. 2.2(e). For the activation functions corresponding to the red and orange curves in Fig. 2.5(d), the final MSE decreases somewhat with an increase in g_ϕ , but generally remains much higher than the other two activation function responses. We conclude that these two responses are not as well suited for learning the XOR function. Overall, these results demonstrate that the flexibility of our architecture to achieve specific forms of nonlinear activation functions is important for the successful operation of an ONN.

2.6.2 Handwritten Digit Classification

The second task we consider for demonstrating the activation function is classifying images of handwritten digits from the MNIST dataset, which has become a standard benchmark problem for ANNs [133]. The dataset consists of 70,000 grayscale 28×28 pixel images of handwritten digits between 0

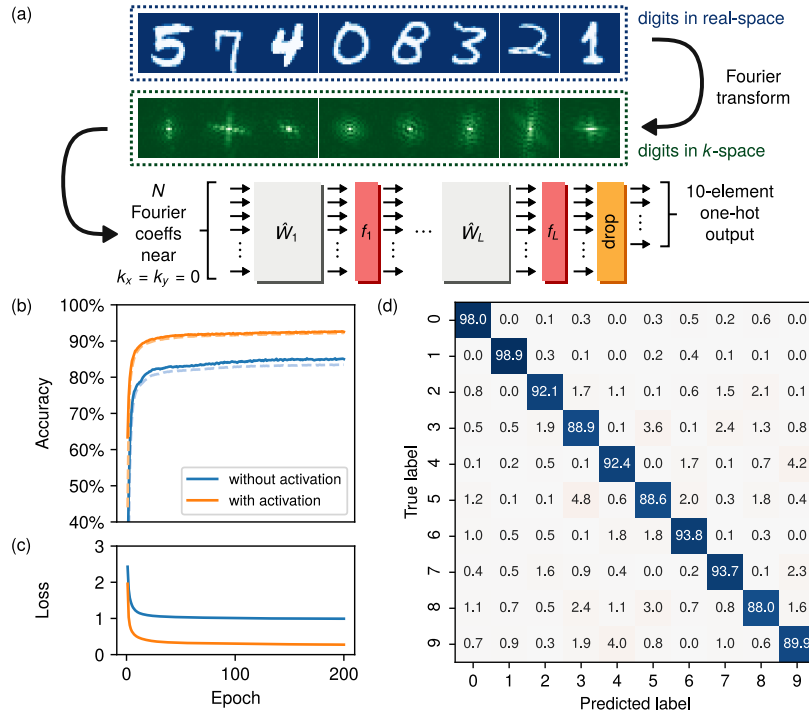


Figure 2.6: (a) Schematic of an optical image recognition setup based on an ONN. Images of handwritten numbers from the MNIST database are preprocessed by converting from real-space to k -space and selecting N Fourier coefficients associated with the smallest magnitude k -vectors. (b) Test accuracy (solid lines) and training accuracy (dashed lines) during training for a two layer ONN without activation functions (blue) and with activation functions (orange). $N = 16$ Fourier components were used as inputs to the ONN and each vector was normalized such that its L_2 norm is unity. The activation function parameters were $g_\phi = 0.05\pi$ and $\phi_b = 1.00\pi$. (c) Cross entropy loss during training. (d) Confusion matrix, specified in percentage, for the trained ONN with the electro-optic activation function.

and 9. Several representative images from the dataset are shown in Fig. 2.6(a).

To reduce the number of input parameters, and hence the size of the neural network, we use a preprocessing step to convert the images into a Fourier-space representation. Specifically, we compute the 2D Fourier transform of the images which is defined mathematically as $c(k_x, k_y) = \sum_{m,n} e^{jk_x m + jk_y n} g(m, n)$, where $g(m, n)$ is the gray scale value of the pixel at location (m, n) within the image. The amplitudes of the Fourier coefficients $c(k_x, k_y)$ are shown below their corresponding images in Fig. 2.6(a). These coefficients are generally complex-valued, but because the real-space map $g(m, n)$ is real-valued, the condition $c(k_x, k_y) = c^*(-k_x, -k_y)$ applies.

We observe that the Fourier-space profiles are mostly concentrated around small k_x and k_y , corresponding to the center region of the profiles in Fig. 2.6(a). This is due to the slowly varying spatial features in the images. We can therefore expect that most of the information is carried by the

small- k Fourier components, and with the goal of decreasing the input size, we can restrict the data to N coefficients with the smallest $k = \sqrt{k_x^2 + k_y^2}$. An additional advantage of this preprocessing step is that it reduces the computational resources required to perform the training process because the neural network dimension does not need to accommodate all $28^2 = 784$ pixel values as inputs.

Fourier preprocessing is particularly relevant for ONNs for two reasons. First, the Fourier transform has a straightforward implementation in the optical domain using techniques from Fourier optics involving standard components such as lens and spatial filters [79]. Second, this approach allows us to take advantage of the fact that ONNs are *complex*-valued functions. That is to say, the N complex-valued coefficients $c(k_x, k_y)$ can be handled by an N -dimensional ONN, whereas to handle the same input using a real-valued neural network requires a twice larger dimension. The ONN architecture used in our demonstration is shown schematically in Fig. 2.6(a). The N Fourier coefficients closest to $k_x = k_y = 0$ are fed into an optical neural network consisting of L layers, after which a drop-mask reduces the final output to 10 components. The intensity of the 10 outputs are recorded and normalized by their sum, which creates a probability distribution that may be compared with the one-hot encoding of the digits from 0 to 9. The loss function is defined as the cross-entropy between the normalized output intensities and the correct one-hot vector.

During each training epoch, a subset of 60,000 images from the dataset were fed through the network in batches of 500. The remaining 10,000 image-label pairs were used to form a test dataset. For a two-layer network with $N = 16$ Fourier components, Fig. 2.6(b) compares the classification accuracy over the training dataset (solid lines) and testing dataset (dashed lines) while Fig. 2.6(c) compares the cross entropy loss during optimization. The blue curves correspond to an ONN with no activation function (e.g. a linear optical classifier) and the orange curves correspond to an ONN with the electro-optic activation function configured with $g_\phi = 0.05\pi$, $\phi_b = 1.00\pi$, and $\alpha = 0.1$. The gain setting in particular was selected heuristically. We observe that the nonlinear activation function results in a significant improvement to the ONN performance during and after training. The final validation accuracy for the ONN with the activation function is 93%, which amounts to an 8% difference as compared to the linear ONN which achieved an accuracy of 85%.

The confusion matrix computed over the testing dataset is shown in Fig. 2.6(d). We note that the prediction accuracy of 93% is high considering that only $N = 16$ complex Fourier components were used, and the network is parameterized by only $2 \times N^2 \times L = 1024$ free parameters. Moreover, this prediction accuracy is comparable with the 92.6% accuracy achieved in a fully-connected linear classifier with 4010 free parameters taking *all* of the $28^2 = 784$ real-space pixel values as inputs [133]. Finally, in Table 2.3 we show that the accuracy can be further improved by including a third layer in the ONN and by making the activation function gain a trainable parameter. This brings the testing accuracy to 94%. Based on the parameters from Table 2.1 and the scaling from Table 2.2, the 3 layer handwritten digit classification system would consume 4.8 W while performing 7.7×10^{12} MAC/sec. Its prediction latency would be 1.5 ns.

# Layers	Without activation	With activation	
		Untrained	Trained
1	85.00%	89.80%	89.38%
2	85.83%	92.98%	92.60%
3	85.16%	92.62%	93.89%

Table 2.3: Accuracy on the MNIST testing dataset after optimization. The phase gain, g_ϕ , of each layer was optimized during training.

2.7 Experimental realization of arbitrary electro-optic activation functions

In this section, we experimentally demonstrate a the electro-optic architecture for realizing optical-to-optical activation functions described earlier in this chapter using a custom-fabricated silicon-nitride chip. Related proposals of realizing optical nonlinearities using electro-optics have been proposed in Refs. [136, 145, 94, 238]. In our scheme, rather than using traditional optical nonlinearities we fabricate structures on a photonic integrated circuit to measure a small portion of the incoming optical signal power and use electro-optic modulators to modulate the original optical signal. Our prototype relies on thermo-optic modulation but, in principle, the demonstrated activation can use fast modulation mechanisms to enable ONNs operating with GHz-rate computational speeds. This activation circuit allows for the realization of strong nonlinearities without the requirement of having additional optical sources between each layer of the network [238]. We also demonstrate an extension of the circuit capabilities originally proposed in Ref. [257] to realize arbitrary nonlinearities via electrical signal processing with ultra-low activation thresholds. We focus on an implementation of the activation function that does not use optical gain elements.

2.7.1 Fabricated device

The schematic of the proposed nonlinear activation function circuit is shown in Fig. 2.7 (b), and Figure 2.7(c) shows a micrograph of the on-chip nonlinear activation function circuit using a SiN waveguide technology. A schematic of an early prototype chip layout is shown in Figure 2.8; this design was fabricated, but a smaller version of this chip with fewer layers was used for the measurements in Section 2.8.

The circuit consists of a 1:99 directional coupler (DC) and a MZI with a top metal thermal phase shifter. Note that the low speed of the thermal phase shifter limits the operational speed of the prototype device. This limited operational speed is not concerning since the main purpose of this experiment is to demonstrate the capabilities of the proposed circuit to generate nonlinear activation functions. Fabricating the proposed circuit in technologies that provide high-speed intensity

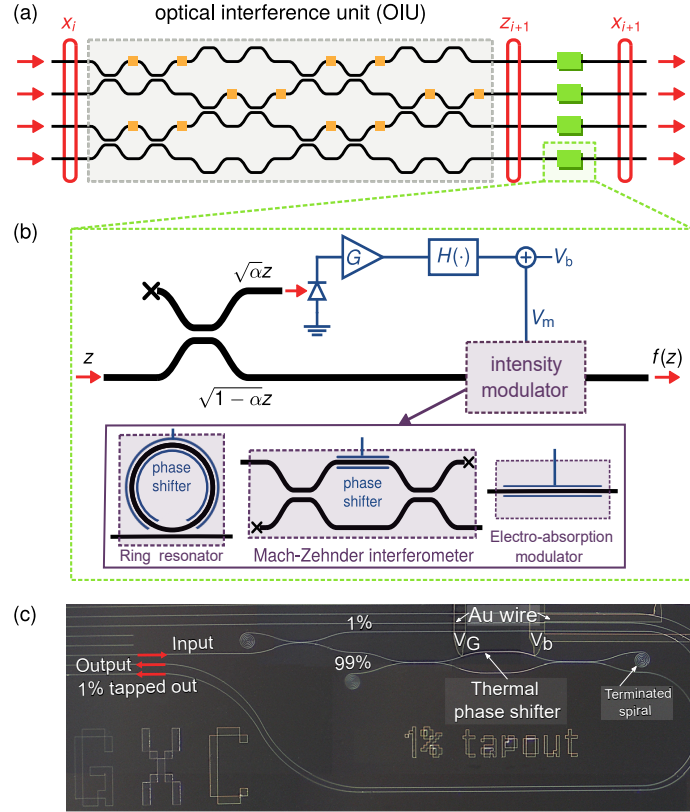


Figure 2.7: (a) Schematic of the optical interferometer mesh implementation of a single layer of the feedforward neural network. (b) Schematic of the proposed optical-to-optical activation function circuit. The black and blue lines represent optical waveguides and electrical signal pathways respectively. (c) Optical image of the fabricated activation function circuit with Mach-Zehnder interferometer modulator.

modulation, such as silicon photonics, can provide fast operational speed [211, 228, 16]. The 1% tapped out port of the DC and the cross-port of the MZI are routed to the edge of the die for edge coupling, while the unused ports are terminated by small spirals, which scatter light due to the small bend radius of the spirals and prevent signal reflection.

Figure 2.9 shows the measured MZI output for various applied voltages to the phase shifter. The thermal phase shifter requires 12.8V for a π -phase shift and MZI cross-port shows an extinction ratio larger than 40 dB. The insertion loss of the device excluding two 3.25 dB fiber-to-waveguide coupling loss is around 1 dB.

The phase shift in a high-speed intensity modulator typically follows the applied modulating voltage linearly as in Eq. 2.2. However, thermal phase shifters induce a phase shift proportional to

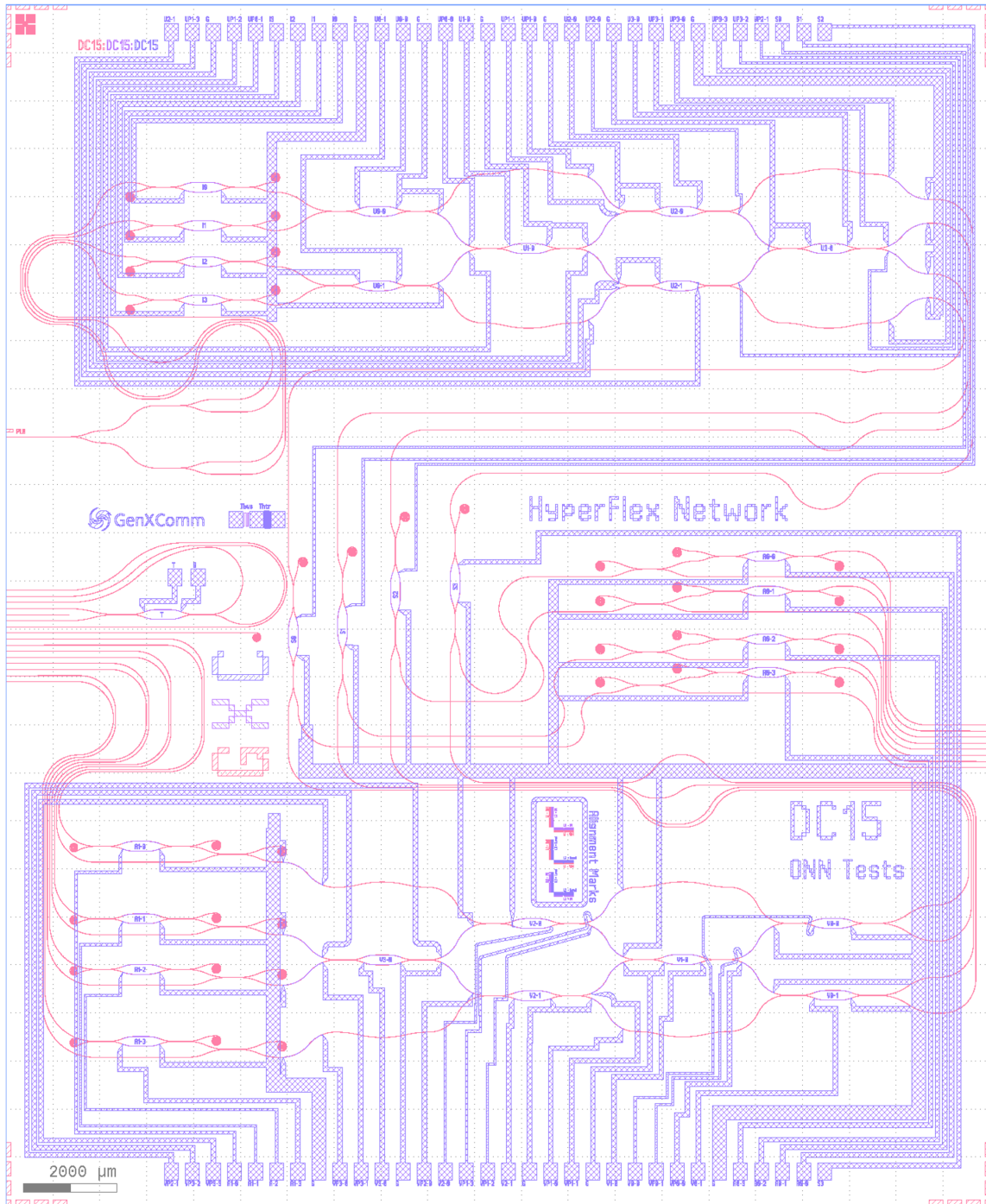


Figure 2.8: Chip layout of the fabricated device: a 4×4 multi-layer optical neural network with multiple layers of nonlinear activation functions. Silicon nitride waveguides are shown in red, while electrical connections for the thermal phase shifters are shown in purple. Note that measured activation function transmissions in Section 2.8 used a smaller version of this chip with fewer layers.

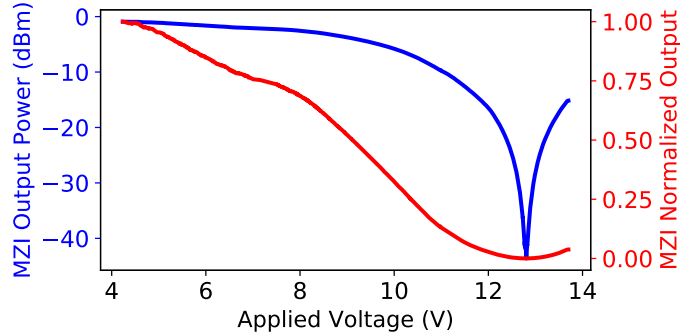


Figure 2.9: The output of the MZI for various voltages applied to the phase shift for 0dBm input power to the directional coupler. The blue line shows the output power of the MZI in dBm, and the red line shows the corresponding normalized MZI output.

the square of the applied modulating voltage given by

$$\Delta\phi_{\text{thermal}} = \pi \left(\frac{V_m}{V_\pi} \right)^2. \quad (2.14)$$

2.8 Experimental results

This section details two sets of experiments demonstrating the capabilities of the proposed activation function circuit. Figure 2.10(a) depicts the measurement setup, while Figs. 2.10(b) and 2.10(c) present the block diagram of the two measurement setups. In the first experiment, the tapped out power is converted to a voltage signal and amplified by an optical receiver circuit and directly used to modulate the thermal phase shifter (Fig. 2.10(b)). This setup implements a limited number of activation functions. In the second setup, the direct controller is replaced by a re-configurable lookup table that generates arbitrary nonlinear activation functions. The microcontroller in Fig. 2.10(c) implements the lookup table.

To perform the experiments, we first beam a 1550 nm laser through a variable optical attenuator (VOA). The VOA allows us to control the amplitude of the input optical signal. Next, the signal is sent through a polarization controller. Integrated waveguides are polarization sensitive, so the polarization controller is used to minimize the coupling loss. The signal is then sent to the on-chip nonlinear activation function circuit through a fiber array. Finally, the activated signal exits through the fiber array. An electrical probe card is used to control the on-chip thermal phase shifter.

2.8.1 Direct controller

We first consider the *direct controller* experimental setup, as shown in Fig. 2.10(b), which routes the 1% tapped out signal from the directional coupler to a 75 MHz Thorlabs photoreceiver (PDB420C)

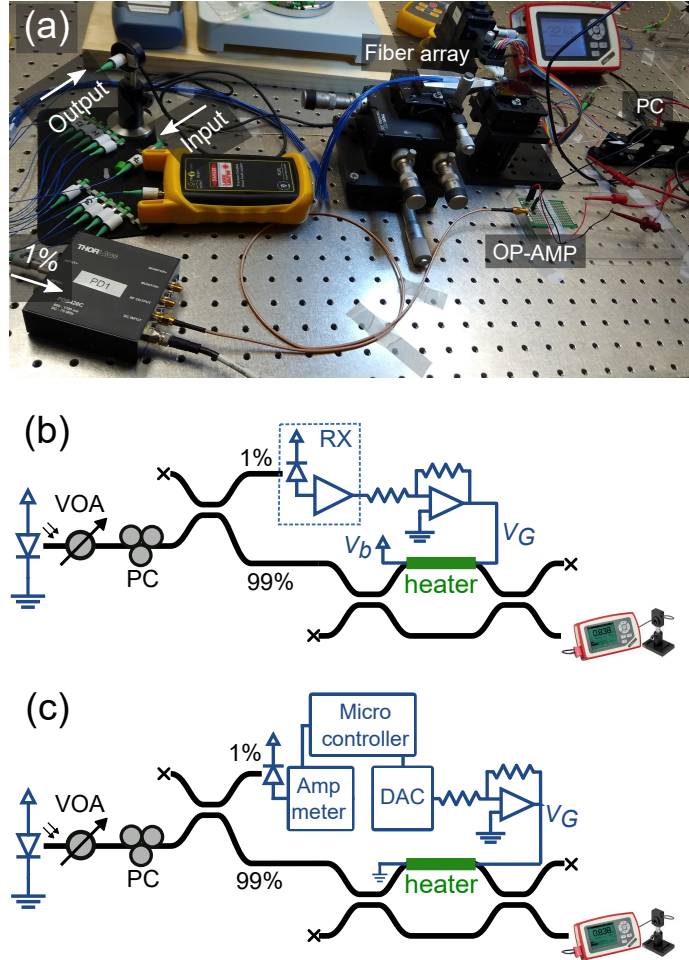


Figure 2.10: (a) A photo of the measurement setup, (b) the block diagram of the measurement setup with direct controller, (c) the block diagram of the measurement setup with the look up table.

with a conversion gain of 250K V/W. The maximum input power to the device is limited to -8 dBm to ensure linear photoreceiver operation. The output of the photoreceiver (RX in Fig. 2.10(b)) is amplified by an operational amplifier (OP-AMP in Fig. 2.10(a)) and is connected to the thermal phase shifter on the top MZI arm. The opposite end of the thermal heater is connected to a power supply for controlling the initial MZI bias. With this biasing configuration, the effective modulating voltage equals the difference of the bias voltage and the tapped out photo-generated voltage: $V_m = V_G - V_b$.

Figure 2.11 plots the normalized output power $|f(z)|^2$ as a function of normalized input power $|z|^2$ and compares it with the simulation result, at four different bias voltages applied to the thermal phase shifter.

In modeling the performance of the device, we assume that no nonlinear signal conditioning

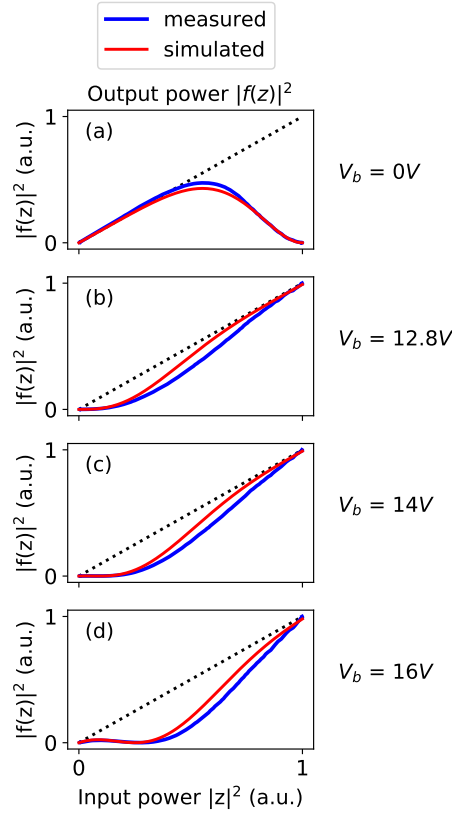


Figure 2.11: The normalized output power ($|f(z)|^2$) of nonlinear activation function circuit for various the normalized input power ($|z|^2$) at several bias point. In all of the figures, the normalized output power is relative to the maximum achievable output power of the circuit, which relates to minimum MZI attenuation.

was applied to the electrical signal pathway, i.e. $V_G = G\Re\alpha|z|^2$. We observe excellent agreement between the measured and simulated activation function response, as shown in Fig. 2.11. The small difference between the measurements and simulation results could be due to the nonlinear response of the photoreceiver. Figures 2.11(b), 2.11(c), and 2.11(d), corresponding to $V_b = 12.8V$, $14V$, and $16V$, exhibit a response which is similar to the ReLU activation function: optical signal transmission is low for small input values and high for large input values. For the bias of $V_b = 14V$ and $16V$, transmission at low input power values is slightly increased compared to the response at $V_b = 12.8V$. Unlike the ideal ReLU response, the activation at $V_b = 14V$ and $16V$ is not entirely monotonic because transmission first goes to zero before increasing [257]. The response shown in Fig. 2.11(a), corresponding to $V_b = 0.0V$, is quite different. It demonstrates a saturation response in which the output is suppressed for higher input values but enhanced for lower input values.

As shown in Fig. 2.11, the bias voltage changes the activation response. The same control

circuitry which programs linear interferometer meshes can control the activation response through the bias voltage. The resulting device is a programmable ONN that can implement a range of activation functions.

A fully integrated ONN in a high-speed photonic platform, such as silicon photonics [228, 16] would include on-chip high-speed modulators and detectors to modulate and detect the sequences of input vectors to the input layer of the ONN and output vectors of the output layer of the ONN, respectively. The same high-speed detector and modulator elements could also be integrated between the optical interference unit to provide the activation function circuit. State of the art integrated transimpedance amplifiers operate at speeds comparable to the optical modulator and detector rates, which are on the order of 50 - 100 GHz [263, 6]. Therefore, the proposed activation function circuit would not be a limiting factor in the speed of the ONN.

2.8.2 Lookup table controller

We now consider the *lookup table controller* experimental setup, which uses a voltage lookup table to implement the nonlinear electrical signal transformation, H . Specifically, the lookup table maps the tapped out photogenerated current to a modulating voltage applied to the MZI phase shifter. To produce the lookup table, two traces of MZI normalized output as a function of applied voltage to the phase shifter and photogenerated current as a function of optical input power P_{in} are used. We linearly combine these two traces to produce a 2-dimensional map of the optical output power of the activation function circuit as a function of input power to the circuit (P_{in}) and normalized output of MZI. The lookup table is determined by overlaying the target activation function on the map; it is then implemented by a microcontroller. Figure 2.10 (c) shows the block diagram of the test setup with a lookup table controller. The 1% tapped output of the DC is connected to a photodetector with a responsivity \mathfrak{R} of 1 A/W. The photogenerated current of the photodetector is measured by a B&K Precision 393 ammeter. As expected, the measured current is proportional to the optical input power $I_{\text{pd}} = \alpha \mathfrak{R} P_{\text{in}}$. The digital output of the ammeter is sent to a microcontroller to specify the modulating voltage for controlling the phase shifter. The voltage of the phase shifter is set using a lookup table for a specific activation function. Figures 2.12(a) and 2.12(b) demonstrate two activation function of `sigmoid` and `modReLU` [245] overlaid on the 2D power throughput map of the activation function circuit, respectively. Figure 2.12(c) and 2.12(d) compare the target `sigmoid` function and target `modReLU` function with their measurement result respectively. Both measured responses agree very well with the target functions.

Using the lookup table to control the activation response provides a tool to heuristically select an activation function response or to directly optimize the activation function using a training routine. This realization of a controllable optical-to-optical nonlinearity allows ONNs to be applied to a broader classes of machine learning tasks [171]. However, implementing the lookup table on a microcontroller limits the operation speed of the activation function circuit to sub-GHz range. For

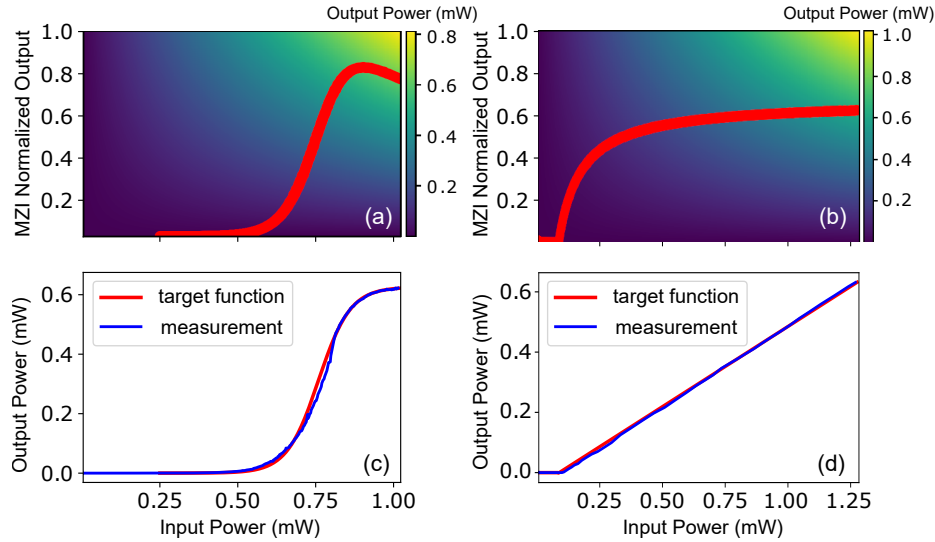


Figure 2.12: (a) Target **sigmoid** function overlaid on the 2D power throughput map of the nonlinear activation function, (b) target **modReLU** function overlaid on the 2D power throughput map of the nonlinear activation function, (c) measurement result compared its target **sigmoid** function, (d) measurement result compared to its target **modReLU** function.

a specific ONN application, one can use a moderate-speed flexible lookup table implemented on a microcontroller or field-programmable gate array to optimize the activation function. The associated transfer function can then be related to the optimized lookup table, and a piecewise linear approximation can synthesize the optimized transfer function. In a high-speed (GHz) implementation, the circuitry of the piecewise linear function can consist of an application-specific integrated circuit in a high-speed analog/RF circuit platform. A number of technologies with high transit frequencies can be utilized for this purpose. Examples include SiGe BiCMOS, i.e. combination of bipolar and complementary metal-oxide-semiconductor (CMOS) technology, III-V technologies, and advanced CMOS technologies provide high-speed platforms for implementing over 50 GHz bandwidth analog/RF circuits [263, 6, 73].

2.9 MNIST classification using experimental results

In this section, we numerically characterize the performance of the activation function on the benchmark machine learning task of classifying images from the MNIST dataset used in Section 2.6.2 using the experimentally measured data of the activation function from the fabricated chip. The ONN setup is shown schematically in Fig. 2.13(a), and consists of a sequence of linear layers, corresponding to interferometer meshes [226], and nonlinear activation layers. The last layer is a drop layer that reduces the vector to a length of 10 elements, suitable for one-hot detection across the 10 digit classes. After the drop layer, the optical intensity is detected and passed through a softmax

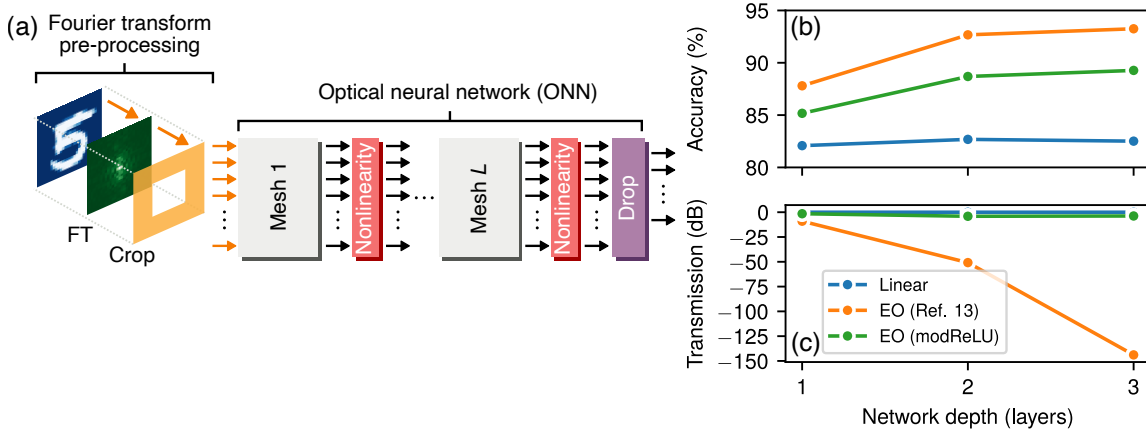


Figure 2.13: (a) Schematic of the optical neural network configuration for classifying handwritten digits from the MNIST dataset. The 28×28 pixel images are first Fourier transformed and cropped before being fed into the ONN. (b) Classification accuracy achieved for various ONN configurations. (c) Optical signal transmission through the neural network, averaged over all samples in the training dataset.

function. As in Ref. [257], before entering the ONN, the images undergo a pre-processing stage consisting of a Fourier transform step and a cropping step. These operations reduce the total size of the input data from $28 \times 28 = 784$ real-space pixels to 16 complex Fourier coefficients. We found that an ONN with 16 inputs resulted in a reasonably high classification performance, but was still feasible to simulate and train numerically.

In practice, the Fourier transformation and cropping steps could be experimentally achieved completely passively with a Fourier optics setup [79].

We now compare the classification performance of the ONN on the digit recognition task for several nonlinearity settings and quantify the optical transmission through the ONN. Figure 2.13(b-c) show the classification accuracy of the ONN on the test dataset and the optical transmission through the ONN as a function of the network depth. The transmission shown in Fig. 2.13(c) is calculated as the mean over the transmission for all samples in the training dataset. These simulations were performed using TensorFlow [4] and the neurophox ONN modeling framework [176, 177], which implements a physical model of the ONN by parameterizing the linear layers in terms of MZI interferometers and phase shifters and complex-valued field quantities. The ONN is trained using the Adam optimizer [116] for 400 epochs with a batch size of 512.

In our comparison, we consider several variants of the ONN in Fig. 2.13(a): a linear ONN with no activation, an electro-optic activation that uses settings similar to those in Ref. [257], and an electro-optic activation implementing the complex modReLU function [245] corresponding to the lookup table implementation measured from our prototype in Fig. 2.12. Unsurprisingly, we observe in Fig. 2.13 that the linear ONN does not benefit from an increase in the network depth because

a sequence of linear transformations is also a linear transformation. In other words, additional linear layers without intermediate nonlinearities do not meaningfully increase the learning capacity of the ONN. The linear ONN achieves a test accuracy of 82% and, because we have assumed lossless interferometer meshes, it exhibits ideal optical transmission which is independent of the number of layers.

In contrast, the electro-optic activation function with settings similar to those used in Section 2.6.2 increases its classification accuracy substantially with additional layers. This ONN achieves a test accuracy of 93% with three layers. However, this relatively high accuracy comes with a high cost in terms of optical signal attenuation. Although nonlinear amplitude responses inherently involve signal attenuation, this activation configuration results in an optical transmission of -144 dB for the network with three layers. In practice, such loss could be prohibitively high due to the finite dynamic range of optical detectors at the output of the ONN.

However, by configuring the electro-optic activation lookup table to synthesize the `modReLU` function, the optical transmission can be increased significantly. We observe that the `modReLU` response results in an optical transmission of -4 dB for the 3 layer network, which is 140 dB larger than the transmission through the network with the electro-optic activation settings from Ref. [257]. However, we note that the ONN with the `modReLU` activation does have a classification accuracy that is reduced by 5% from the activation in Ref. [257]. However, the ONN with the `modReLU` activation still outperforms the linear ONN. The performance of the ONN with the `modReLU` activation could potentially be improved by adjusting (or directly training) the activation threshold. We emphasize that the ability to synthesize the `modReLU` activation is a unique capability of this electro-optic activation function architecture and is an important degree of freedom over all-optical nonlinearities. We note that constraining the ONN to $N = 16$ Fourier coefficients from each input image does somewhat limit accuracy of the MNIST task. Other works have demonstrated that increasing N can lead to an increased classification accuracy in ONNs [178], approaching the performance of conventional artificial neural networks.

2.10 Conclusion

In conclusion, we have introduced an architecture for synthesizing optical-to-optical nonlinearities and demonstrated its use as a nonlinear activation function in a feed forward ONN. Using numerical simulations, we have shown that such activation functions enable an ONN to be successfully applied to two machine learning benchmark problems: (1) learning a multi-input XOR logic function, and (2) classifying handwritten numbers from the MNIST dataset.

Rather than using all-optical nonlinearities, our activation architecture uses intermediate signal pathways in the electrical domain which are accessed via photodetectors and phase modulators.

Specifically, a small portion of the optical input power is tapped out which undergoes analog processing before modulating the remaining portion of the same optical signal. Whereas all-optical nonlinearities have largely fixed responses, a benefit of the electro-optic approach demonstrated here is that signal amplification in the electronic domain can overcome the need for high optical signal powers to achieve a significantly lower activation threshold. For example, we show that a phase modulator V_π of 10 V and an optical-to-electrical conversion gain of 57 dB Ω , both of which are experimentally feasible, result in an optical activation threshold of 0.1 mW. We note that this non-linearity is compatible with the *in situ* training protocol proposed in Ref. [101], which is applicable to arbitrary activation functions.

Our activation function architecture can utilize the same integrated photodetector and modulator technologies as the input and output layers of a fully-integrated ONN, and because each activation function in our proposed scheme is a standalone analog circuit, they can be applied in parallel. This means that an ONN using this activation suffers no reduction in processing speed, despite using analog electrical components. The only trade off made by our design is an increase in latency due to the electro-optic conversion process. However, we find that an ONN with dimension $N = 100$ has a total prediction latency of 2.4 ns/layer, with approximately equal contributions from the propagation of optical pulses through the interferometer mesh and from the electro-optic activation function. Conservatively, we estimate the energy consumption of an ONN with this activation function to be 100 fJ/MAC, but this figure of merit could potentially be reduced by orders of magnitude using highly efficient modulators and amplifier-free optoelectronics [170].

Additionally, we have also presented experimental results of this activation function circuit fabricated on a SiN waveguide technology platform. The capabilities of the fabricated circuit were demonstrated through two experimental setups. In the first experiment, only the nonlinear response of the Mach-Zehnder modulator was used to generate the nonlinear activation function. In this setup, a limited set of activation functions could be realized by varying the bias of the phase shifter. In the second experiment, a lookup table was used to apply a nonlinear modulation signal to the phase shifter which allowed realization of arbitrary nonlinear responses. While the prototype demonstrated in this Chapter relied on thermo-optic modulation, the activation architecture can be readily implemented using much faster modulation mechanisms that are widely used in GHz-rate optical communications [211, 228, 16]. Faster modulation will allow an ONN using this activation to achieve higher computational speeds and lower latencies than conventional digital processors.

Using numerical simulations from the experimentally measured data, we demonstrated that the implemented activation functions improve the accuracy of optical neural networks on the benchmark task of classifying images from the MNIST dataset. Our simulations revealed that the ability to generate arbitrary nonlinear optical transfer functions provides a powerful tool to achieve high performance while maintaining a low optical transmission loss. Compared to a linear ONN with depth of three layers, using the activation from Ref. [257] improves the accuracy of the classification

task by more than 11% but at the cost of over 140 dB optical transmission loss. However, by configuring the lookup table to generate the `modReLU` activation [245] instead of the response from Ref. [257], the optical transmission is improved by more than 140 dB with only 5% degradation in classification accuracy.

Finally, we emphasize that in this activation function, the majority of the signal power remains in the optical domain. There is no need to have a new optical source at each nonlinear layer of the network, as is required in previously demonstrated electro-optic neuromorphic hardware [237, 183, 238] and reservoir computing architectures [130, 62]. While we have focused here on the application of our architecture as an activation function in a feedforward ONN, the synthesis of low-threshold optical nonlinearities using this circuit could be of broader interest for optical computing as well as microwave photonic signal processing applications.

Chapter 3

Optimization on universal linear optical devices

In this chapter, we propose error-tolerant initialization routines and architectures for programmable universal linear optical devices [177]. Universal optical devices can apply arbitrary unitary transformations to a vector of input modes and provide a promising hardware platform for fast and energy-efficient machine learning using light. We simulate the gradient-based optimization of random unitary matrices on universal photonic devices composed of imperfect tunable interferometers. If device components are initialized uniform-randomly, the locally-interacting nature of the mesh components biases the optimization search space towards banded unitary matrices, limiting convergence to random unitary matrices. We detail a procedure for initializing the device by sampling from the distribution of random unitary matrices and show that this greatly improves convergence speed. We also explore mesh architecture improvements such as adding extra tunable beamsplitters or permuting waveguide layers to further improve the training speed and scalability of these devices.

3.1 Introduction

Universal multiport interferometers are optical networks that perform arbitrary unitary transformations on input vectors of coherent light modes. Such devices can be used in applications including quantum computing (e.g. boson sampling, photon walks) [83, 42, 233, 93], mode unscramblers [9], photonic neural networks [225, 102, 258], and finding optimal channels through lossy scatterers [155]. While universal photonic devices have been experimentally realized at a relatively small scale [225, 9], commercial applications such as hardware for energy-efficient machine learning and signal processing can benefit from scaling the devices to up to $N = 1000$ modes. At this scale, fabrication imperfections and components with scale-dependent sensitivities can negatively affect performance.

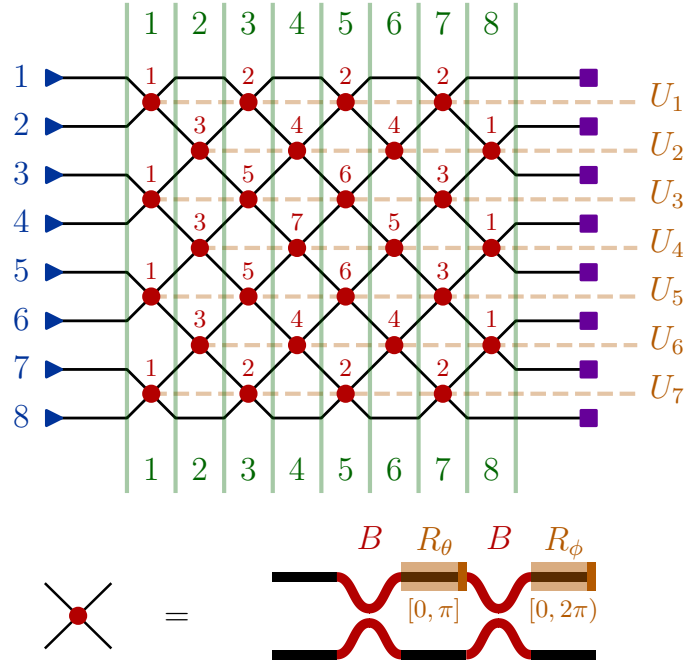


Figure 3.1: Mesh diagram representing the locally interacting rectangular mesh for $N = 8$. The inputs (and single-mode phase shifts at the inputs) are represented by blue triangles. Outputs are represented by purple squares. The MZI nodes are represented by red dots labelled with sensitivity index $\alpha_{n\ell}$ (e.g., $\alpha_{44} = 7$ is the most sensitive node). The nodes represent the Givens rotation U_n (in orange) at vertical layer ℓ (in green). Each photonic MZI node can be represented with 50:50 beamsplitters B (red) and phase shifters R_θ, R_ϕ (orange) with required ranges $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$.

One canonical universal photonic device is the rectangular multiport interferometer mesh [50] shown in Figure 3.1 interfering $N = 8$ modes. In multiport interferometers, an N -dimensional vector is represented by an array of modes arranged in N single-mode waveguides. A unitary operation is applied to the input vector by tuning Mach-Zehnder interferometers (MZIs) represented by the red dots of Figure 3.1. Each MZI is a two-port optical component made of two 50:50 beamsplitters and two tunable single-mode phase shifters. Other mesh architectures have been proposed, such as the triangular mesh [202] (shown in Appendix A.2), the universal cascaded binary tree architecture [156], and lattice architectures where light does not move in a forward-only direction [184, 187, 186].

The scalability of optimizing mesh architectures, especially using gradient-based methods, is limited by the ability of the locally interacting architecture to control the output powers in the mesh. If phase shifts in the mesh are initialized uniform-randomly, light propagates through the device in

a manner similar to a random walk. The off-diagonal, nonlocal elements of the implemented unitary matrix tend to be close to zero because transitions between inputs and outputs that are far apart have fewer paths (e.g., input 1 and output 8 in Figure 3.1 have a single path). The resulting mesh therefore implements a unitary matrix with a banded structure that is increasingly pronounced as the matrix size increases.

In many applications such as machine learning [225] and quantum computing [212, 42], we avoid this banded unitary matrix behavior in favor of random unitary matrices. A random unitary matrix is achieved when the device phase shifts follow a distribution derived from random matrix theory [103, 277, 57, 232, 212]. In the random matrix theory model, we assign a sensitivity index to each component that increases towards the center of the mesh, as shown in Figure 3.1. The more sensitive components toward the center of the mesh require higher transmissivities and tighter optimization tolerances. If the required tolerances are not met, the implemented unitary matrix begins to show the undesired banded behavior.

In Section 3.2, we introduce the photonic mesh architecture and sources of error that can exacerbate the banded unitary matrix problem. In Section 3.3, we explicitly model the component settings to implement a random unitary matrix and ultimately avoid the banded unitary matrix problem. We propose a ‘‘Haar initialization’’ procedure that allows light to propagate uniformly to all outputs from any input. We use this procedure to initialize the gradient-based optimization of a photonic mesh to learn unknown random unitary matrices given training data. We show that this optimization converges even in the presence of significant simulated fabrication errors.

In Sections 3.4 and 3.5, we propose and simulate two alterations to the mesh architecture that further improve gradient-based optimization performance. First, we add redundant MZIs in the mesh to reduce convergence error by up to five orders of magnitude. Second, we permute the mesh interactions while maintaining the same number of tunable components, which increases allowable tolerances of phase shifters, decreases off-diagonal errors, and improves convergence time.

3.2 Photonic Mesh

We define the photonic mesh when operated perfectly and then discuss how beam splitter or phase shift errors can affect device performance.

3.2.1 Photonic unitary implementation

A single-mode phase shifter can perform an arbitrary $U(1)$ transformation $e^{i\phi}$ on its input. A phase-modulated Mach-Zehnder interferometer (MZI) with perfect (50 : 50) beamsplitters can apply to its

inputs a unitary transformation U of the form:

$$\begin{aligned}
 U(\theta, \phi) &:= R_\phi B R_\theta B \\
 &= \frac{1}{2} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \\
 &= i e^{i\frac{\theta}{2}} \begin{bmatrix} e^{i\phi} \sin \frac{\theta}{2} & e^{i\phi} \cos \frac{\theta}{2} \\ \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \end{bmatrix},
 \end{aligned} \tag{3.1}$$

where B is the beamsplitter operator, R_θ, R_ϕ are upper phase shift operators. Equation 3.1 is represented diagrammatically by the configuration in Figure 3.1.¹ If one or two single-mode phase shifters are added at the inputs, we can apply an arbitrary $SU(2)$ or $U(2)$ transformation to the inputs, respectively.

We define the transmissivity and reflectivity of the MZI as:

$$\begin{aligned}
 t &:= \cos^2 \left(\frac{\theta}{2} \right) = |U_{12}|^2 = |U_{21}|^2 \\
 r &:= \sin^2 \left(\frac{\theta}{2} \right) = 1 - t = |U_{11}|^2 = |U_{22}|^2.
 \end{aligned} \tag{3.2}$$

In this convention, when $\theta = \pi$, we have $r = 1, t = 0$ (the MZI “bar state”), and when $\theta = 0$, we have $r = 0, t = 1$ (the MZI “cross state”).

If there are N input modes and the interferometer is connected to waveguides n and $n+1$ then we can embed the 2×2 unitary U from Equation 3.1 in N -dimensional space with a locally-interacting unitary “Givens rotation” U_n defined as:

$$U_n := \begin{matrix} & & n & n+1 & & & \\ \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & U_{11} & U_{12} & \cdots & 0 \\ 0 & \cdots & U_{21} & U_{22} & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix} & \begin{matrix} n \\ n+1 \end{matrix} & \cdot & \end{matrix} \tag{3.3}$$

All diagonal elements are 1 except those labeled U_{11} and U_{22} , which have magnitudes of $\sqrt{r} = \sqrt{1-t}$, and all off-diagonal elements are 0 except those labeled U_{12} and U_{21} , which have magnitudes of \sqrt{t} .

¹Other configurations with two independent phase shifters between the beamsplitters B are ultimately equivalent for photonic meshes [158].

Arbitrary unitary transformations can be implemented on a photonic chip using only locally interacting MZIs [202]. In this chapter, we focus on optimizing a rectangular mesh [50] of MZIs; however, our ideas can be extended to other universal schemes, such as the triangular mesh [157], as well.

In the rectangular mesh scheme [50] of Figure 3.1, we represent $\hat{U}_R \in U(N)$ in terms of $N(N-1)/2$ locally interacting Givens rotations U_n and N single-mode phase shifts at the inputs represented by diagonal unitary $D(\gamma_1, \gamma_2, \dots, \gamma_N)$:

$$\hat{U}_R := \prod_{\ell=1}^N \prod_{n \in \mathcal{S}_{\ell,N}} U_n(\theta_{n\ell}, \phi_{n\ell}) \cdot D(\gamma_1, \gamma_2, \dots, \gamma_N), \quad (3.4)$$

where our layer-wise product left-multiplies from $\ell = N$ to 1,² the single-mode phase shifts are $\gamma_n \in [0, 2\pi)$, and where the Givens rotations are parameterized by $\theta_{n\ell} \in [0, \pi]$, $\phi_{n\ell} \in [0, 2\pi)$.³ We define the top indices of each interacting mode for each vertical layer as the set $\mathcal{S}_{\ell,N} = \{n \in [1, 2, \dots, N-1] \mid n \pmod{2} \equiv \ell \pmod{2}\}$. This vertical layer definition follows the convention of Refs. [106, 102] and is depicted in Figure 3.1, where ℓ represents the index of the vertical layer.

3.2.2 Beamsplitter error tolerances

The expressions in Equations 3.1 and 3.4 assume perfect fabrication. In practice, however, we would like to simulate how practical devices with errors in each transfer matrix B, R_ϕ, R_θ in Equation 3.1 impact optimization performance.

In fabricated chip technologies, imperfect beamsplitters B can have a split ratio error ϵ that change the behavior of the red 50:50 coupling regions in Figure 3.1 or B in Equation 3.1. The resultant scattering matrix U_ϵ with imperfect beamsplitters B_ϵ can be written as:

$$B_\epsilon := \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{1+\epsilon} & i\sqrt{1-\epsilon} \\ i\sqrt{1-\epsilon} & \sqrt{1+\epsilon} \end{bmatrix} \quad (3.5)$$

$$U_\epsilon := R_\phi B_\epsilon R_\theta B_\epsilon.$$

As shown in Appendix A.1, if we assume both beamsplitters have identical ϵ , we find $t_\epsilon := t(1-\epsilon^2) \in [0, 1-\epsilon^2]$ is the realistic transmissivity, $r_\epsilon := r + t \cdot \epsilon^2 \in [\epsilon^2, 1]$ is the realistic reflectivity, and t, r are the ideal transmissivity and reflectivity defined in Equation 3.2.

The unitary matrices in Equation 3.5 cannot express the full transmissivity range of the MZI, with errors of up to ϵ^2 in the transmissivity, potentially limiting the performance of greedy progressive photonic algorithms [37, 159, 71]. Our Haar phase theory, which we develop in the following section,

²In general, for matrix products for a sequence $\{M_\ell\}$, we define the multiplication order $\prod_{\ell=1}^N M_\ell = M_N M_{N-1} \dots M_1$.

³Since $\gamma_n, \phi_{n\ell}$ are periodic phase parameters, they are in half-open intervals $[0, 2\pi)$. In contrast, any $\theta_{n\ell} \in [0, \pi]$ must be in a closed interval to achieve all transmissivities $t_{n\ell} \in [0, 1]$.

determines acceptable interferometer tolerances for calibration of a “perfect mesh” consisting of imperfect beamsplitters [158] given large N . We will additionally show that simulated photonic backpropagation [102] with adaptive learning can adjust to nearly match the performance of perfect meshes with errors as high as $\epsilon = 0.1$ for meshes of size $N = 128$.

3.2.3 Phase shift tolerances

Another source of uncertainty in photonic meshes is the phase shift tolerances of the mesh which affect the matrices R_θ, R_ϕ of Equation 3.1, shown in orange in Figure 3.1. Error sources such as thermal crosstalk or environmental drift may result in slight deviance of phase shifts in the mesh from intended operation. Such errors primarily affect the control parameters $\theta_{n\ell}$ that control light propagation in the mesh by affecting the MZI split ratios. This nontrivial problem warrants a discussion of mean behavior and sensitivities (i.e., the distribution) of $\theta_{n\ell}$ needed to optimize a random unitary matrix.

3.3 Haar Initialization

3.3.1 Cross state bias and sensitivity index

The convergence of global optimization depends critically on the sensitivity of each phase shift. The gradient descent optimization we study in this chapter converges when the phase shifts are correct to within some acceptable range. This acceptable range can be rigorously defined in terms of average value and variance of phase shifts in the mesh that together define an unbiased (“Haar random”) unitary matrix.⁴ To implement a Haar random unitary, some MZIs in the mesh need to be biased towards cross state ($t_{n\ell}$ near 1, $\theta_{n\ell}$ near 0) [37, 212]. This cross state bias correspondingly “pinches” the acceptable range for transmissivity and phase shift near the limiting cross state configuration, resulting in higher sensitivity, as can be seen in Figure 3.3(b).

For an implemented Haar random unitary matrix, low-tolerance, transmissive MZIs are located towards the center of a rectangular mesh [212, 37] and the apex of a triangular mesh as proven in Appendix A.2. For both the triangular and rectangular meshes, the cross state bias and corresponding sensitivity for each MZI depends only on the total number of reachable waveguides ports, as proven in Appendix A.8. Based on this proof, we define the sensitivity index $\alpha_{n\ell} := |I_{n\ell}| + |O_{n\ell}| - N - 1$,⁵ where $I_{n\ell}$ and $O_{n\ell}$ are the subsets of input and output waveguides reachable by light exiting or entering the MZI, respectively, and $|\cdot|$ denotes set size. Figure 3.1 and Figure 3.2(a) show the sensitivity index for the rectangular mesh, which clearly increases towards the center MZI.

⁴A Haar random unitary is defined as Gram-Schmidt orthogonalization of N standard normal complex vectors [232, 212].

⁵Note that $1 \leq \alpha_{n\ell} \leq N - 1$, and there are always $N - \alpha_{n\ell}$ MZIs that have a sensitivity index of $\alpha_{n\ell}$.

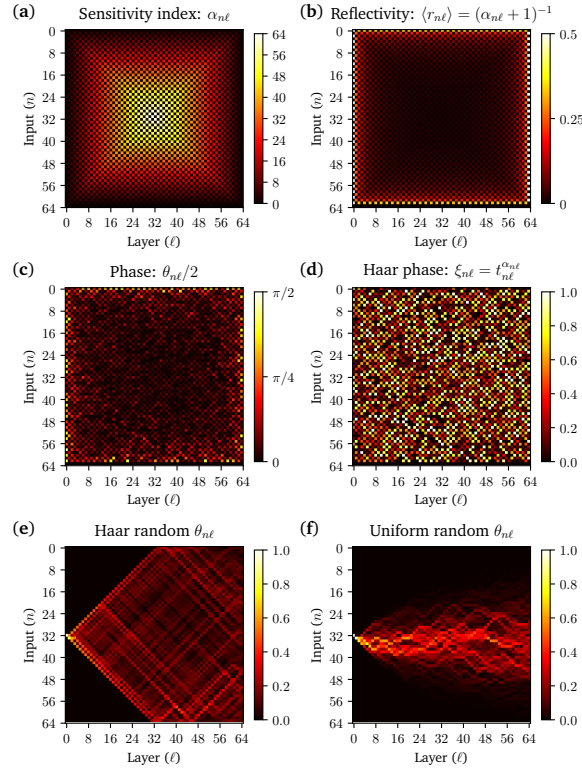


Figure 3.2: (a) The sensitivity index $\alpha_{n\ell}$ for $N = 64$. (b) Checkerboard plot for the average reflectivity $\langle r_{n\ell} \rangle$ in a rectangular mesh. (c) Haar-random matrix and run the decomposition in Ref. [50] to find phases approaching cross state in the middle of the mesh. (d) The Haar phase $\xi_{n\ell}$ for the rectangular mesh better displays the randomness. (e, f) Field measurements (absolute value) from propagation at input 32 in (e) Haar and (f) uniform random initialized rectangular meshes with $N = 64$.

3.3.2 Phase shift distributions and Haar phase

The external $\phi_{n\ell}, \gamma_n$ phase shifts do not affect the the transmissivity $t_{n\ell}$ and therefore obey uniform random distributions [212]. In contrast, the $\theta_{n\ell}$ phase shifts have a probability density function (PDF) that depends on $\alpha_{n\ell}$ [212]:

$$\mathcal{P}_{\alpha_{n\ell}} \left(\frac{\theta_{n\ell}}{2} \right) = \alpha_{n\ell} \sin \left(\frac{\theta_{n\ell}}{2} \right) \left[\cos \left(\frac{\theta_{n\ell}}{2} \right) \right]^{2\alpha_{n\ell}-1}. \quad (3.6)$$

The general shape of this distribution is presented in Figure 3.3(b), showing how an increase in $\alpha_{n\ell}$ biases $\theta_{n\ell}$ towards the cross state with higher sensitivity.

We define the Haar phase $\xi_{n\ell}$ as the cumulative distribution function (CDF) of $\theta_{n\ell}/2$ starting

from $\theta_{n\ell}/2 = \pi/2$:

$$\xi_{n\ell} := \int_{\pi/2}^{\theta_{n\ell}/2} \mathcal{P}_{\alpha_{n\ell}}(\theta) d\theta. \quad (3.7)$$

Using Equations 3.6 and 3.7, we can define $\xi_{n\ell}(\theta_{n\ell}) \in [0, 1]$ that yields a Haar random matrix:

$$\xi_{n\ell} = \left[\cos^2 \left(\frac{\theta_{n\ell}}{2} \right) \right]^{\alpha_{n\ell}} = t_{n\ell}^{\alpha_{n\ell}}, \quad (3.8)$$

where $t_{n\ell}$ represents the transmissivity of the MZI, which is a function of $\theta_{n\ell}$ as defined in Equation 3.2.

3.3.3 Haar initialization

In the physical setting, it is useful to find the inverse of Equation 3.8 to directly set the measurable transmissivity $t_{n\ell}$ of each MZI using a uniformly varying Haar phase $\xi_{n\ell} \sim \mathcal{U}(0, 1)$, a process we call ‘‘Haar initialization’’ shown in Figure 3.2(c, d):

$$\begin{aligned} t_{n\ell} &= \sqrt[\alpha_{n\ell}]{\xi_{n\ell}} \\ \theta_{n\ell} &= 2 \arccos \sqrt{t_{n\ell}} = 2 \arccos \sqrt[\alpha_{n\ell}]{\xi_{n\ell}}, \end{aligned} \quad (3.9)$$

where the expression for $\theta_{n\ell}$ is just a rearrangement of Equation 3.2.

Haar initialization can be achieved progressively using a procedure similar to that in Ref. [159]. If the phase shifters in the mesh are all well-characterized, the transmissivities can be directly set [212]. We will show in Section 3.5 that Haar initialization improves the convergence speed of gradient descent optimization significantly.

We can also use Equation 3.9 to find the average transmissivity and reflectivity for an MZI parameterized by $\alpha_{n\ell}$ as is found through simulation in Ref. [37]:

$$\begin{aligned} \langle t_{n\ell} \rangle &= \int_0^1 d\xi_{n\ell} \sqrt[\alpha_{n\ell}]{\xi_{n\ell}} = \frac{\alpha_{n\ell}}{\alpha_{n\ell} + 1} \\ \langle r_{n\ell} \rangle &= \frac{1}{\alpha_{n\ell} + 1} = \frac{1}{|I_{n\ell}| + |O_{n\ell}| - N}. \end{aligned} \quad (3.10)$$

The average reflectivity $\langle r_{n\ell} \rangle$ shown in Figure 3.2(b) gives a simple interpretation for the sensitivity index shown in Figure 3.2(a). The average reflectivity is equal to the inverse of the total number of inputs and outputs reachable by the MZI minus the number of ports on either side of the device, N . This is true regardless of whether $\alpha_{n\ell}$ is assigned for a triangular or rectangular mesh.

To see what the Haar initialization has accomplished, we can compare the field propagation through the rectangular mesh from a single input when Haar initialized versus uniform initialized in Figure 3.2(e). Physically, this corresponds to light in the mesh spreading out quickly from the input of the mesh and ‘‘interacting’’ more near the boundaries of the mesh (inputs, outputs, top, and

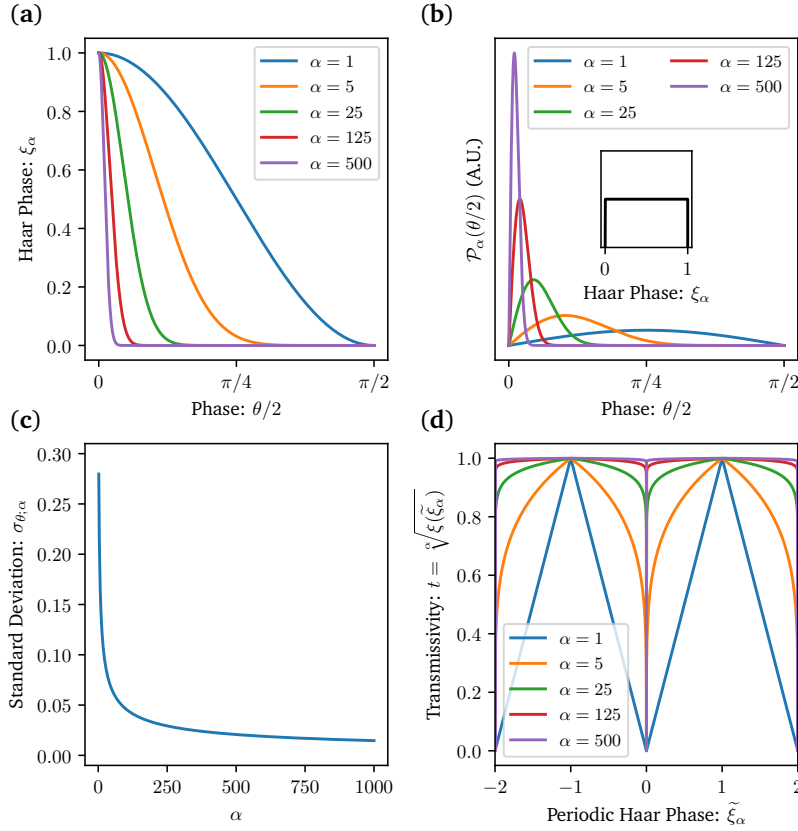


Figure 3.3: (a) Plot of the relationship between ξ_α and θ . (b) We show that phase shift standard deviation $\sigma_{\theta;\alpha}$ decreases as α increases. (c) A plot of $\sigma_{\theta;\alpha}$ as α increases. (d) The transmissivity of an MZI component as a function of a periodic Haar phase has a power law relationship. The periodic Haar phase $\tilde{\xi}_\alpha$ is mapped to the Haar phase by a function $\xi : \mathbb{R} \rightarrow [0, 1]$ as discussed in Appendix A.6.

bottom), as compared to the center of the mesh which has high transmissivity. In contrast, when phases are randomly set, the light effectively follows a random walk through the mesh, resulting in the field propagation pattern shown in Figure 3.2(f).

3.3.4 Tolerance dependence on N

While Haar initialization is based on how the average component reflectivity scales with N , optimization convergence and device robustness ultimately depend on how phase shift tolerances scale with N . The average sensitivity index in the mesh is $\langle \alpha_{n\ell} \rangle = (N + 1)/3$. As shown in Figure 3.3(b, c), the standard deviation $\sigma_{\theta;\alpha}$ over the PDF \mathcal{P}_α decreases as α increases. Therefore, a phase shifter's allowable tolerance, which roughly correlates with $\sigma_{\theta;\alpha}$, decreases as the total number of input and output ports affected by that component increases. Since $\langle \alpha_{n\ell} \rangle$ increases linearly with N ,

the required tolerance gets more restrictive at large N , as shown in Figure 3.3(c). We find that the standard deviation is on the order 10^{-2} radians for most values of N in the specified range. Thus, if thermal crosstalk is ignored [225], it is possible to implement a known random unitary matrix in a photonic mesh assuming perfect operation. However, we concern ourselves with on-chip optimization given just input/output data, in which case the unitary matrix is unknown. In such a case, the decreasing tolerances do pose a challenge in converging to a global optimum as N increases. We demonstrate this problem for $N = 128$ in Section 3.5.

To account for the scalability problem in global optimization, one strategy may be to design a component in such a way that the mesh MZIs can be controlled by Haar phase voltages as in Figure 3.3(d) and Equation 3.9. The transmissivity dependence on a periodic Haar phase (shown in Figure 3.3(d) and discussed in Appendix A.6), is markedly different from the usual sinusoidal dependence on periodic θ_{nl} . The MZIs near the boundary vary in transmissivity over a larger voltage region than the MZIs near the center, where only small voltages are needed get to full transmissivity. This results in an effectively small control tolerance near small voltages. This motivates the modifications to the mesh architecture which we discuss in the next section.

3.4 Architecture Modifications

We propose two architecture modifications that can relax the transmissivity tolerances in the mesh discussed in Section 3.3 and result in significant improvement in optimization.

First, by adding extra tunable MZIs, it is possible to greatly accelerate the optimization of a rectangular mesh to an unknown unitary matrix. The addition of redundant tunable layers to a rectangular mesh is depicted in green in Figure 3.4(a). The authors in Ref. [37] point out that using such “underdetermined meshes” (number of inputs less than the number of tunable layers in the mesh) can overcome photonic errors and restore fidelity in unitary construction algorithms. Adding layers to the mesh increases the overall optical depth of the device, but embedding smaller meshes with extra beamsplitter layers in a rectangular mesh of an acceptable optical depth does not pose intrinsic waveguide loss-related problems.

Another architectural modification to improve training on a rectangular mesh is to shuffle outputs at regular intervals within the mesh. This shuffling relaxes component tolerances and uniformity of the number of paths for each input-output transition. We use this intuition to formally define a permuting rectangular mesh. For simplicity,⁶ assume $N = 2^K$ for some positive integer K . Define “rectangular permutation” operations P_k that allow inputs to interact with waveguides at most 2^k away for $k < K$. These rectangular permutation blocks can be implemented using a rectangular mesh composed of MZIs with fixed cross state phase shifts, as shown in Figure 3.4(b), or using low-loss waveguide crossings.

⁶If N is not a power of 2, then one might consider the following approximate design: $K = \lceil \log_2 N \rceil$. Define $b(K) = \sqrt[K]{N}$, and let each P_k have $\lceil b^k \rceil$ layers.

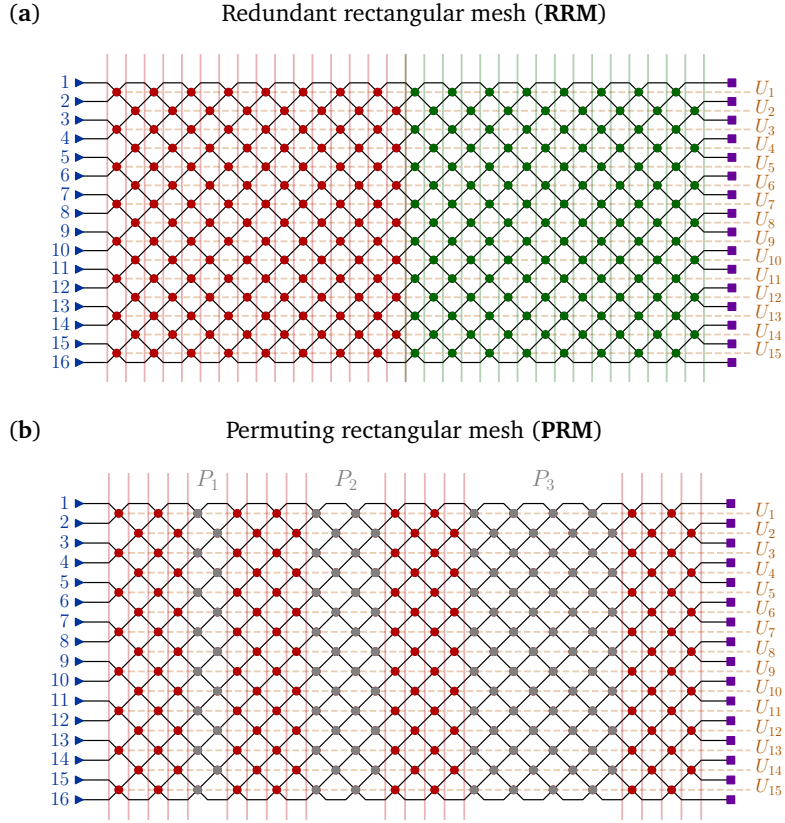


Figure 3.4: (a) A 16×16 rectangular mesh (red). Extra tunable layers (green) may be added to significantly reduce convergence time. (b) A 16-input, 30-layer permuting rectangular mesh. The rectangular permutation layer is implemented using either waveguide crossings or cross state MZIs (gray).

We now add permutation matrices P_1, P_2, \dots, P_{K-1} into the middle of the rectangular mesh as follows:

$$\hat{U}_{\text{PR}} := M_K \left(\prod_{k=1}^{K-1} P_k M_k \right) \quad (3.11)$$

$$M_k := \prod_{\ell=(k-1)\lceil \frac{N}{K} \rceil}^{\min(k\lceil \frac{N}{K} \rceil, N)} \prod_{n \in \mathcal{S}_{\ell, N}} U_n(\theta_{n\ell}, \phi_{n\ell}),$$

where $\lceil x \rceil$ represents the nearest integer larger than x .

There are two operations per block k : an $\lceil \frac{N}{K} \rceil$ -layer rectangular mesh which we abbreviate as M_k , and the rectangular permutation mesh P_k where block index $k \in [1 \cdots K-1]$. This is labelled in Figure 3.4(b).

3.5 Simulations

Now that we have discussed the mesh modifications and Haar initialization, we simulate global optimization to show how our framework can improve convergence performance by up to five orders of magnitude, even in the presence of fabrication error.

3.5.1 Mesh initialization

We begin by discussing the importance of initializing the mesh to respect the cross state bias and sensitivity of each component for Haar random unitary matrices discussed in Section 3.3. Uniform random phase initialization is problematic because it is agnostic of the sensitivity and average behavior of each component. We define this distribution of matrices as $\mathcal{U}_R(N, L)$ for a rectangular mesh for N inputs and L layers. As shown previously in Figure 3.2(f), any given input follows a random walk-like propagation if phases are initialized uniform-randomly, so there will only be non-zero matrix elements within a “bandsize” about the diagonal. This bandsize decreases as circuit size N increases as shown in Figure 3.5.

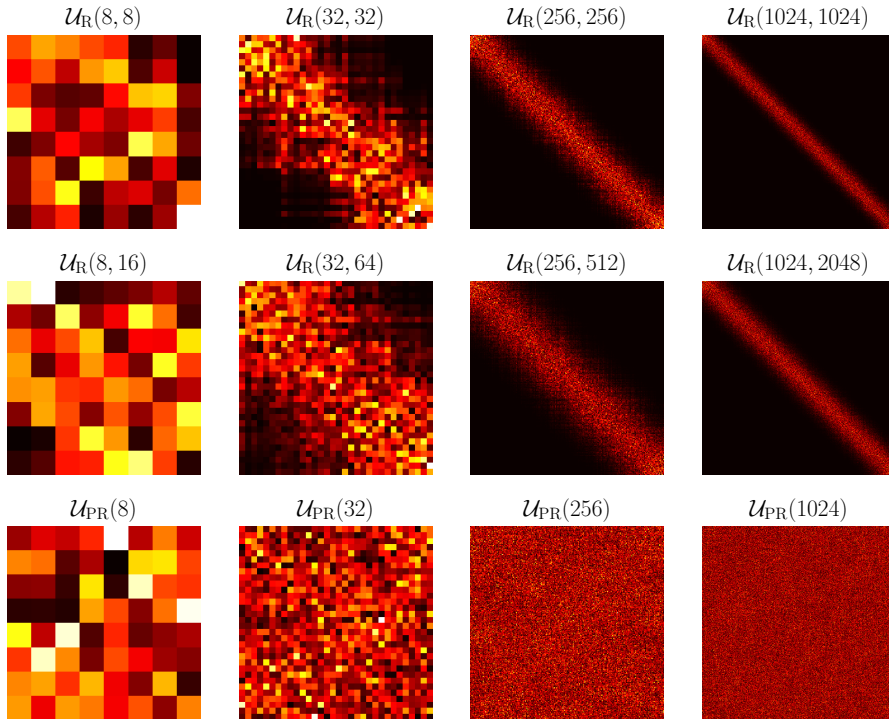


Figure 3.5: Elementwise absolute values of unitary matrices resulting from rectangular ($U \sim \mathcal{U}_R$) and permuting rectangular ($U \sim \mathcal{U}_{PR}$) meshes where meshes are initialized with uniform-random phases.

We compare the bandsizes of banded unitary matrices in simulations qualitatively as we do in Figure 3.5 or quantitatively as we do in Appendix A.3. We randomly generate $U \sim \mathcal{U}_R(N, N)$, $U \sim \mathcal{U}_{PR}(N)$ (permuting rectangular mesh with N tunable layers), and $U \sim \mathcal{U}_R(N, N + \delta N)$ (redundant rectangular mesh with δN extra tunable layers). Figure 3.5 shows a significant reduction in bandsize as N grows larger for rectangular meshes. This phenomenon is not observed with permuting rectangular meshes which generally have the same bandsize as Haar random matrices (independent of N) as shown in in Figure 3.5 and Appendix A.3. This correlates with permuting rectangular meshes having faster optimization and less dependence on initialization.

Instead of initializing the mesh using uniform random phases, we use Haar initialization as in Equation 3.9 to avoid starting with a banded unitary configuration. This initialization, which we recommend for any photonic mesh-based neural network application, dramatically improves convergence because it primes the optimization with the right average behavior for each component. We find in our simulations that as long as the initialization is calibrated towards higher transmissivity ($\theta_{n\ell}$ near 0), larger mesh networks can also have reasonable convergence times similar to when the phases are Haar-initialized.

The proper initialization of permuting rectangular meshes is less clear because the tolerances and average behavior of each component have not yet been modeled. Our proposal is to initialize each tunable block M_k as an independent mesh using the same definition for $\alpha_{n\ell}$, except replacing N with the number of layers in M_k , $\lceil N/K \rceil$. This is what we use as the Haar initialization equivalent in the permuting rectangular mesh case, although it is possible there may be better initialization strategies for the nonlocal mesh structure.

3.5.2 Optimization problem and synthetic data

After initializing the photonic mesh, we proceed to optimize the mean square error cost function for an unknown Haar random unitary U :

$$\underset{\theta_{n\ell}, \phi_{n\ell}, \gamma_n}{\text{minimize}} \quad \frac{1}{2N} \left\| \hat{U}(\theta_{n\ell}, \phi_{n\ell}, \gamma_n) - U \right\|_F^2, \quad (3.12)$$

where the estimated unitary matrix function \hat{U} maps N^2 phase shift parameters $\theta_{n\ell}, \phi_{n\ell}, \gamma_n$ to $U(N)$ via Equations 3.4 or 3.11, and $\|\cdot\|_F$ denotes the Frobenius norm. Since trigonometric functions parameterizing \hat{U} are non-convex, we know that Equation 3.12 is a non-convex problem. The non-convexity of Equation 3.12 suggests learning a single unitary transformation in a deep neural network might have significant dependence on initialization.

To train the network, we generate random unit-norm complex input vectors of size N and generate corresponding labels by multiplying them by the target matrix U . We use a training batch size of $2N$. The synthetic training data of unit-norm complex vectors is therefore represented by $X \in \mathbb{C}^{N \times 2N}$. The minibatch training cost function is similar to the test cost function, $\mathcal{L}_{\text{train}} = \|\hat{U}X - UX\|_F^2$.

The test set is the identity matrix I of size $N \times N$. The test cost function, in accordance with the training cost function definition, thus matches Equation 3.12.

3.5.3 Training algorithm

We simulate the global optimization of a unitary mesh using automatic differentiation in `tensorflow`, which can be physically realized using the *in situ* backpropagation procedure in Ref. [102]. This optical backpropagation procedure physically measures $\partial\mathcal{L}_{\text{train}}/\partial\theta_{n\ell}$ using interferometric techniques, which can be extended to any of the architectures we discuss in this chapter.

The on-chip backpropagation approach is also likely faster for gradient computation than other training approaches such as the finite difference method mentioned in past on-chip training proposals [225]. We find empirically that the Adam update rule (a popular first-order adaptive update rule [118]) outperforms standard stochastic gradient descent for the training of unitary networks. If gradient measurements for the phase shifts are stored during training, adaptive update rules can be applied using successive gradient measurements for each tunable component in the mesh. Such a procedure requires minimal computation (i.e., locally storing the previous gradient step) and can act as a physical test of the simulations we will now discuss. Furthermore, we avoid quasi-Newton optimization methods such as L-BFGS used in Ref. [37] that cannot be implemented physically as straightforwardly as first-order methods.

The models were trained using our open source simulation framework `neurophox`⁷ using a more general version of the vertical layer definition proposed in Refs. [106, 102]. The models were programmed in `tensorflow` [2] and run on an NVIDIA GeForce GTX1080 GPU to improve optimization performance.

3.5.4 Architecture comparison

We now compare training results for rectangular, redundant rectangular, and permuting rectangular meshes given $N = 128$. In our comparison of permuting rectangular meshes and rectangular meshes, we analyze performance when beamsplitter errors are distributed throughout the mesh as either $\epsilon = 0$ or $\epsilon \sim \mathcal{N}(0, 0.01)$ and when the $\theta_{n\ell}$ are randomly or Haar-initialized (according to the PDF in Equation 3.6). We also analyze optimization performances of redundant rectangular meshes where we vary the number of vertical MZI layers.

From our results, we report five key findings:

1. Optimization of $N = 128$ rectangular meshes results in significant off-diagonal errors due to bias towards the banded matrix space of $\mathcal{U}_R(128)$, as shown in Figure 3.7.

⁷To reproduce the results of this chapter, the reader can be directed to `neurophox`, an open-source Python package that implements the optimizations and simulations of this paper in `numpy` and `tensorflow`. The exact code used to generate the results is provided in the `neurophox-notebooks` repository.

2. Rectangular meshes converge faster when Haar-initialized than when uniformly random initialized, as in Figure 3.7, in which case the estimated matrix converges towards a banded configuration shown in Appendix A.7.
3. Permuting rectangular meshes converge faster than rectangular meshes despite having the same number of total parameters as shown in Figure 3.7.
4. Redundant rectangular meshes, due to increase in the number of parameters, have up to five orders of magnitude better convergence when the number of vertical layers are doubled compared to rectangular and permuting rectangular meshes, as shown in Figure 3.6.
5. Beamsplitter imperfections slightly reduce the overall optimization performance of permuting and redundant rectangular meshes, but reduce the performance of the rectangular mesh significantly. (See Figure 3.7 and Appendix A.4.)

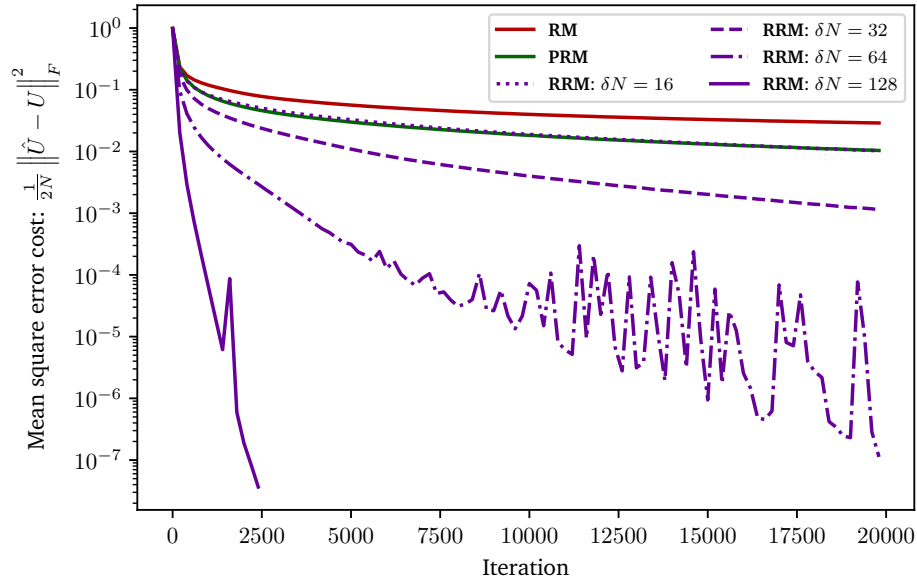


Figure 3.6: A comparison of test error in `tensorflow` for $N = 128$ between rectangular (RM), permuting rectangular (PRM), and redundant rectangular (RRM) meshes for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256. Ideal = Haar random initialized $\theta_{n\ell}$ with $\epsilon = 0$. δN is the additional layers added in the redundant mesh. We stopped the $\delta N = 128$ run within 4000 iterations when it reached convergence within machine precision. Redundant meshes with 32 additional layers converge better than permuting rectangular meshes, and with just 16 additional layers, we get almost identical performance.

The singular value decomposition (SVD) architecture discussed in Refs. [157, 225] consists of optical lossy components flanked on both sides by rectangular meshes and are capable of implementing any linear operation with reasonable device input power. Note that with some modifications (e.g.

treating loss and gain elements like nonlinearities in the procedure of Ref. [102]), SVD architectures can also be trained physically using *in situ* backpropagation. We simulate the gradient-based optimization of SVD architectures using automatic differentiation in Appendix A.5.

3.6 Discussion

3.6.1 Haar initialization

For global optimization and robustness of universal photonic meshes, it is important to consider the required biases and sensitivities for each mesh component. Implementing any Haar random matrix requires that each component independently follows an average reflectivity within some tolerance. This requirement becomes more restrictive with the number of input and output ports accessible by each mesh component. For the rectangular mesh, this means the center mesh components are close to cross state and the most sensitive.

In a Haar-initialized mesh, as shown in Figure 3.2, the light injected into a single input port spreads out to all waveguides in the device uniformly regardless of N . This is a preferable initialization for global optimization because Haar random matrices require this behavior. In contrast, when randomly initializing phases, the light only spreads out over a limited band of outputs. This band gets relatively small compared to the mesh gets larger as shown in Figure A.2.

The average reflectivities given by Haar initialization may be useful for inverse design approaches [189] for compact tunable or passive multiport interferometers. The component tolerances may inform how robust phase shifters need to be given error sources such as thermal crosstalk [225]. The thermal crosstalk might make it difficult to achieve required tolerances for devices interfering up to $N = 1000$ modes that generally have phase shift tolerances just above 10^{-2} radians.⁸

In our simulations in Section 3.5, we assume that the control parameter for photonic meshes is linearly related to the phase shift. However, in many current phase shifter implementations, such as thermal phase shifters [225], the phase is a nonlinear function of the control parameter (i.e., the voltage) and has minimum and maximum values, unlike the unbounded phase used in our optimization. In addition, like the Haar phase in our theory, the voltage acts as the CDF for transmissivities in the physical device, up to a normalization factor. Particular attention needs to be given to phase uncertainty as a function of voltage, since the Haar random distribution of internal MZI phases has small variance for large N , as we show in Figure 3.3(c). As mentioned in Section 3.3, the ideal transmissivity-voltage dependence with this consideration would be identical to the transmissivity vs Haar phase dependence in Figure 3.3(d).

⁸Ref. [225] propose a standard deviation of $\sim 10^{-3}$ might be possible with further circuit characterization, which might be scalable based on Figure 3.3(c).

3.6.2 Applications of mesh optimization

Meshes can be tuned using either self-configuration [202, 157] or global optimizations (gradient-based [102] or derivative-free [240]). The algorithmic optimizations proposed in Refs. [202, 157] assume that each component in the mesh can cover the entire split ratio range, which is not the case in presence of 50:50 beamsplitter errors. This ultimately leads to lower fidelity in the implemented unitary operation, which can be avoided using a double-MZI architecture [158, 256] or a vertical layer-wise progressive algorithm [159]. We explore a third alternative to overcome photonic errors; gradient-based global optimization is model-free and, unlike algorithmic approaches, can efficiently tune photonic neural networks [102]. This model-free property makes gradient-based optimization robust to fabrication error; we show in Figure 3.7(a) that meshes with split ratio error variances of up to $\sigma_\epsilon = 0.1$ can be optimized nearly as well as a perfect mesh, particularly for permuting rectangular meshes.

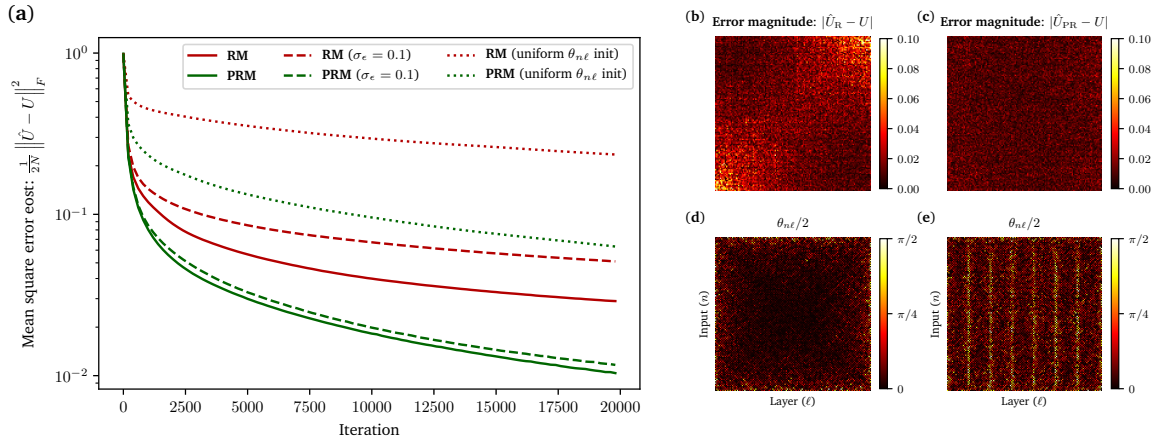


Figure 3.7: We implement six different optimizations for $N = 128$ where we vary the choice of permuting rectangular mesh (PRM) or rectangular mesh (RM); the initialization (random $\theta_{n\ell}$ or Haar-initialized $\theta_{n\ell}$); and photonic transmissivity error displacements ($\epsilon = 0$ or $\epsilon \sim \mathcal{N}(0, 0.01)$, where $\sigma_\epsilon^2 = 0.01$ is the variance of the beamsplitter errors). Conditions: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256, simulated in `tensorflow`. (a) Comparison of optimization performance (defaults are Haar initialization and $\epsilon_{n\ell} = 0$ unless otherwise indicated). Optimized error magnitude spatial map for (b) rectangular mesh shows higher off-diagonal errors and than (c) permuting rectangular. The optimized $\theta_{n\ell}$ phase shifts (see Appendix A.6) for (d) rectangular meshes are close to zero (cross state) near the center as opposed to (e) permuting rectangular meshes which have a striped pattern (likely due to initialization). NOTE: by $|\cdot|$, we refer to the elementwise norm.

In the regime of globally optimized meshes, we propose two strategies to modify the rectangular architecture: adding waveguide permutation layers and adding extra tunable vertical MZI layers. Both approaches relax the cross state requirements on the MZIs and accelerate the mesh optimization process. Nonlocal interference works by allowing inputs that are far away physically in the mesh to

interact. These approaches are inspired by several recent proposals in machine learning and coherent photonics to design more error tolerant and efficient meshes, many of which use single layers of MZIs and nonlocal waveguide interactions [75, 151, 71, 106]; such designs can also be considered to be in the same class of permuting architectures as our proposed permuting rectangular mesh. Adding extra tunable vertical layers, as proposed in Ref. [37], simply adds more tunable paths for the light to achieve a desired output. As shown in Figure 3.7, we achieve up to five orders of magnitude improvement in convergence at the expense of doubling the mesh size and parameter space.

Like permuting rectangular meshes, multi-plane light conversion successfully applies the non-local interference idea for efficient spatial mode multiplexing [127, 128]. In this protocol, alternating layers of transverse phase profiles and optical Fourier transforms (analogous to what our rectangular permutations accomplish) are applied to reshape input modes of light [127, 128]. A similar concept is used in unitary spatial mode manipulation, where stochastic optimization of deformable mirror settings allow for efficient mode conversion [163]. Thus, the idea of efficient unitary learning via a Fourier-inspired permuting approach has precedent in contexts outside of photonic MZI meshes.

An on-chip optimization for multi-plane light conversion has been accomplished experimentally in the past using simulated annealing [240]. The success of simulated annealing in experimentally training small unitary photonic devices [240] (rather than gradient descent as is used in this chapter) suggests there are other algorithms aside from gradient descent that may effectively enable on-chip training.

We propose that similar simulated annealing approaches might be made more efficient by sampling Haar phases from uniform distributions and flashing updates onto the device. Similar derivative-free optimizations may also be useful for quantum machine learning [216, 215, 113]. Whether such approaches can compete with backpropagation for classical applications remains to be investigated. For experimental on-chip tuning, simulated annealing has the attractive property of only requiring output detectors. For practical machine learning applications, however, there is currently more literature for backpropagation-based optimization. Furthermore, gradient-based approaches allow for continuous control of phase shifters during the optimization.

Our `tensorflow` simulations may be useful in the design of optical recurrent neural networks (RNNs) that use unitary operators parameterized by photonic meshes. Such “unitary RNNs” (URNNs) have already been simulated on conventional computers and show some promise in synthetic long-term memory tasks [106, 56]. Unitary RNNs are physically implementable using a single mesh with optical nonlinearities and recurrent optoelectronic feedback, suggesting that the architecture discussed in this chapter is a scalable, energy-efficient option for machine learning applications. It is possible that some tunable features such as the “bandedness” of unitaries implemented by rectangular MZI meshes can be useful (e.g. as an attention mechanism in sequence data) for certain deep learning tasks that use URNNs.

3.7 Conclusion

The scalability of gradient-based optimization of Haar random unitary matrices on universal photonic meshes is limited by small reflectivities and MZI phase shifter sensitivities arising from the constraint of locally interacting components. As shown in Section 3.3, the required average reflectivity and sensitivity for each MZI is inversely related to the total number of inputs and outputs affected by the MZI. If the tolerance requirements are not met by the physical components, optimization algorithms will have difficulty converging to a target unitary operator. As shown in Section 3.5 for the case of $N = 128$, convergence via *in situ* backpropagation is generally not achieved if phase shifters are initialized randomly. However, Haar initialization can sufficiently bias the optimization for convergence to a desired random unitary matrix, even in the presence of significant simulated beamsplitter fabrication errors.

In Section 3.4, we propose adding extra tunable beamsplitters or mesh nonlocalities to accelerate mesh optimization. Naive (uniform random) initialization on a standard photonic mesh has difficulty learning random unitary matrices via gradient descent. By introducing non-localities in the mesh, we can improve optimization performance without the need for extra parameters. A Haar-initialized redundant architecture can achieve five orders of magnitude less mean square error for a Haar random unitary matrix and decrease optimization time to such a matrix by at least two orders of magnitude, as shown in Figure 3.6. Our findings suggest that architecture choice and initialization of photonic mesh components may prove important for increasing the scalability and stability of reconfigurable universal photonic devices and their many classical and quantum applications [9, 225, 233, 113, 215, 216, 11, 157, 156].

Part II

Quantum information processing using light

Chapter 4

Photonic quantum programmable gate arrays

In this chapter, we present a photonic integrated circuit architecture for a quantum programmable gate array (QPGA) capable of preparing arbitrary quantum states and operators [21]. The architecture consists of a lattice of phase-modulated Mach-Zehnder interferometers, which perform rotations on path-encoded photonic qubits, and embedded quantum emitters, which use a two-photon scattering process to implement a deterministic controlled- σ_z operation between adjacent qubits. By appropriately setting phase shifts within the lattice, the device can be programmed to implement any quantum circuit without hardware modifications. We provide algorithms for exactly preparing arbitrary quantum states and operators on the device and we show that gradient-based optimization can train a simulated QPGA to automatically implement highly compact approximations to important quantum circuits with near-unity fidelity.

4.1 Introduction

There has been growing interest in universal photonic devices which can be dynamically reconfigured to implement any linear optical transformation to a set of coherent optical modes. [202, 157, 50, 92] These devices are often implemented as a mesh of phase-modulated Mach-Zehnder interferometers (MZIs) which can be configured progressively [202] or simultaneously [178] to apply arbitrary unitary transformations to an input vector of spatial modes. Such devices have a wide range of applications in classical information processing [92, 160, 9, 276, 186, 185], and integrated universal photonic circuits provides an especially promising hardware platform for high-throughput, energy-efficient machine learning. [225, 100, 177, 259]

These devices also have promising applications in quantum information processing: recent demonstrations of boson sampling [233], quantum transport dynamics [93], photonic quantum walks [83], counterfactual communication [8], and probabilistic two-photon gates [42] have all been performed on this type of programmable photonic hardware. Photonic systems offer a range of unique advantages over other substrates for quantum information processing: optical quantum states have long coherence times and can be maintained at room temperature, since they interact very weakly with their environment; photonic qubits are optimal information carriers for distant nodes within quantum networks; and MZIs provide simple, high-fidelity implementations of single-qubit operations which can be integrated into a photonic chip.

However, photonic quantum computation poses several intrinsic difficulties. The non-interacting nature of photons makes implementing deterministic multi-photon quantum gates a challenge; many existing proposals [123] and demonstrations [42] of linear optical quantum computing rely on non-deterministic “heralded” gates, or encode multi-qubit quantum states in exponentially many spatial modes [212]. Since photons must propagate at the speed of light, photonic quantum processing must be done along the path of the photon by sequential optical components, making complex quantum circuits prohibitively large to implement with free-space optics. These systems and even some integrated photonic circuits also often suffer from a lack of reconfigurability, as the design of task-specific optical circuitry must be modified to perform different computations. [239]

Here we describe a photonic lattice architecture for a reconfigurable and universal quantum programmable gate array (QPGA) which can implement any quantum operation, in principle deterministically and with perfect fidelity in the case of ideal physical components. Our design is similar to a universal linear optical component [157], but employs nonlinear interactions from precisely placed quantum emitters to enable an N -qubit state to be encoded using $\mathcal{O}(N)$ number of spatial modes. The proposed device can be programmed to implement any quantum circuit decomposed into one- and two-qubit gates performed by physical lattice components on an integrated photonic circuit. Phase-modulated MZIs apply arbitrary single-qubit operations to qubits which are path-encoded by single photons in a superposition of pairs of waveguides and two-photon scattering processes induced by strongly-coupled quantum emitters implement controlled gates between adjacent qubits.

We provide exact algorithms in Section 4.3 for obtaining the appropriate phase shifter parameters to prepare arbitrary quantum states and operators on-chip. In Section 4.4, we discuss how optimization techniques from machine learning can be used to automatically discover high-fidelity approximations to desired quantum operations which are significantly more compact than their explicitly-decomposed exact representations.

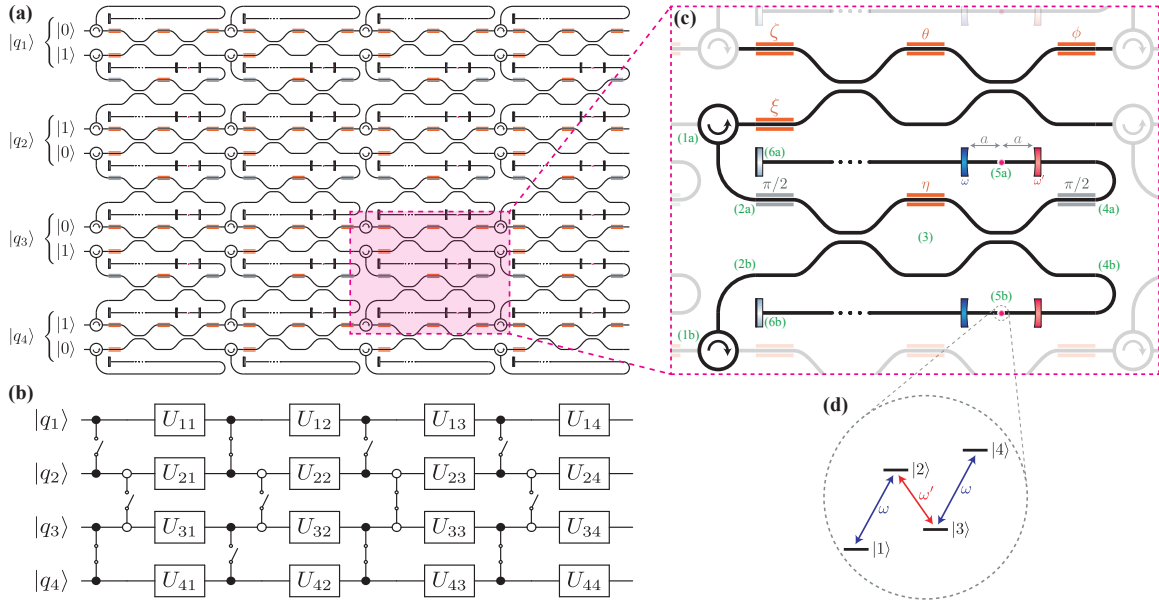


Figure 4.1: The architecture for the quantum programmable gate array shown at various levels of detail. (a) Physical layout of a four-qubit QPGA with a depth of four layers. Each logical qubit is path-encoded by a single photon in a pair of waveguides, with the parity of which waveguide represents $|0\rangle$ and $|1\rangle$ depending on the parity of the qubit index. (b) A quantum circuit diagram depicting the logical representation of the operator performed by the QPGA in the first panel. The “switch” symbols between two-qubit operations indicate that the connectivity of the gates can be reconfigured without changing the physical chip architecture. Solid control dots indicate $c\sigma_z$, while open dots indicate $\bar{c}\sigma_z$. (c) A single unit cell within the lattice. The ζ, ξ, θ, ϕ phase shifters are continuously variable trainable parameters, while $\eta = 0, \frac{\pi}{2}$ determines the connectivity of the $c\sigma_z$ gates between neighboring qubits. The pink dots represent quantum emitters embedded a distance a between two dichroic reflectors, depicted as blue and red rectangles, which selectively reflect light at frequencies ω and ω' , respectively. The delay lines are matched in length to ω' and terminate in reflectors. (d) Four-level energy structure of the quantum emitters embedded in the waveguides.

4.2 Photonic quantum programmable gate arrays

The concept for a photonic quantum programmable gate array is shown in Fig. 4.1a, and the equivalent logical quantum circuit is depicted in Fig. 4.1b. The architecture consists of a set of waveguide pairs which each contain single photon pulses. A lattice of phase-modulated MZIs perform single-qubit rotations, and circulators, MZIs, and embedded four-level systems (4LS) collectively implement two-qubit controlled- σ_z ($c\sigma_z$) gates between adjacent qubits. By choosing suitable phase shifter parameters, arbitrary multi-qubit quantum states and operators can be implemented from single-qubit and $c\sigma_z$ primitives within the lattice, as discussed in Section 4.3. In the following subsections, we discuss the mechanisms of each component of the architecture in greater detail.

4.2.1 Single-qubit operations

Qubits are implemented as temporally separated single photons, each injected into a pair of waveguides at a frequency ω and with a long pulse length $\tau \gg \omega^{-1}$. All physical gates within the device conserve photon occupancy within waveguide pairs. A chip designed to process N -qubit states has $2N$ number of waveguides, and the computational basis $\{|0\rangle, |1\rangle\}$ of each qubit is represented by the photon occupancy of the top and bottom waveguide in each pair, with parity alternating with qubit index as shown in Fig. 4.1a.

Single-qubit gates are implemented with a standard approach using phase-modulated MZIs. An MZI with four phase shifters in the configuration shown in the upper half of Fig. 4.1c can apply any operation $U \in U(2)$ to its inputs, which suffices to implement arbitrary single-qubit gates. [123, 42, 202] Assuming the photons are spectrally narrow about ω (see Appendix B.1 for a more complete treatment of arbitrary photon spectra), the transformation implemented by the MZI on the input modes takes the form:

$$\begin{aligned}
 U(\zeta, \xi, \theta, \phi) &= R_{\xi}^{\zeta} H R_{\phi_0}^{\theta} H R_0^{\phi} \\
 &= \frac{1}{2} \begin{pmatrix} e^{i\zeta} & 0 \\ 0 & e^{i\xi} \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{pmatrix} \\
 &= \frac{1}{2} \begin{bmatrix} e^{i(\zeta+\phi)} (e^{i\theta} + 1) & e^{i(\xi+\phi)} (e^{i\theta} - 1) \\ e^{i\zeta} (e^{i\theta} - 1) & e^{i\xi} (e^{i\theta} + 1) \end{bmatrix},
 \end{aligned} \tag{4.1}$$

where H is the Hadamard operator¹ and $R_{\phi_2}^{\phi_1}$ denotes a phase shift of ϕ_1 applied to the top waveguide and ϕ_2 to the bottom. Here, and for the rest of this paper, successive matrices are left-multiplied to be consistent with circuit diagrams.

4.2.2 Two-photon gates

In addition to arbitrary single-qubit gates, the QPGA needs to be able to implement two-qubit entangling operations in order to be a universal quantum device. This is accomplished by nonlinear interactions between two photons scattering off of a pair of quantum emitters embedded within the waveguides. The emitters could be implemented by quantum dots coupled to photonic crystal waveguides [115, 132, 231] or plasmonic nanowires [7], diamond vacancy centers [199, 271, 15], or many other experimental setups. There have been many proposals for implementing two-qubit gates using scattering-based processes [59, 222, 49, 252, 54]; the scattering dynamics discussed in this section are adapted from the scheme described by Zheng et al. [274], with the notable difference that spatial modes rather than momentum states form the computational basis for the physical qubits. In this section we show that this scattering process implements a $c\sigma_z$ operation up to local

¹Whether to use $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$ or $B = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$ to represent the beamsplitter operation is somewhat a matter of convention, with classical optics tending to prefer the latter and quantum information often using the former. They are equivalent up to a phase shift of ζ, θ by $\pi/2$.

phase shifts.²

Consider an arbitrary two-qubit logical input state $|\Psi\rangle = \alpha|11\rangle + \beta|10\rangle + \gamma|01\rangle + \delta|00\rangle$. The state consists of two photons superpositioned over two pairs of waveguides shown in Figure 4.1. Define bosonic operators $\hat{a}_{0,d}^{[q]\dagger}, \hat{a}_{1,d}^{[q]\dagger}$ which create a photon for qubit q with direction $d \in \{L, R\}$ in the $|0\rangle$ and $|1\rangle$ waveguide, respectively. The corresponding two-photon physical input state $|\psi\rangle$ just before (1a, 1b) is:

$$|\psi\rangle = \alpha \hat{a}_{1,R}^{[1]\dagger} \hat{a}_{1,R}^{[2]\dagger} |\emptyset\rangle + \beta \hat{a}_{1,R}^{[1]\dagger} \hat{a}_{0,R}^{[2]\dagger} |\emptyset\rangle + \gamma \hat{a}_{0,R}^{[1]\dagger} \hat{a}_{1,R}^{[2]\dagger} |\emptyset\rangle + \delta \hat{a}_{0,R}^{[1]\dagger} \hat{a}_{0,R}^{[2]\dagger} |\emptyset\rangle, \quad (4.2)$$

where $|\emptyset\rangle$ denotes the vacuum state (not to be confused with the computational $|0\rangle$ state).

Consider the lower half of Figure 4.1c. Two circulators at (1a, 1b) direct the $|1\rangle$ modal component of each photon into the waveguides at (2a, 2b). The photons pass through an MZI at (3) which has a transfer matrix³:

$$T(\eta) = R_0^{\pi/2} H R_0^\eta H R_0^{\pi/2} = \frac{1}{2} \begin{pmatrix} -e^{i\eta} - 1 & ie^{i\eta} - i \\ ie^{i\eta} - i & e^{i\eta} + 1 \end{pmatrix} \quad (4.3)$$

Define bosonic operators $\hat{b}_d^{\text{top}\dagger}, \hat{b}_d^{\text{bot}\dagger}$, which create a photon with frequency ω in direction d at (4a, 4b), respectively.⁴ The transfer matrix acts only on the $|1\rangle$ component of each photon, so we can relate the operators:

$$\begin{pmatrix} \hat{a}_{1,R}^{[1]\dagger} \\ \hat{a}_{1,R}^{[2]\dagger} \end{pmatrix} = T(\eta) \begin{pmatrix} \hat{b}_R^{\text{top}\dagger} \\ \hat{b}_R^{\text{bot}\dagger} \end{pmatrix}, \quad \begin{pmatrix} \hat{a}_{1,L}^{[1]\dagger} \\ \hat{a}_{1,L}^{[2]\dagger} \end{pmatrix} = T^\top(\eta) \begin{pmatrix} \hat{b}_L^{\text{top}\dagger} \\ \hat{b}_L^{\text{bot}\dagger} \end{pmatrix}, \quad (4.4)$$

while $\hat{a}_{0,d}^{[q]\dagger}$ are unaffected. Using the relations described in Eq. 4.4, the input state after propagating through the MZI at (4a, 4b) is:

$$\begin{aligned} |\psi\rangle &= e^{i\eta} \frac{\alpha}{2} \sin \eta \left((\hat{b}_R^{\text{top}\dagger})^2 - (\hat{b}_R^{\text{bot}\dagger})^2 \right) |\emptyset\rangle \\ &\quad - e^{i\eta} \alpha \cos \eta \hat{b}_R^{\text{top}\dagger} \hat{b}_R^{\text{bot}\dagger} |\emptyset\rangle \\ &\quad - e^{i\eta/2} \left(\beta \cos \frac{\eta}{2} \hat{a}_{0,R}^{[2]\dagger} + \gamma \sin \frac{\eta}{2} \hat{a}_{0,R}^{[1]\dagger} \right) \hat{b}_R^{\text{top}\dagger} |\emptyset\rangle \\ &\quad - e^{i\eta/2} \left(\beta \sin \frac{\eta}{2} \hat{a}_{0,R}^{[2]\dagger} - \gamma \cos \frac{\eta}{2} \hat{a}_{0,R}^{[1]\dagger} \right) \hat{b}_R^{\text{bot}\dagger} |\emptyset\rangle \\ &\quad + \delta \hat{a}_{0,R}^{[1]\dagger} \hat{a}_{0,R}^{[2]\dagger} |\emptyset\rangle. \end{aligned} \quad (4.5)$$

²Lattice cells of inverted parity (see Figure 4.1a) actually implement $\overline{c\sigma_z}$, such that σ_z is applied only to $|00\rangle$, but the dynamics are the same, so for brevity we discuss only one parity here.

³The $\pi/2$ phase shifts are necessary to conserve photon number within each waveguide pair by making the round-trip transfer matrix $T(\eta)^\top T(\eta)$ diagonal.

⁴The $\hat{b}^{\text{top}\dagger}, \hat{b}^{\text{bot}\dagger}$ notation was chosen to avoid confusion with the qubit indices or basis states $\hat{a}_{\{0,1\}}^{[q]\dagger}$.

The \hat{b}^\dagger photons propagate down the waveguides from (4a, 4b) until they interact with the embedded quantum emitters at (5a, 5b), while we assume the system acts trivially on the $\hat{a}_{0,R}^{[q]\dagger}$ photons.

We now consider the sections between (4a) to (6a) and (4b) to (6b). We will show that the two-photon state, upon passing through these sections, will gain a π phase shift applied only to the first term of $|\psi\rangle$ in Eq. 4.5, and thus a $c\sigma_z$ operation is implemented on the input state $|\Psi\rangle$. To show this, we first consider the dynamics of the photons in the section between sites (4a) to (6a) within a single isolated waveguide; the lower waveguide between (4b) and (6b) behaves identically. For simplicity, while we consider each waveguide in isolation, we drop the $\hat{b}^{\{\text{top},\text{bot}\}}$ superscripts and omit the $\hat{a}_{0,R}^{[q]\dagger}$ operators.

The regions of interest are shown in the middle of Fig. 4.1c, which contain quantum emitters with the four-level energy structures shown in Fig. 4.1d. The energy level of each state $|i\rangle$ is Ω_i ; we assume that $\Omega_4 - \Omega_3 = \Omega_2 - \Omega_1 = \omega$, and denote $\omega' \equiv \Omega_3 - \Omega_2$. The quantum emitters at (5a, 5b) are placed a distance a between a pair of narrow-band filters, which are reflective at frequencies ω and ω' , respectively, and transparent otherwise. Reflectors terminate the ends of the waveguides at (6a, 6b); the waveguides between the ω' filters and the reflectors form a delay line with a length which is a multiple of $\frac{2\pi}{\omega'}$. The real-space Hamiltonian that describes the coupling of such an atom to the waveguide without the filters is given by [207, 221, 274]:

$$\begin{aligned} \mathcal{H} = & \frac{\hbar}{i} \int dx \left[v_g \hat{b}_R^\dagger(x) \frac{\partial}{\partial x} \hat{b}_R(x) - v_g \hat{b}_L^\dagger(x) \frac{\partial}{\partial x} \hat{b}_L(x) + v_r \hat{c}^\dagger(x) \frac{\partial}{\partial x} \hat{c}(x) \right] + \hbar \sum_{n=1}^4 \Omega_n |n\rangle\langle n| \\ & + \hbar \int dx \delta(x) \left[\left(\sqrt{\frac{\Gamma v_g}{2}} \hat{b}_R^\dagger(x) + \sqrt{\frac{\Gamma v_g}{2}} \hat{b}_L^\dagger(x) + \sqrt{\Gamma' v_r} \hat{c}^\dagger(x) \right) (|1\rangle\langle 2| + |3\rangle\langle 2| + |3\rangle\langle 4|) + \text{H.c.} \right]. \end{aligned} \quad (4.6)$$

Here, the first term describes the free waveguide dynamics, the second term describes the embedded four-level system shown in Fig. 4.1d, and the third term is the interaction Hamiltonian. The decay rate into the waveguide is Γ , the coupling Γ' describes the extrinsic loss of the excited states to degrees of freedom outside the waveguide, which is modeled as emission into a reservoir by the \hat{c}^\dagger, \hat{c} operators, and $v_g \{v_r\}$ is the group velocity of the photons in the waveguide {reservoir}. The transition frequencies ω, ω' obey $|\omega - \omega'| \gg \Gamma$.

The scattering dynamics can be summarized by four steps occurring simultaneously in the top and bottom waveguides. (1) Photon A at frequency ω causes the atom, which is initialized in state $|1\rangle$, to partially transition from $|1\rangle \rightarrow |3\rangle$ with an amplitude of $|3\rangle$ corresponding to the photon occupancy in the waveguide. This emits an auxiliary photon A' with frequency ω' , which is reflected by one of the narrow-band mirrors and travels down the delay line. (2) While photon A' is in the delay line, photon B , also at frequency ω , is injected into the system. Interaction with the $|1\rangle$ component of the atomic states results in the transition $|1\rangle \rightarrow |3\rangle$ and releases an auxiliary photon B' with frequency ω' down the delay line, while interaction with the $|3\rangle$ component imparts a π

phase shift onto B and reflects it back into the waveguide. (3) Photon A' arrives back at the 4LS after traversing the delay line. By time reversal arguments, sending the output photon A' back into the atom retrieves photon A , which exits the inner cell through its original waveguide. (4) Photon B' arrives back at the 4LS, retrieving photon B as in step 3.

A conceptual animation depicting the two-photon scattering process in a QPGA cell can be found in the supplementary materials. We now discuss each step in greater detail. Derivations of the reflection coefficients and output states can be found in Appendix B.2.

Step 1. At time $t = 1$, photon A with frequency ω and state $|\psi_1^{\text{in}}\rangle = \alpha_A |\omega\rangle + \beta_A |\emptyset\rangle$ is incident on the 4LS, which is initialized to the state $|1\rangle$. From calculations detailed in Appendix B.2, the output state is:

$$|\psi_1^{\text{out}}\rangle = \alpha_A (r_{11} |\omega\rangle \otimes |1\rangle + r_{13} |\omega'\rangle \otimes |3\rangle) + \beta_A |\emptyset\rangle \otimes |1\rangle, \quad (4.7)$$

where the amplitudes r_{11} and r_{13} are:

$$r_{11} = e^{2i\omega a} \frac{\Gamma' - \Gamma (e^{2i\omega' a} - e^{-2i\omega a})}{-\Gamma' + \Gamma (e^{2i\omega' a} + e^{2i\omega a} - 2)}, \quad (4.8)$$

$$r_{13} = \frac{\Gamma (e^{2i\omega a} - 1) (e^{2i\omega' a} - 1)}{-\Gamma' + \Gamma (e^{2i\omega' a} + e^{2i\omega a} - 2)}. \quad (4.9)$$

If the boundary condition that

$$a = \frac{n\pi}{\omega + \omega'} \text{ for some } n \in \mathbb{N} \quad (4.10)$$

is satisfied, then in the strong-coupling limit ($\Gamma/\Gamma' \rightarrow \infty$), $r_{11} = 0$ and $r_{13} = -1$, so $|\psi_1^{\text{out}}\rangle = -\alpha_A |\omega'\rangle \otimes |3\rangle + \beta_A |\emptyset\rangle \otimes |1\rangle$. Thus, the atom transitions from $|1\rangle \rightarrow |3\rangle$, stores the input photon, and releases an auxiliary A' photon at frequency ω' into the delay line.

Step 2. At time $t = 2$, photon B with state $|\psi_2^{\text{in}}\rangle = \alpha_B |\omega\rangle + \beta_B |\emptyset\rangle$ is incident on the 4LS. After scattering, the output state is:

$$\begin{aligned} |\psi_2^{\text{out}}\rangle &= \alpha_B r_{11} \alpha_A r_{11} |\omega\rangle \otimes |\omega\rangle \otimes |1\rangle \\ &\quad + \alpha_B r_{13} \alpha_A r_{11} |\omega'\rangle \otimes |\omega\rangle \otimes |3\rangle \\ &\quad + \alpha_B R_3 \alpha_A r_{13} |\omega\rangle \otimes |\omega'\rangle \otimes |3\rangle \\ &\quad + \alpha_B r_{11} \beta_A |\omega\rangle \otimes |\emptyset\rangle \otimes |1\rangle \\ &\quad + \alpha_B r_{13} \beta_A |\omega'\rangle \otimes |\emptyset\rangle \otimes |3\rangle \\ &\quad + \beta_B |\emptyset\rangle \otimes |\psi_1^{\text{out}}\rangle, \end{aligned} \quad (4.11)$$

where the states are ordered as (photon B \otimes photon A \otimes atom), and where the reflection amplitude

of the resonant $|3\rangle \rightarrow |4\rangle \rightarrow |3\rangle$ transition is:

$$R_3 = \frac{\Gamma' e^{2i\omega a} + \Gamma(1 - e^{2i\omega a})}{-\Gamma' - \Gamma(1 - e^{2i\omega a})}. \quad (4.12)$$

As before, if the condition of Eq. 4.10 is satisfied, then $R_3 = -1 = e^{i\pi}$, so photon B gains a π phase. For simplicity, in the rest of this section, we focus on the case where Eq. 4.10 holds. Substituting the on-resonance coefficients of $r_{11} \rightarrow 0$, $r_{13} \rightarrow -1$, and $R_3 \rightarrow -1$ the output state at the end of step 2 is:

$$\begin{aligned} |\psi_2^{\text{out}}\rangle &= \alpha_B \alpha_A |\omega\rangle \otimes |\omega'\rangle \otimes |3\rangle - \alpha_B \beta_A |\omega'\rangle \otimes |\emptyset\rangle \otimes |3\rangle \\ &\quad - \beta_B \alpha_A |\emptyset\rangle \otimes |\omega'\rangle \otimes |3\rangle + \beta_B \beta_A |\emptyset\rangle \otimes |\emptyset\rangle \otimes |1\rangle. \end{aligned} \quad (4.13)$$

Step 3. At time $t = 3$, photon A' has traveled down the delay line, which has a length which is a multiple of $\frac{2\pi}{\omega'}$, and is returning to the atom. Its frequency ω' is resonant with the $|3\rangle \leftrightarrow |2\rangle$ transition, and the reflection coefficients r_{33} and r_{31} have expressions which are identical to Eqs. 4.8 and 4.9, respectively, except with ω, ω' exchanged, such that when $a = \frac{n\pi}{\omega + \omega'}$, we have that $r_{33} = 0$ and $r_{31} = -1$.

The state of the returning A' photon is $|A'\rangle = -\alpha_A |\omega'\rangle + \beta_A |\emptyset\rangle$, and it only interacts with the $|*\rangle \otimes |\omega'\rangle \otimes |3\rangle$ components of the system state, mapping $|*\rangle \otimes |\omega'\rangle \otimes |3\rangle \mapsto -1 \cdot |*\rangle \otimes |\omega\rangle \otimes |1\rangle$. Therefore, the system state at the end of step 3 is:

$$\begin{aligned} |\psi_3^{\text{out}}\rangle &= -\alpha_B \alpha_A |\omega\rangle \otimes |\omega\rangle \otimes |1\rangle - \alpha_B \beta_A |\omega'\rangle \otimes |\emptyset\rangle \otimes |3\rangle \\ &\quad + \beta_B \alpha_A |\emptyset\rangle \otimes |\omega\rangle \otimes |1\rangle + \beta_B \beta_A |\emptyset\rangle \otimes |\emptyset\rangle \otimes |1\rangle. \end{aligned} \quad (4.14)$$

Step 4. At time $t = 4$, photon B' is returning to the 4LS from the delay line. The reflection coefficients are the same as in step 3, and photon only interacts nontrivially with the $|\omega'\rangle \otimes |*\rangle \otimes |3\rangle$ components of $|\psi_3^{\text{out}}\rangle$, so the final output state is:

$$\begin{aligned} |\psi_4^{\text{out}}\rangle &= -\alpha_B \alpha_A |\omega\rangle \otimes |\omega\rangle \otimes |1\rangle + \alpha_B \beta_A |\omega\rangle \otimes |\emptyset\rangle \otimes |1\rangle \\ &\quad + \beta_B \alpha_A |\emptyset\rangle \otimes |\omega\rangle \otimes |1\rangle + \beta_B \beta_A |\emptyset\rangle \otimes |\emptyset\rangle \otimes |1\rangle. \end{aligned} \quad (4.15)$$

At the end of the gate operation, the emitter is restored to its original $|1\rangle$ state and is disentangled from photons A and B , and the two-photon state acquires a π phase shift only on the component corresponding to the presence of both A and B . Thus, the gate operation in the computational basis of spatial modes is:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad (4.16)$$

which is exactly the quantum controlled- σ_z gate.

We now return to describing the evolution of the state where we left off at Eq. 4.5. Using the

$|B\rangle \otimes |A\rangle \otimes |4LS\rangle$ ordering from Eqs. 4.11-4.15, we rewrite this equation to describe the states of the top and bottom photon-photon-4LS systems:

$$|\psi^{\text{top}}\rangle = +e^{i\eta}\frac{\alpha}{2}\sin\eta|\omega\rangle \otimes |\omega\rangle \otimes |1\rangle - \left(e^{i\eta}\alpha\cos\eta + e^{i\eta/2}\beta\cos\frac{\eta}{2}\right)|\omega\rangle \otimes |\emptyset\rangle \otimes |1\rangle \\ - e^{i\eta/2}\gamma\sin\frac{\eta}{2}|\emptyset\rangle \otimes |\omega\rangle \otimes |1\rangle + \delta|\emptyset\rangle \otimes |\emptyset\rangle \otimes |1\rangle, \quad (4.17)$$

$$|\psi^{\text{bot}}\rangle = -e^{i\eta}\frac{\alpha}{2}\sin\eta|\omega\rangle \otimes |\omega\rangle \otimes |1\rangle - e^{i\eta/2}\beta\cos\frac{\eta}{2}|\omega\rangle \otimes |\emptyset\rangle \otimes |1\rangle \\ - \left(e^{i\eta}\alpha\cos\eta + e^{i\eta/2}\gamma\cos\frac{\eta}{2}\right)|\emptyset\rangle \otimes |\omega\rangle \otimes |1\rangle + \delta|\emptyset\rangle \otimes |\emptyset\rangle \otimes |1\rangle. \quad (4.18)$$

The photons scatter off of the quantum emitters, producing ancillary photons which travel down the delay lines and back and release the original photons, but with a π phase shift applied to the $|\omega\rangle \otimes |\omega\rangle \otimes |1\rangle$ component of the state where both photons are present. Thus, the first term changes sign for each of Eqs. 4.17 and 4.18, and the output state when the photons finally return to the MZI in Figure 4.1c, at (4a, 4b) is:

$$|\psi\rangle = e^{i\eta}\frac{\alpha}{2}\sin\eta\left(-(\hat{b}_L^{\text{top}\dagger})^2 + (\hat{b}_L^{\text{bot}\dagger})^2\right)|\emptyset\rangle \\ - e^{i\eta}\alpha\cos\eta\hat{b}_L^{\text{top}\dagger}\hat{b}_L^{\text{bot}\dagger}|\emptyset\rangle \\ - e^{i\eta/2}\left(\beta\cos\frac{\eta}{2}\hat{a}_{0,L}^{[2]\dagger} + \gamma\sin\frac{\eta}{2}\hat{a}_{0,L}^{[1]\dagger}\right)\hat{b}_L^{\text{top}\dagger}|\emptyset\rangle \\ - e^{i\eta/2}\left(\beta\sin\frac{\eta}{2}\hat{a}_{0,L}^{[2]\dagger} - \gamma\cos\frac{\eta}{2}\hat{a}_{0,L}^{[1]\dagger}\right)\hat{b}_L^{\text{bot}\dagger}|\emptyset\rangle \\ + \delta\hat{a}_{0,R}^{[1]\dagger}\hat{a}_{0,R}^{[2]\dagger}|\emptyset\rangle, \quad (4.19)$$

where we assume that the photons described by the $\hat{a}_{0,R}^{[q]\dagger}$ operators in Eq. 4.5 have been reflected and now travel in the L direction.

Propagating this state through the MZI at (3) one last time using $|\psi_{\text{out}}\rangle = T^\dagger(\eta)|\psi\rangle$, we obtain the final output state at (2a, 2b):

$$|\psi^{\text{out}}\rangle = e^{2i\eta}\alpha\cos(2\eta)\hat{a}_{1,L}^{[1]\dagger}\hat{a}_{1,L}^{[2]\dagger}|\emptyset\rangle \\ - e^{2i\eta}\frac{\alpha}{2}\sin(2\eta)\left((\hat{a}_{1,L}^{[1]\dagger})^2 - (\hat{a}_{1,L}^{[2]\dagger})^2\right)|\emptyset\rangle \\ + e^{i\eta}\beta\hat{a}_{1,L}^{[1]\dagger}\hat{a}_{0,L}^{[2]\dagger}|\emptyset\rangle \\ + e^{i\eta}\gamma\hat{a}_{0,L}^{[1]\dagger}\hat{a}_{1,L}^{[2]\dagger}|\emptyset\rangle \\ + \delta\hat{a}_{0,L}^{[1]\dagger}\hat{a}_{0,L}^{[2]\dagger}|\emptyset\rangle. \quad (4.20)$$

The output photons propagate to the circulators at (1a, 1b) and are reinjected back into their

original waveguides. In order to preserve photon numbers between waveguide pairs, the second term in Eq. 4.20 must be zero, since $(\hat{a}_{1,L}^{[1]\dagger})^2$ and $(\hat{a}_{1,L}^{[2]\dagger})^2$ correspond to injection of two photons into the same waveguide. This fact constrains η to phase shifts which are multiples of $\frac{\pi}{2}$.

We note the gate action of the entire system at $\eta = 0$ is identity and the action at $\eta = \frac{\pi}{2}$ is the $c\sigma_z$ operation, up to a phase shift of $\frac{\pi}{2}$ which can be included in the ζ, ξ phase shifters at the subsequent column in the lattice:

$$U(\eta = 0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbb{1} \quad (4.21)$$

$$U(\eta = \frac{\pi}{2}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left(R_0^{\pi/2} \otimes R_{\pi/2}^0 \right) c\sigma_z. \quad (4.22)$$

To summarize the results of this section, the photons are directed by circulators through an MZI and toward the scattering sites. Depending on the value of η , the four-level systems either interact with one ($\eta = 0$) or two ($\eta = \frac{\pi}{2}$) photons, and they impart a π phase shift onto the two-photon component of the state they receive. The photons retrace their path and return to their original waveguides to be operated on by the next column of gates in the lattice.

4.2.3 Fidelity and fault tolerance

The calculations in the previous sections have shown that in an ideal case, our photonic architecture can perfectly implement arbitrary single-qubit operators and $c\sigma_z$. However, this makes some assumptions about the construction of the device. Namely, we assume that waveguides are lossless, that photons are injected with frequency ω and vanishing spectral width $\delta\omega \rightarrow 0$, and that the excited states of the scattering systems are lossless with $\Gamma' \rightarrow 0$, such that the Purcell enhancement factors are large, with $P = \Gamma/\Gamma' \rightarrow \infty$.

In reality, photons would have finite spectral width and the local emitters would have finite Purcell factors, meaning the QPGA would implement logical operators with fidelity below unity. As photons propagate through the imperfect gates implemented by the physical circuit, the errors will in general accumulate to render very deep circuits useless. However, this can be addressed by a variety of error-correcting methods. The errors which could occur in a physical implementation of this circuit can broadly be classified into three types: spectral unitary errors from the MZIs, depolarizing errors from the scattering sites, and photon loss from the quantum emitters and waveguides.

MZIs acting on photons with finite spectral width (and dispersive effects in the waveguides) can reshape the photon pulse and transmit a portion of the pulse to the top and bottom waveguides which differs from the target amount. The photon is not lost to or entangled with the environment, so this error can be represented by a unitary operation \tilde{U} with a characteristic error ϵ which acts as $\tilde{U} |\psi^{\text{in}}\rangle = \sqrt{1-\epsilon} |\psi^{\text{targ}}\rangle + \epsilon |\psi_{\perp}^{\text{targ}}\rangle$, where $|\psi_{\perp}^{\text{targ}}\rangle$ is some state orthogonal to the desired output state $|\psi^{\text{targ}}\rangle$. [124] This error can, in principle, be trained around using the gradient-based circuit

optimization approach discussed in Section 4.4. However, as shown in Appendix B.1, the fidelity of the MZIs can be quite high even for short pulses (a 1ns pulse has infidelity of 10^{-10}), so the dominant source of error would come from the scattering operations.

The infidelity in the two-photon gates $\mathcal{F}^{-1} = 1 - |\langle \psi_A, \psi_B, \psi_{4LS} | c\sigma_z^{AB} \otimes \mathbb{1}^{4LS} | \psi_A, \psi_B, 1 \rangle|^2$ introduced by finite excitation loss and spectral width results in a photon-photon-emitter state which is not fully entangled during operation nor fully disentangled at the end of the operation. If we trace out the degrees of freedom of the four-level system, we obtain a mixed two-photon output state which is the desired output state, but with a probability $p = \mathcal{F}^{-1}$ of applying a second σ_z operation which undoes the original gate action. This corresponds to the well-studied quantum depolarizing error model [121, 81, 168], which describes quantum gates as being faulty by randomly applying Pauli operators with some effective error probability per gate (EPG). [119, 168] Fault tolerance⁵ requires an EPG below a certain threshold p_{th} , usually estimated as $p_{th} \approx 10^{-4}$ [80, 194], but for some architectures and scenarios as high as $p_{th} \approx 10^{-2}$ [119]. In this system, the EPG approaches zero as the Purcell factor tends to infinity, with $P = 40$ yielding a 6% infidelity. [274]

Photon leakage from the waveguide or from spontaneous emission from the scattering sites represents the dominating error mechanism in this design and can be completely and efficiently corrected using concatenated coding [120, 59] or by using one of the Bose-Chaudhuri-Hocquenghem family of codes [84] to correct for erasure errors [122]. Such codes allow for loss thresholds per gate above 1.7% [230] and possibly as high as 5% [122], corresponding to Purcell factors of $P \approx 100$ to $P \approx 30$. [274]

Relatively small QPGAs which do not employ error correction may already be feasible to implement. If one assumes current realistic values for silicon waveguide loss of 0.3 dB/cm [58], quantum emitters with a Purcell factor of $P = 80$ [35], and a unit cell path length of 500 μm , then photon loss is about 4% per unit cell. Thus, the total loss could be kept below 50% for a circuit as large as 16 layers, which is sufficient to perform high-fidelity approximate quantum Fourier transforms on four qubits. (see Figure 4.7). This estimate ignores the optical circulators, the details of which are not critical to the design, and which currently have comparatively high losses of around 3-6 dB [272, 139], which would bring the loss per unit cell to about 50%. With some modification to the QPGA design, one can conceive a similar device which performs the same function but does not require optical circulators, e.g. by using a single unit cell to emulate a large gate array while storing many photon pulses in a large ring. However, with rapid experimental in waveguide-cavity systems and nonreciprocal on-chip devices, larger-scale QPGAs may be feasible to implement in the foreseeable future.

⁵It should be noted that while $c\sigma_z$ and Clifford gates have fault tolerant constructions, it has been shown that no single error correcting code has transversal implementations for all gates required for universal and classically non-simulable circuits [68], necessitating additional constructions if other gates are to be included. [39]

4.3 Exact quantum state and operator preparation

Having established how the design presented in Section 4.2 acts on physical photonic qubits, we now discuss how the idealized logical model of the device can be programmed to prepare quantum states and to implement quantum operators. We assume no error in the device here and describe algorithms to implement the desired actions with perfect fidelity, albeit sometimes using circuits of great depth. In reality, finite device errors may make the more compact approximate circuit decompositions discussed in Section 4.4 more relevant than the exact decompositions presented in this section.

4.3.1 Universality of the design

The MZIs in the lattice can implement any single-qubit gate by parameterizing it through the ζ, ξ, θ, ϕ phase shifts. The nonlinear interactions between waveguide pairs implements $c\sigma_z$, which can be used in conjunction with H to implement a controlled-NOT ($c\sigma_x$) gate⁶ as $c\sigma_x = (\mathbf{1} \otimes H)c\sigma_z(\mathbf{1} \otimes H)$. [168] Since the set of single-qubit operations and $c\sigma_x$ gate comprises a universal gate set [18], the device is universal, such that a sufficient number of layers can be used to implement an arbitrary multi-qubit gate.

Phase shifter parameters which implement various common single- and two-qubit quantum gates are detailed in Appendix B.3. Notably, two-qubit gates can have differing $c\sigma_z$ parities, meaning that some require an even or odd number of successive $c\sigma_z$ gates to implement. This would be problematic in an architecture with fixed $c\sigma_z$ connectivity, as aligning circuit elements within a fixed lattice would be impossible; this necessitates a mechanism such as the η -shifted MZI described by Eq. 4.3 which can toggle the gate action between qubits.

4.3.2 State preparation

Arbitrary quantum states can be prepared on a lattice with nearest-neighbor connectivity using a circuit based on Ref. [112] consisting of a sequence of multi-controlled single-qubit rotations. Although the general worst-case complexity of this algorithm is $\mathcal{O}(n^2 2^n)$, an important class of quantum states, including Dicke states [23] and general symmetric states [112], can be efficiently prepared using such a lattice with a depth which is polynomial in the number of qubits.

Suppose we have a state $|\psi\rangle = \sum_{q \in \{0,1\}^n} \alpha_q |q\rangle$ with $\alpha_q \in \mathbb{C}$ which we would like to prepare. Let ξ_x for $x \in \{0,1\}^k$ and $1 \leq k \leq n$ denote the projection of $|\psi\rangle$ onto the computational basis vector $|x\rangle$, tracing over all qubits subsequent to k :

$$\xi_x = \sum_{x' \in \{0,1\}^{n-k}} \langle x, x' | \psi \rangle. \quad (4.23)$$

⁶Due to the nearest-neighbor connectivity of the architecture, $c\sigma_x$ between non-adjacent qubits must be implemented with a sequence of SWAP gates, which can in turn be implemented using three $c\sigma_x$ gates. [168]

For each string x of length k , define a k -ly controlled single-qubit rotation operator $U_{x_1 \dots x_k}$ acting on qubit $k + 1$ which maps:

$$U_{x_1 \dots x_k} |x_1 \dots x_k\rangle |0\rangle = \frac{\xi_{x_1 \dots x_k 0}}{\xi_{x_1 \dots x_k}} |x_1 \dots x_k\rangle |0\rangle + \frac{\xi_{x_1 \dots x_k 1}}{\xi_{x_1 \dots x_k}} |x_1 \dots x_k\rangle |1\rangle. \quad (4.24)$$

Each k -controlled operation can be implemented on the nearest-neighbor architecture of the lattice with $\mathcal{O}(k^2)$ depth in the lattice using the implementation depicted in Figure 4.10 of Ref. [168].

The brute-force algorithm for preparing $|\psi\rangle$ is the application of 2^n of these operations, as shown in the circuit diagram of Figure 4.2.

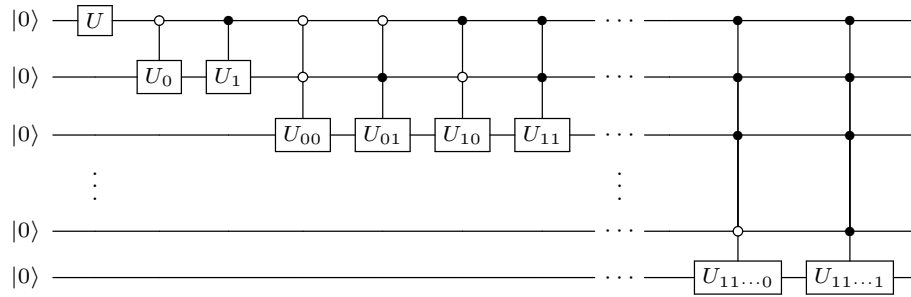


Figure 4.2: Brute-force state preparation algorithm to map $|0\rangle^{\otimes n} \mapsto |\psi\rangle$ using up to 2^n controlled rotations.

It can be shown by induction that after the first k rotations, the resulting state takes the form $\sum_{x_1 \dots x_k \in \{0,1\}^k} \xi_{x_1 \dots x_k} |x_1 \dots x_k\rangle$, so after all 2^n operations, the output state is:

$$\sum_{x_1 \dots x_n \in \{0,1\}^n} \xi_{x_1 \dots x_n} |x_1 \dots x_n\rangle = \sum_{q \in \{0,1\}^n} \alpha_q |q\rangle = |\psi\rangle. \quad (4.25)$$

Although this algorithm is not efficient for arbitrary quantum states, it is capable of efficiently preparing many interesting and important states. For example, an n -qubit GHZ state can be prepared on a nearest-neighbor lattice using n layers by setting $U = H$, applying singly-controlled $c\sigma_x$ between successive qubits, and discarding all other $U_{x_1 x_2 \dots x_k}$ operators.

4.3.3 Implementation of general quantum operators

Arbitrary $U(2^n)$ operations can be exactly implemented on the lattice using a nullification algorithm similar to the decomposition routines for classical optical meshes presented in Refs. [202, 50]. A more in-depth treatment of this problem can be found in Ref. [165].

In linear algebra, QR factorization decomposes any unitary matrix as $U = QR$, where R is diagonal and unitary and Q is a product of two-level Givens rotations [78, 165], which are operations

acting trivially on all but two basis vectors $|m\rangle, |n\rangle$:

$$\begin{aligned} G_{m,n}(\theta, \phi) &= e^{i\phi} \cos \theta |m\rangle\langle m| - \sin \theta |m\rangle\langle n| \\ &+ e^{i\phi} \sin \theta |n\rangle\langle m| + \cos \theta |n\rangle\langle n|. \end{aligned} \quad (4.26)$$

For any unitary matrix U , there exist values of θ, ϕ which “nullifies” a target element in row m or n of U . [50] Let $G_{m,n}^j$ denote the Givens rotation to nullify the element of U in row m , column j against the element in row n , column j . It can be shown [165] that after applying $\mathcal{O}(4^n)$ Givens rotations, we obtain an identity matrix:

$$\left[\prod_{j=1}^{2^n-1} \prod_{m=j+1}^{2^n} G_{m,m-1}^{2^n-j} \right] U = \mathbf{1}. \quad (4.27)$$

The operations $G_{m,m-1}^{2^n-j}$ do not correspond to any standard quantum gates, but if the basis vectors are permuted to be ordered in the reflected binary code [255], then the Givens rotations between adjacent vectors $|m\rangle, |m-1\rangle$ can be written as a product of $(n-1)$ -ly controlled single-qubit rotations [248], each of which can be performed with a lattice depth of $\mathcal{O}(n^2)$. Thus, the target operator U can be implemented as:

$$U = \prod_{j=1}^{2^n-1} \prod_{m=j}^{2^n-j} G_{\gamma(2^n-m+1), \gamma(2^n-m)}^{\gamma(j)\dagger}, \quad (4.28)$$

where $\gamma(j)$ denotes the index j in reflected binary ordering. The permutation for each of the $\mathcal{O}(4^n)$ Givens rotations requires $\mathcal{O}(n^3)$ $c\sigma_x$ gates, so the worst-case complexity is $\mathcal{O}(n^3 4^n)$.

As with state preparation, although implementing the most general quantum operators is hard, many important quantum operators, such as the quantum Fourier transform, may be efficiently implemented using a lattice of polynomial depth.

4.4 Gradient-based circuit optimization

In the previous section we discussed preparation of arbitrary quantum states or operators by obtaining appropriate phase shifter values to implement an exact decomposition of the desired operation using only single-qubit and nearest-neighbor $c\sigma_z$ gates. In this section, we demonstrate a method, building on our previous work for classical MZI networks [177, 259] and on work for continuous-variable quantum neural networks [10], of automatically discovering high-fidelity approximate decompositions of a target operator using a gradient-based optimization approach. As shown in Section 4.4.4, these “learned” implementations of quantum operators are often far more compact than an explicit decomposition, allowing for lattices with a fraction of the physical depth.

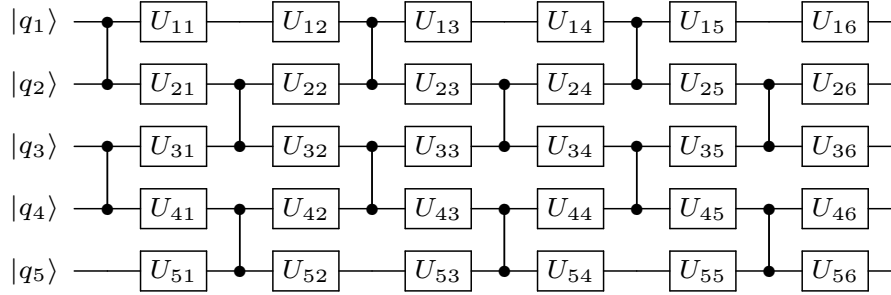


Figure 4.3: Fixed connectivity scheme employed in training. The $c\sigma_z$ operators in odd columns are implicitly constructed by embedding σ_x operations before and after physical $\overline{c\sigma_z}$ gates.

Let $U_{il} = U(\zeta_{il}, \xi_{il}, \theta_{il}, \phi_{il})$ denote the operation described by Eq. 4.1 acting on qubit i performed by a single MZI in layer l of the lattice. Each layer of the lattice refers to the column of MZIs implementing U_{il} and a subsequent column of $c\sigma_z^{i,j}$ gates between qubits i and j .

Because the strength of the $c\sigma_z$ interaction is not a continuous variational parameter (since the only valid settings are $\eta = 0$ (off) or $\eta = \frac{\pi}{2}$ (on), as discussed in Section 4.2.2), in our numerical experiments, we employ a checkerboard-style connectivity where half of the $c\sigma_z$ gates are disconnected, as shown in Figure 4.3. In a given layer, the $c\sigma_z$ gates are applied to each pair of adjacent qubits with an offset determined by the parity of the layer index. Additionally, we implicitly embed logical σ_x gates in the single-qubit operators preceding and following two-qubit gates in odd layers, such that $U_{i,n} \mapsto U_{i,n}\sigma_x$ and $U_{i,n+1} \mapsto \sigma_x U_{i,n+1}$ for odd n ; this transforms the $\overline{c\sigma_z}$ gates applied in odd layers into $c\sigma_z$ gates without adding to the depth of the lattice.

The operation performed on an N -qubit quantum state by a lattice of depth L with this connectivity scheme is given by:

$$\mathcal{U}_{\vec{\Theta}} = \prod_{l=1}^L \left[\bigotimes_{i \in \mathcal{C}(l)} c\sigma_z^{i,i+1} \cdot \bigotimes_{i=1}^N U(\zeta_{il}, \xi_{il}, \theta_{il}, \phi_{il}) \right], \quad (4.29)$$

where $\vec{\Theta}$ denotes all free parameters $\{\zeta_{il}, \xi_{il}, \theta_{il}, \phi_{il}\}$ in the lattice, where the set of $c\sigma_z$ connections is $\mathcal{C}(l) = \{1, 3, 5, \dots, 2\lceil \frac{N}{2} \rceil - 1\}$ [$\mathcal{C}(l) = \{2, 4, 6, \dots, 2\lfloor \frac{N}{2} \rfloor\}$] for odd [even] l , and where left-multiplication and padding with identity are implicit.

Let $\mathcal{F}(\tilde{\psi}, \psi) = |\langle \tilde{\psi} | \psi \rangle|^2$ denote the fidelity between states $|\tilde{\psi}\rangle$ and $|\psi\rangle$. To implement a target operator \hat{U} , the optimization routine finds a set of parameters $\vec{\Theta}$ which maximizes the average fidelity $\mathcal{F} = |\langle \psi_{\text{in}} | \mathcal{U}_{\vec{\Theta}}^\dagger \hat{U} | \psi_{\text{in}} \rangle|^2$ over a “training set” of input states $\{\psi_{\text{in}}\}$. The algorithm computes the gradient $\nabla_{\vec{\Theta}} \mathcal{F}$ of the fidelity over the training states with respect to the phase shift parameters and iteratively updates $\vec{\Theta}$ by a step size η as $\vec{\Theta} \mapsto \vec{\Theta} + \eta \nabla_{\vec{\Theta}} \mathcal{F}$ over the course of the training. In the case of operator implementation, $\{\psi_{\text{in}}\}$ are an ensemble of uniformly randomly sampled state vectors, while for state preparation, $\{\psi_{\text{in}}\} = \{|0\rangle^{\otimes N}\}$.

In the following subsections we present a series of numerical experiments in which a simulated logical model of a QPGA is trained to implement a variety of quantum states and operators. The numerical model was programmed using a custom backend built with `TensorFlow` [2], and the source code for all experiments in this paper is available at github.com/fancompute/qpga.

For operator preparation simulations, we generate the training set $\{\psi_{\text{in}}\}$ of random n -qubit state vectors by randomly choosing 2^n component magnitudes uniformly between $[0, 1)$, then renormalizing the state vector and assigning each component a random phase between $[0, 2\pi)$. The number of training samples is empirically chosen, but always greatly exceeds 2^n . The corresponding target output states are produced by running the input states through an explicitly constructed quantum circuit simulated using the `SQUANCH` Python framework [19]. For state preparation simulations, the training set is simply the zero state input $|0\rangle^{\otimes N}$ and the corresponding single output state is directly compared against the target state.

For all simulations, we used the checkerboard connectivity scheme described in the previous section. We initialized all ζ, ξ, θ, ϕ phase shifters uniformly from $[0, 2\pi)$, optimized the gate array using the Adam optimizer [117] with learning rate annealing, and performed the training on an NVIDIA Tesla K80 GPU.

4.4.1 GHZ state preparation

Greenberger–Horne–Zeilinger (GHZ) states [85] are maximally-entangled multi-qubit states of the form $\frac{1}{\sqrt{2}}(|00\dots 0\rangle + |11\dots 1\rangle)$ and have important applications in quantum information and quantum cryptography. [97]

Figure 4.4 shows the optimization progress of a four-qubit GHZ state. We simulated a small four-qubit QPGA with a fixed depth of 20 layers. (As noted in Section 4.3.2, a 4-qubit GHZ state can be exactly implemented using only 4 circuit layers, but we use the same simulated 20-layer device with the checkerboard $c\sigma_z$ connectivity for all simulations in this section.) For visualization purposes, we used a low learning rate and only displayed the first 100 iterations of training. Using a deeper lattice with longer training, arbitrarily high fidelities can be reached.

The stochastic nature of the initialization and optimization routines means that the training converges non-deterministically. Shallower circuits have fewer variational parameters to optimize and fewer layers to allow entanglement to propagate between nearest-neighbor qubits, which can result in a final fidelity which is far from unity. Deeper circuits have more parameters to optimize but require greater computational resources to simulate (and experimentally would have more pronounced physical errors if this were being considered). The number of layers in the circuit was empirically chosen to be a small depth which would consistently reach $\mathcal{F} \approx 1$.

Due to the uniform initialization of the phase shifters in the lattice, the model initially outputs a random, non-maximally entangled quantum state lacking any apparent structure. As the optimization routine proceeds, the lattice produces states which have increasingly large $|0000\rangle$ and $|1111\rangle$

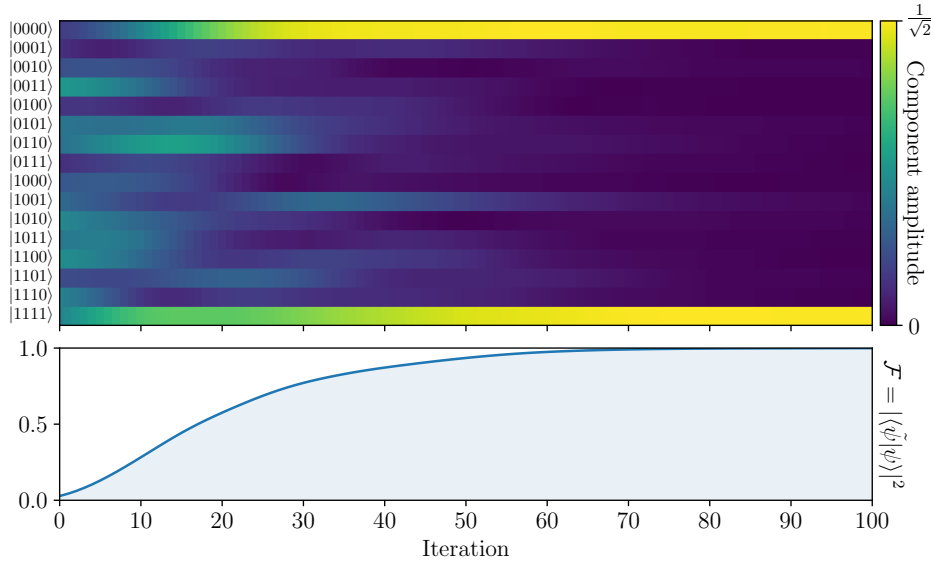


Figure 4.4: Optimization of a quantum circuit to prepare a four-qubit GHZ state. (Top) Evolution of the output state $|\tilde{\psi}\rangle$ over the course of training. The vertical axis represents the magnitude of the projection $\langle\tilde{\psi}|b_j\rangle$ of the output state onto each computational basis state $|b_j\rangle$. (Bottom) Fidelity between the output state and target state over the course of training, reaching a maximum value of $\mathcal{F} \approx 99.94\%$. The shared horizontal axis indicates iterations during training.

components, with the relative phase between these components approaching 0, while the other components of the output state have vanishing amplitudes. After 100 iterations, the model generates a state matching the target state with 99.94% fidelity.

4.4.2 Random state preparation

As discussed in Section 4.3.2, states with certain structures and symmetries are easier to prepare than general quantum states. To demonstrate the generality of the gradient-based circuit optimization routine, we use it to prepare a sample of random quantum states.⁷ The states are generated by choosing 2^n component magnitudes and phases uniformly, as described at the beginning of Section 4.4. We choose $n = 4$ qubits and fix a depth of 20 layers; the fidelities between the output states and target states over the course of training is shown in Figure 4.5. The results show that a QPGA of this depth is sufficient to create an arbitrary 4-qubit state with high fidelity.

⁷While the gradient-based circuit decomposition method will not bypass the exponential complexity of approximating general quantum states and operators (see Ref. [168], section 4.5.4), it is still informative to show that the method can implement states without specific structure.

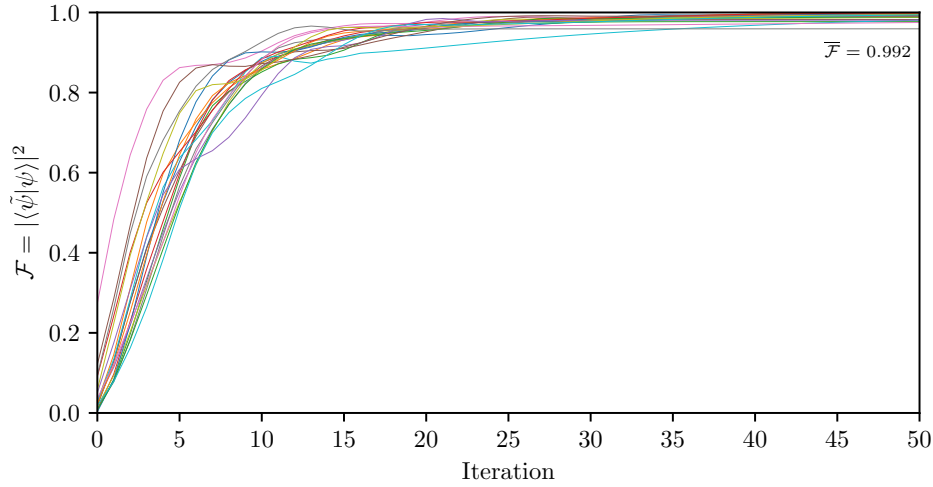


Figure 4.5: Training a 20-layer QPGA to prepare an ensemble of randomly sampled four-qubit states. Fidelities between the output and target states are shown over the course of each optimization. The average fidelity at the end of training is $\bar{\mathcal{F}} = 99.2\%$.

4.4.3 Quantum Fourier transform

The quantum Fourier transform is an important operator which plays a key role in many quantum computing algorithms, especially the eigenvalue estimation routine. [168] The quantum Fourier transform operating on n qubits takes the form:

$$U_{\text{QFT}} = \frac{1}{n} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} e^{2\pi i j k / n} |j\rangle\langle k|. \quad (4.30)$$

For this simulation, we compare the trainable circuit against the exact circuit implementation of the QFT, which has a complexity of $\mathcal{O}(n^2)$ (although the QFT can be approximated to within an inverse polynomial in n using only $\mathcal{O}(n \log n)$ gates [90]).

Figure 4.6 shows the optimization of a QPGA to implement a quantum Fourier transform on four input qubits. The explicit decomposition of the QFT circuit requires 57 layers⁸, but a trained QPGA with only 20 layers achieves a near-unity fidelity of $\mathcal{F} = 99.94\%$.

4.4.4 Circuit compactness analysis

In the previous sections, we have shown that gradient-based circuit optimization can produce high-fidelity operators which are significantly more compact than their explicitly-decomposed counterparts and are implementable on QPGAs with significantly fewer layers. To better characterize this, we performed a search over qubit number and circuit depth to find trained circuits which match the

⁸We train against the explicit circuit provided in Ref. [168], Fig. 5.1, but additionally add $\lfloor \frac{n}{2} \rfloor$ SWAP gates, since the output qubits in the Fourier basis are otherwise in reverse order.

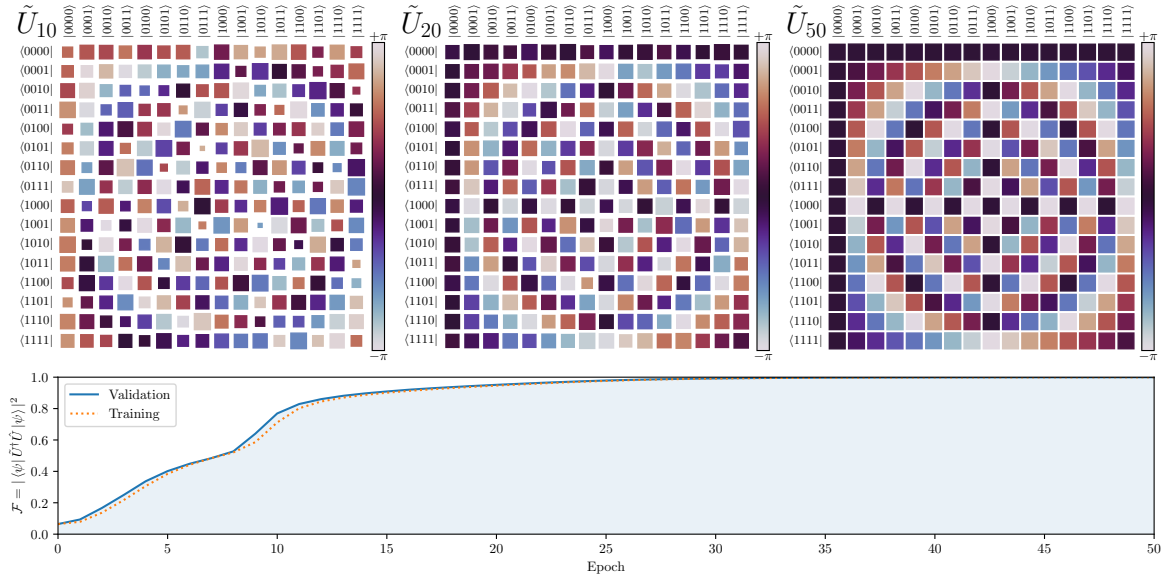


Figure 4.6: Optimization of a 20-layer QPGA to prepare a quantum Fourier transform on the four input qubits. (Top) The operators \tilde{U}_i implemented by the QPGA after i training epochs. Each square array represents the magnitude (relative to the maximum element) and phase of the projection of the operator onto the lexicographically-ordered computational basis states, encoded in the respective size and hue of the squares. The final \tilde{U}_{50} is visually indistinguishable from \hat{U} . (Bottom) Fidelity between the implemented and target operator over the course of training. The final fidelity is $\mathcal{F} = 99.94\%$. An animated version of this figure showing the training of the implemented operator can be found in the supplementary materials.

target operator to within some specified fidelity threshold. We used the quantum Fourier transform as the target operator for this benchmark due to its prevalence and complexity. The results are plotted in Figure 4.7.

To perform the compactness analysis, we iteratively trained QPGAs of increasing depth to implement an n -qubit QFT to a desired fidelity threshold, chosen to be $\mathcal{F} > 99.9\%$. Multiple optimization routines were run at each depth since training does not converge deterministically due to random initialization and the potential for getting stuck in a local maxima, which is more pronounced at larger qubit numbers.⁹ We note that the final gradient-based QFT implementations typically require only 1/4 to 1/3 as many layers as their explicitly decomposed counterparts.

⁹For circuits with many qubits, more sophisticated initialization routines which take the locally-connected structure of the architecture into account such as Haar initialization [177] may be necessary to ensure a reasonable chance of convergence.

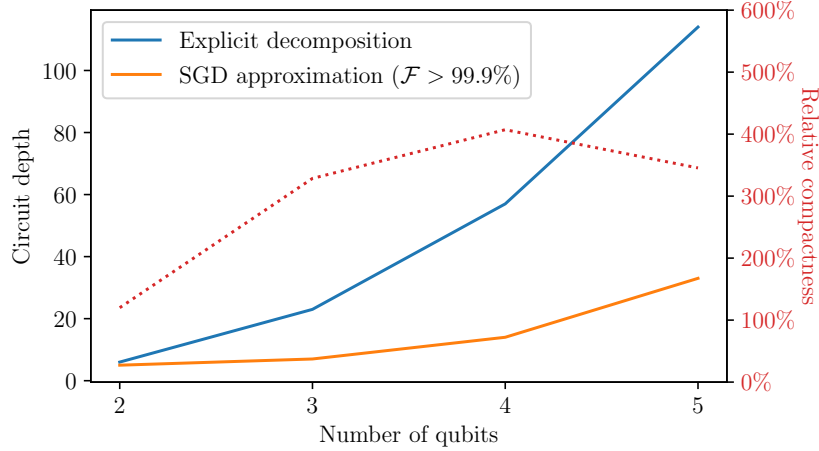


Figure 4.7: Required circuit depths to implement a quantum Fourier transform for a range of qubit numbers using explicit decomposition (top solid line, blue) and using gradient-based decomposition (bottom solid line, orange) which achieves a fidelity above 99.9%. Relative compactness of explicit vs. gradient-based decompositions is depicted by the red dotted line. The approximate decompositions are significantly more compact than the explicitly constructed circuits.

4.5 Conclusion

In this paper we have presented a photonic architecture for a quantum programmable gate array capable of implementing arbitrary quantum states, operators, and computations. The architecture, presented in Section 4.2, extends universal programmable optics to the quantum domain by employing two-photon interactions from quantum emitters embedded in the waveguides. This allows for deterministic multi-qubit gates which use a number of waveguides that is linear in the number of qubits. The design parameterizes arbitrary quantum circuits as a lattice of single-qubit gates implemented by phase-modulated Mach-Zehnder interferometers and two-qubit $c\sigma_z$ gates with variable connectivity implemented by a scattering process described in Section 4.2.2. By setting phase shifter parameters to implement appropriate single-qubit operations and to enable two-photon interactions where needed, the lattice can be dynamically programmed to implement any quantum circuit without hardware modifications.

In Section 4.3, we showed that the logical system implemented by the QPGA is computationally universal: any quantum operation can be mapped onto a corresponding set of phase shifter parameters given a sufficiently large lattice. We described an explicit algorithm for preparing arbitrary quantum states on the lattice which are efficient for some subclasses of quantum states, and we discussed how QR decomposition can map $U(2^n)$ unitaries onto a series of controlled rotations in the lattice.

In Section 4.4, we showed how gradient-based optimization techniques prevalent in machine learning can be used to automatically implement high-fidelity approximations to desired quantum

operations. We trained simulated QPGAs with fixed $c\sigma_z$ connectivity to prepare a variety of important quantum states and operators, and we showed that these approximate circuit implementations are often significantly more compact than their explicitly-decomposed counterparts.

While this work is purely theoretical, there has been tremendous recent experimental progress in both of the key technologies required to realize this device: programmable photonic processors [225, 92, 93, 91, 48, 208, 276, 186, 185] and strongly coupled quantum emitters [271, 199, 132, 115, 231, 15, 7, 35]. The ongoing advancements in these technologies may allow for feasible near-future implementation of the device described in this paper.

Chapter 5

Photonic quantum computing

In this chapter we present an architecture for a photonic quantum computer which can perform any computation using only a single coherently controlled atomic qubit [20]. Photonics offers unique advantages as a substrate for quantum information processing, but imposes fundamental scalability challenges. Nondeterministic schemes impose massive resource overheads, while deterministic schemes require prohibitively many identical quantum emitters to realize sizeable quantum circuits. Here we propose a scalable architecture for a photonic quantum computer which needs minimal quantum resources to implement any quantum circuit: a single coherently controlled atom. Optical switches endow a photonic quantum state with a synthetic time dimension by modulating photon-atom couplings. Quantum operations applied to the atomic qubit can be teleported onto the photonic qubits via projective measurement, and arbitrary quantum circuits can be compiled into a sequence of these teleported operators. This design negates the need for many identical quantum emitters to be integrated into a photonic circuit and allows effective all-to-all connectivity between photonic qubits. The proposed device has a machine size which is independent of quantum circuit depth, does not require single-photon detectors, operates deterministically, and is robust to experimental imperfections.

5.1 Introduction

Photonics offers many advantages for quantum information processing [253, 172, 275]: optical qubits have very long coherence times, are maintainable at room temperature, and are optimal for quantum communication. The main difficulty faced by all quantum computing (QC) architectures is scalability, but this is especially true for photonic systems. Optical qubits must propagate, so processing must be done mid-flight by passing the photons through sequential optical components. Since photonic quantum gates are physical objects (as opposed to, e.g. sequential laser pulses for atomic qubits), machine size scales with circuit depth, making complex quantum circuits prohibitively large

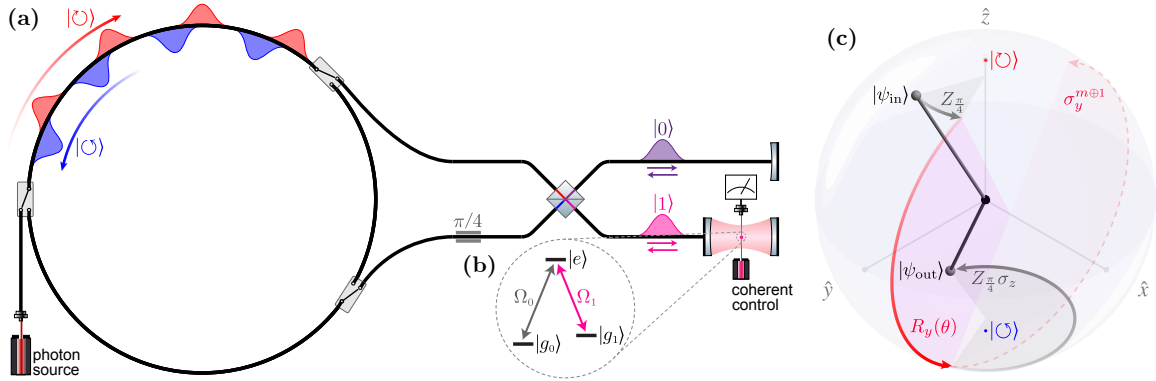


Figure 5.1: The photonic quantum computer architecture described in this work. **(a)** The physical design of the device. Photonic qubits counter-propagate through a fiber storage ring and optical switches can selectively direct photons through a scattering unit to interact with an atom in a cavity which is coherently controlled by a laser. **(b)** The energy structure of the atom: Ω_1 is resonant with the cavity mode and photon carrier frequency, while Ω_0 is far-detuned. **(c)** Bloch sphere depiction of the state of a photonic qubit in the $\{|0\rangle, |1\rangle\}$ basis and an operation applied by one pass through the scattering unit. The rotations about \hat{z} by fixed angles (grey) are applied by the phase shifter and beamsplitter, while the rotation about \hat{y} by a controllable angle θ (solid red) is applied to the atom using the cavity laser. Projectively measuring the atom teleports this rotation onto the photon, but may overshoot the target angle θ by π (dotted red) depending on the measurement outcome m . This operation is a universal single-qubit primitive: by composing several of these operations and adapting subsequent rotation angles based on measurement outcomes, arbitrary single-qubit gates can be deterministically constructed. See Supplement 2 for a visualization of the gate mechanism.

to implement even using compact integrated photonics.

Further limiting the scalability of photonic quantum computers is the difficulty of integrating many high-fidelity multi-photon gates into an optical circuit. This is an issue both for nondeterministic gate schemes [123, 125], which impose massive resource overheads for fault tolerant operation due to low gate success probabilities [138], and for deterministic scattering-based approaches [243, 247, 59, 205, 274]. Although scattering-based two-photon gates can be individually implemented with high fidelity [89, 242, 250, 25, 72], unrealistically large numbers of identical quantum emitters are needed to realize sizeable quantum circuits [21], a problem which is exacerbated in solid-state quantum emitters by poor indistinguishability due to homogeneous and inhomogeneous broadening [144, 5]. An architecture for a quantum computer which uses only a single quantum emitter to implement all gates in a quantum circuit would thus substantially improve the scalability and experimental feasibility of scattering-based photonic quantum computation.

Here we show that the emerging concept of synthetic dimensions [266, 32] naturally lends itself to such an architecture. Synthetic dimensions have recently generated great interest for exploring topological physics in photonics [174], but have not been extensively applied to quantum photonic systems. To form a synthetic dimension, one designs the couplings between states of a system,

either by repurposing the usual geometric dimensions, such as space [143] or time [203, 260, 147, 104, 134, 181, 154], or by augmenting these dimensions with internal degrees of freedom, such as frequency [268, 175, 26, 204, 261, 254], spin [43, 146, 235, 64], orbital angular momentum [142, 267], or Floquet-induced side bands [24, 149]. Since couplings between states within the synthetic dimension can be dynamically reconfigured and are not fixed by physical structure, one can scalably implement lattices with intricate connectivity. This allows multiple photonic qubits to be manipulated in synthetic space by a single quantum emitter without requiring spatially separated structures.

Our proposed design consists of a fiber loop coupled to a cavity containing a single coherently controlled atomic qubit. Optical switches endow the counter-circulating photonic states with a synthetic temporal dimension by allowing coupling between these states. By scattering photons against the atom and subsequently rotating and projectively measuring the atomic state, operations can be teleported onto the photonic qubits; these operations can be composed to deterministically construct any quantum circuit. Readout of the photonic quantum state can be performed without the need for single-photon detectors by sequentially swapping the state of the atom with each photonic qubit.

Our scheme has several unique characteristics. Most notably, the only controllable quantum resource is the single atomic qubit, which serves as a proxy to indirectly manipulate the photonic qubits. All quantum¹ operations and measurements on the photonic qubits are carried out by operations performed on this atom which are teleported onto the photons. This reduces the primary implementation challenge to preparing a single strongly coupled atom-cavity system, which has been experimentally demonstrated many times [53, 74, 214, 152, 55, 72, 205, 45]. The synthetic time dimension allows the single atom to serve as the nonlinearity for all quantum gates and provides effective all-to-all connectivity between the photonic qubits. The programmable nature of the teleported gates allows the atom to sequentially apply each required single- and two-photon gate without complex photon routing. This negates the requirement of conventional photonic QC schemes for many identical quantum emitters to be integrated into a photonic circuit. Finally, this design does not require single-photon detectors, which are a significant limitation to photonic QC. Instead, measurement of the atomic state can be performed with near-100% efficiency using the quantum jump technique, greatly improving the scalability of this design [162, 59, 74].

5.2 Design

The architecture for the scheme is shown in Figure 5.1(a). Qubits are encoded as trains of single photon pulses counter-propagating through an optical storage ring, where the two propagation directions $\{|\odot\rangle, |\ominus\rangle\}$ form the computational basis. A single-photon source injects photon pulses into

¹The only components which act on the photonic qubits are the 50:50 beamsplitter and $\pi/4$ phase shifter, which are fully static and serve only as a fixed change of basis of the photonic quantum state, and the optical switches, which only actuate between photon pulses in regions where the photon wavefunction is negligible.

the ring; each photon is spectrally narrow about a carrier frequency ω_c , has a pulse width τ , and occupies its own time bin with temporal spacing $\Delta t \gg \tau$. (The photon source need not be deterministic as long as the time bin of each photon can be resolved. Alternately, a dedicated single-photon source may not be needed, as the atom-cavity system discussed below could itself be used as the source by using the control laser to excite the atom [153, 60].)

The storage ring contains a pair of asymmetrically placed² optical switches, which can selectively direct photons from the ring through a static 50:50 beamsplitter and $\pi/4$ phase shifter and into a pair of waveguides. One of these waveguides is coupled to a cavity containing a single atom with a Λ -shaped three-level energy structure, shown in Figure 5.1(b). The atom has non-degenerate ground states $|g_0\rangle$ and $|g_1\rangle$ and an excited state $|e\rangle$, and the $|g_1\rangle \leftrightarrow |e\rangle$ transition at frequency Ω_1 is resonant with cavity mode frequency and photon carrier frequency ω_c . The atom is coherently controlled by a laser which applies rotations between $|g_0\rangle$ and $|g_1\rangle$, and its state can be measured in the $\{|g_0\rangle, |g_1\rangle\}$ basis. We refer to the subsystem consisting of everything except the storage ring and photon source (the right half of Figure 5.1(a)) as the “scattering unit”. The round-trip optical path length through the scattering unit is matched to the path length around the storage ring so that a photon returns to its original time bin after passing through the scattering unit.

After a photon scatters against the atom and is returned to the storage ring, a rotation is applied to the state of the atomic qubit and a projective measurement is performed, teleporting the rotation onto the photonic qubit, as shown in the next section. By composing three of these teleported rotations, arbitrary single-qubit gates can be deterministically constructed. A controlled phase-flip ($c\sigma_z$) gate between two photons can also be constructed with a similar process, enabling universal quantum computation. Readout of the final quantum state can be performed without the need for single-photon detectors by sequentially swapping the state of the atom with each photonic qubit.

5.2.1 Rotation teleportation mechanism

Here we outline the mechanism by which a rotation gate may be teleported onto a photonic qubit; we show in the next section that by composing these teleported rotations, arbitrary single-qubit gates may be constructed. A derivation of the mechanism described here is shown in greater detail in Appendices C.1 and C.2. Suppose we wish to apply a rotation to photon j , which occupies time bin t_j and is circulating in the storage ring in state $|\psi_{\text{in}}\rangle = \alpha |\circlearrowleft\rangle + \beta |\circlearrowright\rangle$, where $|\circlearrowleft\rangle$ and $|\circlearrowright\rangle$ denote the two counter-circulating states. While the optical switches are in the “closed” state, photons remain inside the storage ring; to operate on photon j , we “open” the switches at time $t_j - \Delta t/2$ and close them again at $t_j + \Delta t/2$ to direct photon j into the scattering unit. The photon passes

²To avoid time bin conflicts, the optical switches are asymmetrically placed in the storage ring. If the top switch is placed k time bins (a distance $kc\Delta t$) from the horizontal midpoint of the ring, then, as shown in Figure 5.1(a), the bottom switch is placed at a distance $(k + 1/2)c\Delta t$ from the midpoint, and the waveguide it connects to is $c\Delta t/2$ shorter than its counterpart. This ensures that when the bottom switch is set to the open state, directing a counterclockwise pulse into the bottom waveguide, the nearest clockwise pulse is $c\Delta t/2$ distance away.

through a $\pi/4$ phase shifter, which applies (up to a global phase) a $Z_{\pi/4} \equiv R_z(\pi/4) = \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$ rotation, and a 50:50 beamsplitter, which applies $B = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$. Before interacting with the atom, the photon is a superposition of modes in the top and bottom waveguides; we label these spatial modes as $|0\rangle$ and $|1\rangle$, respectively. We can thus relate the basis states of the ring and scattering unit via the unitary transformation $\{|0\rangle, |1\rangle\} = BZ_{\pi/4} \{|\odot\rangle, |\ominus\rangle\}$.

The $|0\rangle$ component of the photon state is reflected by a mirror in the top waveguide, imparting a π phase shift, while the $|1\rangle$ component undergoes a cavity-assisted interaction with the atom in the bottom waveguide, which is initialized in the state $|+\rangle \equiv \frac{1}{\sqrt{2}}(|g_0\rangle + |g_1\rangle)$. The $|g_1\rangle \leftrightarrow |e\rangle$ transition frequency Ω_1 is resonant with the cavity mode and photon frequency ω_c , while the $|g_0\rangle \leftrightarrow |e\rangle$ frequency Ω_0 is far-detuned. Thus, relative to the phase of the $|0\rangle$ mode, a π phase shift is applied to the $|1\rangle \otimes |g_1\rangle$ component of the $|\text{photon}\rangle \otimes |\text{atom}\rangle$ quantum state, implementing the unitary transformation corresponding to a controlled-Z gate between the atom and the photon, $c\sigma_z = e^{i\pi|1\rangle\langle 1| \otimes |g_1\rangle\langle g_1|}$. After scattering, the photon passes back through the beamsplitter and phase shifter and is returned to the storage ring. The joint state $|\Phi\rangle$ of the photon-atom system after a round trip through the scattering unit is:

$$|\Phi\rangle = (Z_{\pi/4} B \otimes \mathbf{1}) c\sigma_z (BZ_{\pi/4} \otimes \mathbf{1}) (|\psi_{\text{in}}\rangle \otimes |+\rangle). \quad (5.1)$$

After the photon has returned to the storage ring, a rotation $R_x(-\theta) = \exp(i\sigma_x\theta/2)$ is applied to the atomic qubit. Finally, a projective measurement of the atomic state in the $\{|g_0\rangle, |g_1\rangle\}$ basis is performed, obtaining a bit $m \in \{0, 1\}$. As shown in Appendix C.1, this atomic measurement projects the state of the photonic qubit to:

$$\begin{aligned} |\psi_{\text{out}}\rangle &= Z_{\pi/4} \sigma_z (-\sigma_y)^{m \oplus 1} R_y(\theta) Z_{\pi/4} |\psi_{\text{in}}\rangle \\ &= i^m Z_{\frac{5\pi}{4}} R_y(\theta + \pi(m \oplus 1)) Z_{\pi/4} |\psi_{\text{in}}\rangle, \end{aligned} \quad (5.2)$$

where $R_y(\theta) = \exp(-i\sigma_y\theta/2)$ and $m \oplus 1$ denotes addition modulo 2. Thus, the measurement teleports the $R_x(\theta)$ rotation of the atom to the $R_y(\theta)$ or $R_y(\theta + \pi)$ rotation of the photon, depending on m . The full sequence of operations is shown in Figure 5.2.

This teleportation scheme is an inversion of the paradigm of teleportation-based quantum computing [110, 82, 167]: in both cases, the original data qubit is entangled with an ancilla using a $c\sigma_z$ operation, but instead of rotating and measuring the data qubit to teleport the modified state onto the ancilla, in our scheme we rotate and measure the ancilla (the atom) to teleport a rotation onto the data qubit (the photon).

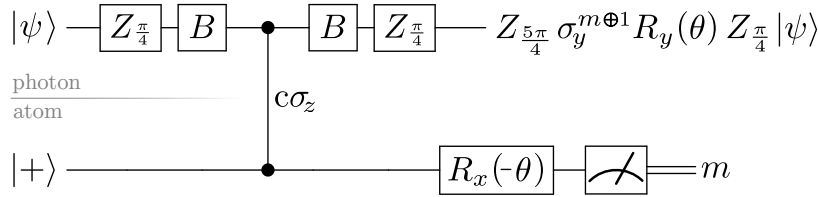


Figure 5.2: Quantum gate sequence corresponding to one pass of a photon through the scattering unit. The projective measurement teleports the rotation applied to the atomic qubit onto the photonic qubit.

5.2.2 Constructing arbitrary single-qubit gates

We now show that the teleported gate operation of Eq. 5.2 is sufficient to construct arbitrary single-qubit gates. The purpose of the $Z_{\frac{\pi}{4}}$ operations performed by the phase shifter is to rotate the basis in which the $R_y(\theta)$ gate is applied. Two passes of a photon through the phase shifter corresponds to a rotation on the Bloch sphere (see Figure 5.1(c)) about \hat{z} by 90° ; this change of basis causes a subsequent $R_y(\theta)$ to effectively rotate about \hat{x} . An additional two passes through the phase shifter rotates \hat{x} to $-\hat{y}$, allowing $R_y(\theta)$ to act about \hat{y} again. The goal here is to construct an operation that has the form $U = R_y(\theta_3)R_x(\theta_2)R_y(\theta_1)$, which is sufficient to implement any single-qubit gate up to an overall phase decomposed via Euler angles [110].

Consider a sequence of three teleported rotation gates (Eq. 5.2) about angles $\theta_1, \theta_2, \theta_3$ which yield measurement results m_1, m_2, m_3 . As we build up the target operator U with these successive rotations, the outcomes m_1, m_2, m_3 can result in extraneous Pauli gates between rotations which effectively offset the target angles $\theta_1, \theta_2, \theta_3$ by π , as in the second line of Eq. 5.2. Intuitively, this is equivalent to constructing an arbitrary rotation in 3D space using only fixed 90° rotations about \hat{z} , together with variable rotations about \hat{y} which may overshoot by π .

Borrowing a concept from measurement-based quantum computation [110, 167, 200], we apply rotations to the atomic qubit about *adaptive angles* of $\theta_2(m_1)$ and $\theta_3(m_2, m_1)$, each of which depends on the results of the preceding measurements. This allows us to propagate the Pauli errors from the middle of the gate to the front and consolidate them as a single error term. The sequence of three rotations performed in this adaptive basis thus implements the operation:

$$U = \varepsilon(m_3, m_2, m_1) \times Z_{\frac{\pi}{4}} R_y(\theta_3(m_2, m_1)) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}}, \quad (5.3)$$

where the rotations are implicitly programmed to implement U in the basis rotated by $Z_{\frac{\pi}{4}}$ and where the error term $\varepsilon(m_3, m_2, m_1)$ is σ_x , σ_y , or σ_z up to a global phase. This error term ε can then either (i) be implicitly removed by programming a subsequent gate U' to instead implement $U'\varepsilon^{-1}$ or (ii) be explicitly removed by scattering the photon against the atom initialized in the non-interacting $|g_0\rangle$ state or in the fully-interacting $|g_1\rangle$ state, applying σ_x or σ_z , respectively. The full derivation

for this gate construction process is shown in much greater detail in Appendix C.2.

5.2.3 Two-photon gates

In addition to implementing single-qubit gates, a two-photon entangling gate is needed for universal computation. A controlled phase-flip gate $c\sigma_z$ between two photonic qubits j and k can be constructed through a sequence of three scattering interactions in a manner similar to the protocol described by Duan and Kimble [59]. However, the beamsplitter and phase shifter, which are needed to implement the single-qubit gates in our scheme, only allow us to apply operations of the form shown in Eq. 5.1 to the photon-atom system with each pass of a photon through the scattering unit. This prevents us from performing the exact protocol described in Ref. [59] despite the similarities of the proposed physical systems.

We can resolve this complication by modifying the protocol to terminate with a measurement on the atom. We denote the operation applied to the photon-atom state by a pass of photon j through the scattering unit interacting with the atom a as:

$$\zeta^{ja} \equiv (Z_{\frac{\pi}{4}} B)^j c\sigma_z^{ja} (BZ_{\frac{\pi}{4}})^j. \quad (5.4)$$

To implement $c\sigma_z^{jk}$ between photons j and k , we pass photon j through the scattering unit, then k , then j again, separated by $R_y(\pm\frac{\pi}{2})$ rotations applied to the atom, as described in Appendix C.4. This results in the state

$$\zeta^{ja} R_y^a(\pi/2) \zeta^{ka} R_y^a(-\pi/2) \zeta^{ja} (|\psi_{jk}\rangle \otimes |+\rangle), \quad (5.5)$$

where $|\psi_{jk}\rangle$ is the arbitrary state of photons j and k and where the atom is initialized to $|+\rangle$. After this scattering sequence, we measure the state of the atom, which projects the two-photon state to:

$$(Z_{\frac{\pi}{4}} B \otimes Z_{\frac{\pi}{4}} B) \left(BZ_{(-1)^m \frac{\pi}{2}} B \otimes \mathbf{1} \right) \times c\sigma_z^{jk} \times (BZ_{\frac{\pi}{4}} \otimes BZ_{\frac{\pi}{4}}) |\psi_{jk}\rangle, \quad (5.6)$$

where the extraneous single qubit terms $BZ_{\frac{\pi}{4}}$, $Z_{\frac{\pi}{4}} B$, and $BZ_{(-1)^m \frac{\pi}{2}} B$ are artifacts of the photons passing through the beamsplitter and phase shifter. These extra gates are not problematic: when constructing a circuit from single-qubit gates and $c\sigma_z$, they may be removed by programming previous and subsequent single-qubit gates to include the inverse gates.

It is worth noting two alternative implementations of the photon-photon $c\sigma_z$ gate. First, using the SWAP operation implemented by Eq. 5.7, the states of one photonic qubit and the atom can be exchanged, and the second photon can directly interact with the state of the first, as discussed in Appendix C.4. Second, the protocol demonstrated by Ref. [89] can be implemented on this system, reducing the amortized number of scattering passes per $c\sigma_z$.

Our proposed device can thus implement arbitrary single-qubit gates and a two-photon $c\sigma_z$ gate.

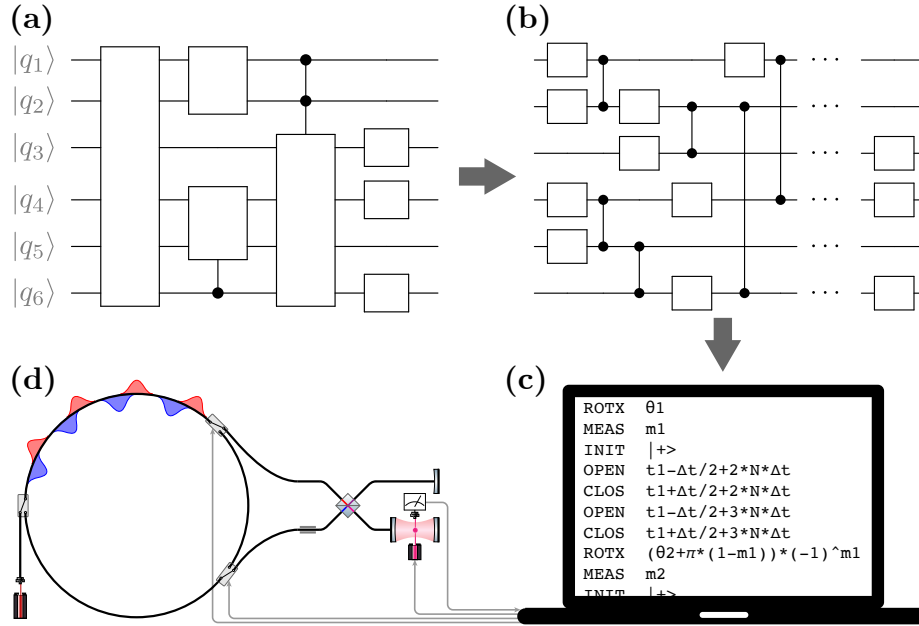


Figure 5.3: Conceptual illustration of compiling a quantum circuit into an instruction sequence to be performed on the device. (a) A generic target quantum circuit. (b) Decomposition into an equivalent circuit of single-qubit and $c\sigma_z$ gates. (c) The circuit is further decomposed into a sequence of scattering interactions. This sequence can be assembled on a classical computer into an instruction set with six distinct primitives which correspond to physical actions. (d) The controllable elements of the quantum device are the optical switches, cavity laser, and atomic state readout.

This comprises a universal gate set [18], so the device can perform any quantum computation.

5.2.4 Arbitrary circuit compilation

To implement an arbitrary n -qubit operator $U \in U(2^n)$, one could employ a three-step circuit compilation process outlined in Figure 5.3. First, decompose U into a sequence of single-qubit gates and $c\sigma_z$ operations. This is a well-studied problem [165] and can be done using the same operator preparation routine described in our previous work [21], but with an additional $\mathcal{O}(n)$ speedup, as this scheme has all-to-all instead of nearest-neighbor connectivity between qubits. Second, represent each $c\sigma_z$ as in Eq. 5.6 and decompose each single-qubit gate via Euler angles into rotations which may be teleported onto the photonic qubits. Finally, use a classical control system to modify the adaptive rotations which are applied to the atomic qubit based on the measurement outcomes during operation and to explicitly correct for ε Pauli errors when necessary. A more detailed discussion of the compilation process and an example instruction sequence to implement a three-qubit quantum Fourier transform can be found in Appendices C.5 and C.6.

5.2.5 Quantum state readout

After applying the desired quantum operation using the circuit compilation routine outlined above, the state of the photonic qubits must be measured to obtain a classical result. This can be done without the need for single photon detectors with their limited detection efficiencies by sequentially swapping the quantum states of each photonic qubit with that of the atom and repeatedly measuring the atomic state. To perform this SWAP operation, we scatter the desired photonic qubit j against the atom three times; between scattering operations, we apply the rotation $R_y(\pi/2)R_x(\pi)$ to the atomic qubit. Denoting this rotation as ρ^a and using ζ^{ja} as defined in Eq. 5.4, it is easily verified that

$$(BZ_{\frac{\pi}{4}})^j \zeta^{ja} \rho^a \zeta^{ja} \rho^a \zeta^{ja} (Z_{\frac{\pi}{4}} B)^j = e^{i\pi} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (5.7)$$

which is the SWAP operation up to a factor of -1. Here, $(BZ_{\frac{\pi}{4}})^j$ and $(Z_{\frac{\pi}{4}} B)^j$ are the operations applied to photon j on the outgoing and return trip from the scattering unit, respectively.³

5.3 Imperfection analysis

We now present a theoretical model to analyze the performance of our scheme in the presence of experimental imperfections. The main sources of error for our proposed scheme can be grouped into three classes: (i) deformation of the input pulse shape after scattering off the atom-cavity system, (ii) atomic spontaneous emission loss, and (iii) photon leakage due to attenuation and insertion loss while propagating through the storage ring and optical switches.

In our analysis, we assume the cavity mode frequency ω_c is exactly resonant with the atom $|g_1\rangle \leftrightarrow |e\rangle$ transition frequency Ω_1 , since the detuning can be calibrated to be zero in both free-space and nanophotonic systems [214]. Our design and simulations are agnostic to the carrier frequency⁴. We also assume that rotations of the atomic state using the cavity laser and measurement of the state can be done with fidelity $\mathcal{F} \approx 1$, as both processes have been demonstrated experimentally with infidelities significantly lower than the error sources listed above [36, 74, 29, 166, 95, 38]. For all simulations here, we choose a photon pulse width of $\tau = 100/\kappa$, a time range (bin size) $\Delta t = 500/\kappa$, and compute cooperativity with fixed $\gamma_s = \kappa/5$, where κ is the decay rate of the cavity into the waveguide and γ_s is the atomic spontaneous emission rate. This choice of parameters were motivated by a sample of experimental cavity setups enumerated in Figure 5.4 and result in a temporal bin size of order 100 nanoseconds for $\kappa/2\pi \sim 1\text{GHz}$. Greater detail is given in Appendix C.7.

³This construction of the SWAP gate starts with the photon in the middle of the scattering unit. If one starts with the photon in the storage ring, the state SWAP-ed to the atom is actually $BZ_{\frac{\pi}{4}} |\psi_j\rangle$. As before, this can be resolved by including $(BZ_{\frac{\pi}{4}})^{-1}$ in preceding single-qubit gates. Alternately, the atomic state can be directly manipulated after the SWAP gate is applied and before measurement to remove the extraneous $BZ_{\frac{\pi}{4}}$.

⁴In practice, the carrier frequency is chosen according to the transition frequency of the atomic qubit. If, for example, the atomic qubit is assumed to be a ⁸⁷Rb atom, the carrier frequency will be chosen to be $\omega_c \approx 3.8 \times 10^{14}$ Hz [89].

5.3.1 Deformation of photon pulses

We use the analytical technique described by Shen and Fan [223, 224] to exactly solve the single-photon transport problem of the coupled atom-cavity-waveguide system and obtain the output pulse $\phi_{\text{out}}(t)$ when the system is driven by an input pulse $\phi_{\text{in}}(t)$. This treatment captures the full quantum mechanical response of the system to a single-photon input Fock state for an arbitrary initialization of the atom without making the semiclassical assumption of a weak coherent input state.

Figure 5.4(a) shows the output pulse shapes for a single-photon Gaussian input pulse when the atom is initialized in states $|g_0\rangle$ or $|g_1\rangle$. For the $|g_0\rangle$ initialization, the response is identical to an empty cavity, since the $|g_0\rangle \leftrightarrow |e\rangle$ transition is far-detuned from the cavity resonant frequency. In this case, the output pulse is slightly delayed from the input pulse. For the $|g_1\rangle$ initialization, the photon is directly reflected from the front mirror of the cavity since the dressed cavity modes are well-separated in the strong coupling limit from the input photon frequency by the vacuum Rabi splitting, so the delay is minimal. We denote the difference in the delays of the $|g_0\rangle$ and $|g_1\rangle$ scatterings as Δt_{01} (see inset). We compute the pulse shape fidelity $\mathcal{F}_{\text{shape}}$ as the overlap integral of the output pulse with the input pulse after both pulses have been normalized to have unit area, and the pulse shape infidelity is $1 - \mathcal{F}_{\text{shape}}$. This quantity only describes mismatch of the shapes of the input and output pulses, not mismatch of the pulse areas; the infidelity due to photon loss is computed as a separate quantity.

In Figure 5.4(b), we plot the shape infidelity of various states as a function of the single-atom cavity cooperativity $C = 4g^2/\kappa\gamma_s$, where g is the atom-cavity coupling strength. The pulse shape infidelity from scattering off the $|g_1\rangle$ state decreases to negligible values as C increases, while the infidelity of $|g_0\rangle$ reaches an asymptote at 8×10^{-4} due to the delay of the output pulse by a time which is independent of C . The infidelity from scattering against $|+\rangle = (|g_0\rangle + |g_1\rangle)/\sqrt{2}$ thus reaches a value of 4×10^{-4} . Since the atom will usually be initialized to the $|+\rangle$ state during operation of the device, it is desirable to minimize the infidelity of this interaction. This can be done by equally distributing the delays between the $|g_0\rangle$ and $|g_1\rangle$ states by delaying the reference pulse by a time difference $\Delta t_{01}/2$, adding path length $c\Delta t_{01}/4$ to the top waveguide in Figure 5.1(a). This results in a “delay corrected” infidelity of 2×10^{-4} , which is independent of both cavity cooperativity (for $C \gg 1$) and atomic state initialization.

5.3.2 Atomic spontaneous emission loss

In Figure 5.4(b), we also plot the photon leakage probability for a scattering interaction. Atomic spontaneous emission noise from the excited $|e\rangle$ state into modes other than the cavity mode at a rate γ_s results in a partial loss of the photon, resulting in an output pulse with total photon number $\int dt |\phi_{\text{out}}(t)|^2 < 1$. We calculate the probability P_s of spontaneous emission loss as $P_s = 1 - \frac{\int dt |\phi_{\text{out}}(t)|^2}{\int dt |\phi_{\text{in}}(t)|^2}$. Spontaneous emission noise only applies to the $|1\rangle \otimes |g_1\rangle$ component of the photon \otimes atom state; since the atom will usually be initialized to the $|+\rangle$ state, if we average over the possible input

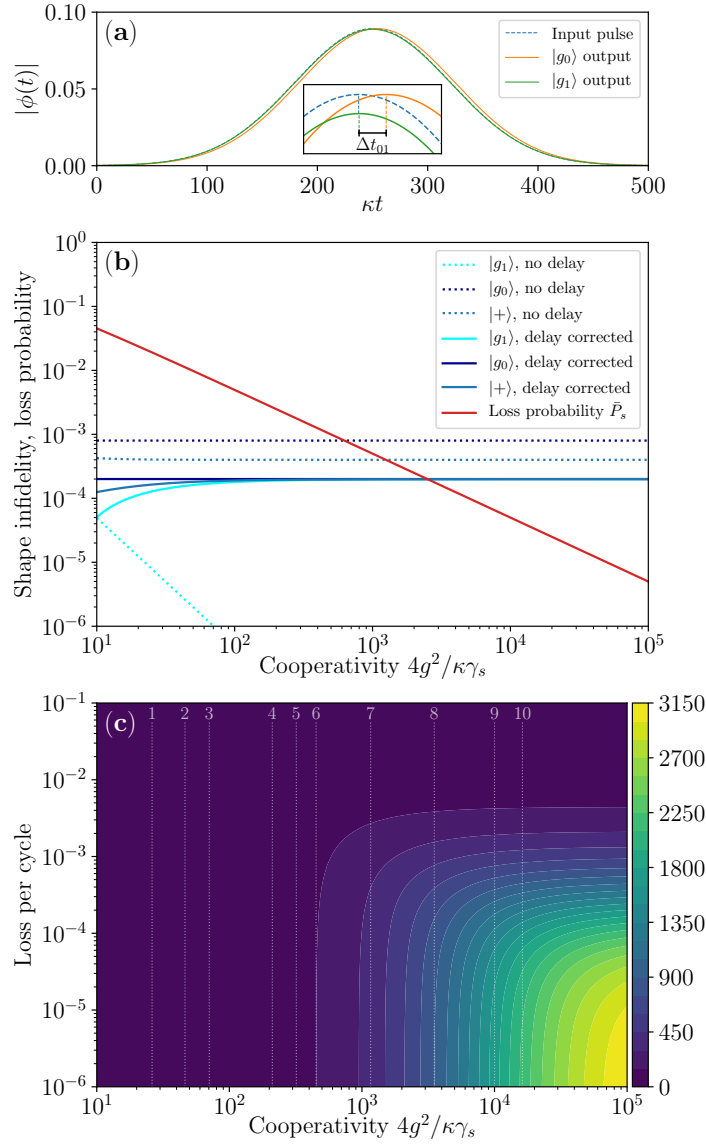


Figure 5.4: (a) Output pulse shapes for $|g_0\rangle$ and $|g_1\rangle$ initialization when a cavity with cooperativity $C = 180$ is driven by a Gaussian input pulse. The inset highlights the behavior near maximum: the $|g_0\rangle$ output pulse is delayed and the $|g_1\rangle$ output has reduced amplitude. (b) Shape infidelity and photon leakage probability as a function of cavity cooperativity. Solid blue lines show the pulse shape infidelity when the reference pulse is delayed by $\Delta t_{01}/2$. (c) Estimated single-qubit circuit depth achievable while maintaining $> 50\%$ fidelity as a function of cavity cooperativity and photon attenuation per cycle, assuming one scattering interaction every cycle and no error correction. Dotted lines show various experimentally demonstrated cooperativity values in similar cavity systems. Lines 1-10 correspond respectively to Refs. [89], [55], [214], [152], [72], [161], [53], [137], [45], and [236].

photon states, we obtain an average leakage probability of $\bar{P}_s = P_s/4$. This average photon loss probability is plotted as the red line in Figure 5.4(b) and ranges from about 5% to 0.0005% over the range of cooperativity values shown.

5.3.3 Optical attenuation and insertion loss

Finally, we account for loss due to attenuation in the optical storage ring and insertion loss from the switches as an average loss per cycle L . To estimate the maximum circuit depth D attainable with an overall fidelity $\mathcal{F}_{\text{target}}$, we compute a “bulk fidelity” per cycle accounting for shape infidelity, spontaneous emission loss, and attenuation while propagating through the storage ring and optical switches. For simplicity, we assume the circuit operates on only a single photonic qubit and that the photon is scattered against the atom with every pass through the storage ring. The achievable circuit depth operating with success probability $\mathcal{F}_{\text{target}}$ is thus the maximum D satisfying $[\mathcal{F}_{\text{shape}} \times (1 - \bar{P}_s) \times (1 - L)]^D \geq \mathcal{F}_{\text{target}}$, which is plotted as a function of cavity cooperativity and propagation loss in Figure 5.4(c) for $\mathcal{F}_{\text{target}} = 50\%$. Using optimistic but not unrealistic values for cooperativity [47, 45, 236] $C = 10^4$ and cycle loss $L = 10^{-4}$, we compute a bulk fidelity of $\mathcal{F} \approx 99.95\%$. This allows for an estimated depth of $D \approx 2000$ scattering operations while maintaining 50% success rate, and results in an error probability per gate (EPG) of $\sim 5 \times 10^{-4}$, below the estimated $\sim 10^{-3}$ EPG threshold for fault tolerance [168, 80, 194, 119]. Additionally, photon loss, which is likely the main error mechanism, can be efficiently corrected up to a per-gate loss of $\sim 10^{-2}$ using concatenated codes [122].

5.4 Discussion

In this chapter, we have shown how to use a single controllable quantum emitter to perform any quantum operation on a set of time-multiplexed photonic qubits. Related to but distinct from this work are proposals for generating time- and frequency-multiplexed 2D cluster states using a single or pair of quantum emitters [188, 69, 270, 34] or by using homodyne measurement of CV quantum systems [239], and experimental demonstrations using parametric nonlinearities [12, 131]. Although 2D cluster states are a universal resource for measurement-based quantum computation [167], the schemes that prepare these states can apply only a single type of quantum operation to the photonic qubits, and require single-photon detectors, with their associated limitations, for universality and state readout. In contrast, our scheme directly implements the quantum circuit model of QC, can deterministically construct any quantum gate, and can perform state readout without the need for photon detectors.

Our work also builds upon the well-known results of Ref. [59]. The physical setups are indeed similar: a cavity containing controllable three-level atom which can mediate interactions between scattered photons. However, Ref. [59] showed how the atom-cavity system can apply a fixed $c\sigma_z$

operation to a pair of photonic qubits, while our work shows how a loop-based design incorporating the cavity can perform *any* quantum operation on any number of photonic qubits. Notably, the single-qubit gate teleportation mechanism described in Sections 5.2.1-5.2.2 and the photodetector-free circuit compilation and state readout routines described in Sections 5.2.4-5.2.5 are, to our knowledge, unique to this work.

The main practical advantage of our scheme is the experimental simplicity of the design. Compared to other photonic QC approaches, our scheme shows a pathway to implement scalable, deterministic, gate-based quantum computation with photonics. Also, our scheme does not require single-photon detectors, which are a limitation for photonic approaches due to their low detection efficiencies. Compared to other platforms for quantum computing where qubits are individual physical structures, such as superconducting circuits and trapped ion systems, having only one controllable qubit provides a significant advantage to scalability: to add more qubits to our design just requires lengthening the fiber loop, while to add more qubits to a superconducting device requires adding complex individually-addressable components.

However, our scheme is not without drawbacks: the design requires high cavity cooperativity and low fiber attenuation, which are challenging but feasible to implement [161], and it relies on optical switches with very low insertion losses, although recent advances in lithium niobate modulators [251] may soon allow for this. Additionally, although having only a single controllable qubit does greatly simplify the experimental setup, it prevents two-qubit and most single-qubit gates in a quantum circuit from being performed in parallel.

More broadly, if we generalize the photon storage mechanism for our proposed device and consider synthetic dimensions other than time multiplexing, we could potentially further improve the scalability of our design. Instead of using counter-propagating optical modes, one could encode each qubit in the polarization basis and combine with fiber or free-space storage loops, as in Refs. [164, 209, 190]. With suitable design of the atom-cavity interacting system, frequency [268, 175] or angular momentum modes [142] could be also used as an alternative synthetic dimension, which could potentially mitigate the reliance of our design on low-loss optical switches, as demonstrated for heralded single-photon sources [108, 111, 109, 150]. These concepts would naturally lend themselves to studying quantum many-body physics of interacting Hamiltonians in synthetic space [174], which is difficult to realize in purely photonic platforms without the strong single-photon nonlinearity of the atom which we employ here [264, 173].

5.5 Conclusion

In this chapter we have presented a scheme for universal quantum computation using a single coherently controlled atom to indirectly manipulate a many-photon quantum state. We have shown that arbitrary single-qubit gates can be deterministically constructed from rotations applied to the

atomic qubit and teleported onto the photonic qubits via projective measurements. Using similar scattering processes, two-photon $c\sigma_z$ gates can be implemented, and readout of the photonic quantum state can be done using only atomic measurements with efficiencies far greater than that of state-of-the-art photon detectors. Our proposed scheme has high fidelity even in the presence of realistic experimental imperfections and offers significant advantages in required physical resources and experimental feasibility over many existing paradigms for photonic quantum computing.

Chapter 6

Quantum emulation

Synthetic dimensions have generated great interest for studying many types of topological, quantum, and many-body physics, and they offer a flexible platform for simulation of interesting physical systems, especially in high dimensions. In this chapter, we describe a programmable photonic device capable of emulating the dynamics of a broad class of Hamiltonians in lattices with arbitrary topologies and dimensions. We derive a correspondence between the physics of the device and the Hamiltonians of interest, and we simulate the physics of the device to observe a wide variety of physical phenomena, including chiral states in a Hall ladder, effective gauge potentials, and oscillations in high-dimensional lattices. Our proposed device opens new possibilities for studying topological and many-body physics in near-term experimental platforms.

6.1 Introduction

The emerging concept of synthetic dimensions in photonics has generated great interest for topological physics [63, 266, 24, 195, 174], optimization [104, 154, 147, 182], and quantum simulation and computation [32, 175, 44, 20, 197, 210, 262, 131]. Synthetic dimensions are formed by controlling couplings between degrees of freedom of a system, either by repurposing the usual geometric dimensions, such as space [143] or time [203, 260, 147, 104, 154, 135, 244, 96], or by augmenting these dimensions with internal degrees of freedom, such as frequency [268, 175, 26, 98, 254], spin [43, 146, 235, 63], or orbital angular momentum [142, 267]. Since couplings in synthetic dimensions can be dynamically reconfigured and are not fixed by a physical structure, one can scalably implement high-dimensional lattices with complex topologies, making this an ideal platform for quantum simulation.

6.2 Design

Here we describe a programmable photonic device capable of simulating the dynamics of interacting bosons in lattices with arbitrary dimensions, topologies, and connectivities using a synthetic time dimension. A large class of prototypical condensed matter Hamiltonians can be described by local two-body interactions on an arbitrary lattice. This class of Hamiltonians, which includes tight-binding models, the Hubbard and Bose-Hubbard models and their various extensions [66], and the Harper-Hofstadter-Hubbard model [206], can in general be described as (using $\hbar = 1$ throughout this chapter):

$$\hat{H} = - \sum_{\langle m,n \rangle} (\kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.}) + \mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m, \quad (6.1)$$

where κ_{mn} and α_{mn} respectively denote the tunneling coefficients and phases between connected sites $\langle m,n \rangle$, \hat{a}_m^\dagger creates a boson at site m , μ is the chemical potential, and U is the Hubbard interaction strength. The first term describes the tunneling of a particle between sites m and n , with a complex tunneling strength with amplitude κ_{mn} and phase α_{mn} ; the second term sets the energy per particle μ ; the third term is an on-site interaction potential with strength U which is active when a site contains more than one particle. This very general class of Hamiltonians exhibits rich phase diagrams and relates to quantum magnetism, high-temperature superconductors, and magnetic insulators, among many other applications. [13, 269, 77]

We propose a system that emulates the dynamics of the Hamiltonian in Eq. 6.1 using a synthetic temporal dimension. The design consists of a waveguide loop exhibiting a Kerr nonlinearity, which we refer to as the “storage ring”, in which a train of single-photon pulses propagates in a single direction, with each pulse occupying its own time bin. A second loop, the “register”, is connected to the storage ring using a Mach-Zehnder interferometer (MZI) with two tunable phase shifters, θ and ϕ . The hardware of the device is chosen to emulate each term of the Hamiltonian with dedicated components. The first term of Eq. 6.1 is implemented by the tunable MZI; the second term arises naturally from the total photon energy in each time bin; the two-photon potential in the third term results from using a Kerr-nonlinear fiber for the storage and register loops. We will briefly derive how each component implements the desired behavior and then describe how to program the device.

6.3 Device physics

A system evolving for a time interval t under the Hamiltonian given in Eq. 6.1 has a propagator $e^{-i\hat{H}t}$. We can split the exponential of the summation into a product of exponentials to within $\mathcal{O}(\kappa^2 + \kappa U \cos \alpha)$, where κ and α are typical values of κ_{mn}, α_{mn} (see Appendix D.1 for a more

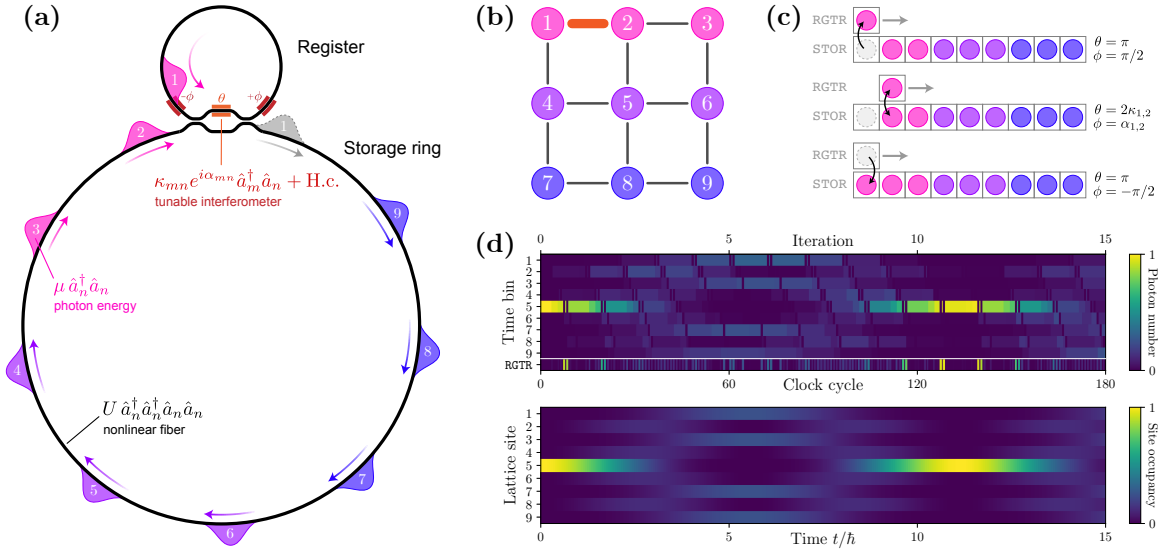


Figure 6.1: Architecture for the programmable photonic quantum emulator described in this chapter. **(a)** The physical design of the device. Classical laser pulses or single-photon pulses propagate clockwise through a fiber storage ring. A programmable Mach-Zehnder interferometer connects the storage ring to a register loop which has an optical path length Δx equal to the length of a single time bin. By setting the phase shift values in the MZI, the hopping coefficients and phases κ_{mn} , α_{mn} can be programmatically adjusted. Photons have energy $\mu \equiv \hbar\omega_0$, and by using a $\chi^{(3)}$ -nonlinear fiber, a nonlinear interaction potential U can be emulated. **(b)** An example 2D grid lattice to be emulated by the device. Node labels correspond to photon pulse indices, and the device as shown in panel (a) is in the process of constructing the orange edge connecting nodes 1 and 2 with $(\kappa_{1,2}, \alpha_{1,2})$. **(c)** Illustration of a single clock cycle of the emulator constructing the interaction $(\kappa_{1,2}, \alpha_{1,2})$ in three steps. First, phase shifters are set to transfer photon 1 into the register. Second, photon 1 is interacted with photon 2 using $\theta = 2\kappa_{1,2}$ and $\phi = \alpha_{1,2}$. Third, the pulse (which now may contain a mixture of photons 1 and 2) is returned to its original time bin. **(d)** The evolution of the state of the bin. The bottom panel depicts the exact evolution of the target Hamiltonian over time, while the top panel shows the state of the emulator at each clock cycle, including register swaps and intermediate states between full iterations. A large value of $\kappa = 0.2$ was used for visual clarity, but more accurate results may be obtained by using smaller κ and running the emulation for a commensurately longer wall-clock time.

detailed derivation):

$$\begin{aligned}
e^{-i\hat{H}t} &= \exp \left[-it \left(- \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) + \mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \right) \right] \\
&\approx \left[\prod_{\langle m,n \rangle} \exp (i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)) \right]^t \times e^{-it(\mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)}.
\end{aligned} \tag{6.2}$$

We therefore have a propagator that is a product of two parts: a continuous time evolution term $e^{-it(\mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)}$, which arises naturally from the photon energy per time bin (μ) and Kerr nonlinearity of the fiber (U), and the $\exp(i\kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.})$ terms, which are implemented in discrete time evolution by a sequence of passes through the tunable MZI. We now show how the device physics emulates the dynamics of the propagator.

For the chemical potential term, we can write the Hamiltonian for a photon with an arbitrary spectrum as $\hat{H}_{\text{EM}} = \int dk \sum_m \frac{1}{2} \hbar \omega_k (\hat{a}_{m,k}^\dagger \hat{a}_{m,k} + \hat{a}_{m,k} \hat{a}_{m,k}^\dagger)$. If we can assume that the photons are spectrally narrow about a carrier frequency ω_0 , we can approximate this as $\hat{H}_{\text{EM}} \approx \frac{1}{2} \hbar \omega_0 \sum_m (\hat{a}_m^\dagger \hat{a}_m + \hat{a}_m \hat{a}_m^\dagger) = \hbar \omega_0 \sum_m (\hat{a}_m^\dagger \hat{a}_m + \frac{1}{2}) \equiv \mu \sum_m \hat{a}_m^\dagger \hat{a}_m$, which directly gives us the desired chemical potential term.

The nonlinear potential naturally arises from the use of a nonlinear fiber. Consider a section of a Kerr-nonlinear fiber corresponding to one time bin, with length Δx and volume V . The material polarization at frequency ω induced by an electric field $E(\omega)$ is given by $\mathcal{P}_{\text{NL}}(\omega) = 3\epsilon_0 \chi^{(3)}(\omega) |E(\omega)|^2 E(\omega)$, where $\chi^{(3)}$ is the third-order susceptibility tensor, which can be treated as a scalar for isotropic media such as glass. The energy density \mathcal{U}_{NL} is related as $\mathcal{P}_{\text{NL}} = \partial \mathcal{U}_{\text{NL}} / \partial E^*$, and the Hamiltonian of this system, again assuming a narrow bandwidth about ω_0 , is $\hat{H}_{\text{NL}} = \int_V \mathcal{U}(\omega_0) d^3 \vec{r}$. After quantizing the field amplitudes as $E(\omega_0) \mapsto \sqrt{\frac{\hbar \omega_0}{2\epsilon_0 V}} (\hat{a}_{k_0}^\dagger e^{+i(\omega_0 t - k_0 z)} + \text{H.c.})$ and transforming into real space, we obtain $\hat{H}_{\text{NL}} = \left(\frac{9\hbar^2 \omega_0^2}{8\epsilon_0 n_0^4 V^2} \int_V \chi^{(3)} d^3 \mathbf{r} \right) \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + C \equiv U \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + C$, where the nonlinear potential coefficient is $U = \frac{9\hbar^2 \omega_0^2}{8\epsilon_0 n_0^4 V^2} \int_V \chi^{(3)} d^3 \mathbf{r}$ and where C is some constant corresponding to an overall energy shift. Applying this to each time bin gives us the desired $U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m$ nonlinear potential term.

Finally, the hopping terms arise from programmatically modulating the phase shifts in the MZI. To interfere two photons m and n with strength κ_{mn} and phase shift α_{mn} , the basic idea is to swap pulse m into the register ring, wait for pulse n to reach the MZI, interfere the pulses, then return pulse m to the storage ring when time bin m cycles back. Consider the MZI shown in Figure 6.1(a) with phase shifters $\pm\phi$ and θ . Define bosonic mode operators $\hat{a}_n^\dagger, \hat{a}_0^\dagger$ and $\hat{b}_n^\dagger, \hat{b}_0^\dagger$, which create a photon in time bin n or time bin 0 (the register bin), and at the input or output of the MZI,

respectively. We can relate the output and input mode operators as:

$$\begin{aligned}
\begin{bmatrix} \hat{b}_0^\dagger \\ \hat{b}_n^\dagger \end{bmatrix} &= \begin{pmatrix} \cos \frac{\theta}{2} & ie^{i\phi} \sin \frac{\theta}{2} \\ ie^{-i\phi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \begin{bmatrix} \hat{a}_0^\dagger \\ \hat{a}_n^\dagger \end{bmatrix} \\
&= \exp \left[i\frac{\theta}{2} \left(e^{i\phi} \hat{a}_0^\dagger \hat{a}_n + e^{-i\phi} \hat{a}_n^\dagger \hat{a}_0 \right) \right] \begin{bmatrix} \hat{a}_0^\dagger \\ \hat{a}_n^\dagger \end{bmatrix} \\
&\equiv \hat{M}_{0,n}(\theta, \phi) \begin{bmatrix} \hat{a}_0^\dagger \\ \hat{a}_n^\dagger \end{bmatrix}.
\end{aligned} \tag{6.3}$$

It is easily verified that the following identity holds: $\hat{M}_{0,m}(\pi, -\pi/2) \hat{M}_{0,n}(\theta, \phi) \hat{M}_{0,m}(\pi, +\pi/2) = \exp \left[i\theta/2 \left(e^{i\phi} \hat{a}_m^\dagger \hat{a}_n + e^{-i\phi} \hat{a}_n^\dagger \hat{a}_m \right) \right] \equiv \hat{T}_{m,n}(\theta, \phi)$. If we define $\kappa \equiv \theta/2$ and $\alpha \equiv +\phi$, we obtain the transfer matrix:

$$\hat{T}_{mn}(\kappa, \alpha) = \exp \left[i\kappa \left(e^{i\alpha} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha} \hat{a}_n^\dagger \hat{a}_m \right) \right]. \tag{6.4}$$

The middle θ phase shifter thus allows us to control the strength of the coupling κ , while the outer phase shifters $\pm\phi$ control the hopping phases.

By performing this sequence of passes through the MZI $\hat{T}_{\langle m,n \rangle} \equiv \prod_{\langle m,n \rangle} \hat{T}_{mn}(\kappa_{mn}, \alpha_{mn})$ for every photon pair $\langle m, n \rangle$ which corresponds to an adjacent pair of lattice sites m and n in the Hamiltonian, we complete one ‘‘iteration’’ of the emulator. If we allow the system to evolve for t iterations, we obtain a total transfer matrix which is exactly the first term in Eq. 6.2:

$$\hat{T}_{\langle m,n \rangle}^t = \left(\prod_{\langle m,n \rangle} \exp \left[i\kappa_{mn} \left(e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.} \right) \right] \right)^t \tag{6.5}$$

Therefore, all three components of the propagator are present, and the evolution of a state in the device for t iterations is described term-by-term by the propagator in Eq. 6.2. To adjust the relative values of continuous time evolution variables (μ, U) and discrete time evolution variables (κ, α) , one can adjust the photon energies μ , Kerr interaction strength U , time bin size Δx , or phase shifter values θ, ϕ .

The programmable MZI can construct lattices with arbitrary topology and connectivities. Consider a Hamiltonian of the form in Eq. 6.1 defined over a lattice described by an undirected graph $G = (V, E)$, as shown in Figure 6.1(b). We designate a time bin m for each lattice site $m \in V$, and for each edge $e = (m, n) \in E$ which couples sites m and n with coupling strength κ_{mn} and hopping phase α_{mn} , we perform a sequence of three passes through the MZI to interact time bins m and n , as in Figure 6.1(c). The first pass $\hat{M}_{0,m}(\pi, +\pi/2)$ swaps photon m into the empty register; the second pass $\hat{M}_{0,n}(2\kappa_{mn}, \alpha_{mn})$ performs the interaction between the register and time bin n ; the third pass $\hat{M}_{0,m}(\pi, -\pi/2)$ returns the photon to time bin m . This set of operations takes one ‘‘clock cycle’’ to complete, which is defined as the time for a pulse to fully propagate once around the storage

ring. Constructing all $e \in E$ completes one “iteration” of the emulator, and the state is allowed to evolve for t iterations. The edges can be constructed in any order as long as κ is small, which is always possible to do by decreasing κ, μ, U by some constant factor and running the emulator for a commensurately longer wall-clock time. It is also possible to modify the design of this device to include hardware optimizations for specific graphs. For example, the iteration time of a 2D square lattice with N sites can be reduced from N cycles to $2\sqrt{N}$ cycles per iteration by having two register rings of size N and \sqrt{N} to explicitly handle the vertical and horizontal strides of the graph.

6.4 Simulations and topological signatures

To more concretely show the capabilities of our proposal, we now provide several demonstrations of the device emulating systems of interest with experimentally measurable signatures. We show the device can create an effective gauge potential by emulating a synthetic Hall ladder, we demonstrate the quantum nature of the device by trapping a two-photon state using a synthetic field, and we demonstrate the reconfigurability of the device by emulating the evolution of a Bose-Hubbard Hamiltonian on a four-dimensional tesseract lattice. For these demonstrations, we wrote a Python simulator¹ built with QuTiP [107] which efficiently simulates the detailed physics of the device emulating a system of interest, such as register swaps and time bin interactions, and compares this against the exact Hamiltonian evolution. The simulator represents the state space of the system with a permutationally invariant bosonic lattice representation allowing for tractable simulation of Hamiltonians over moderately large lattices. This simulation method is described in greater detail Appendix D.3.

Figure 6.2 shows an emulated synthetic Hall ladder and obtains a similar bandstructure as the recent experimental results of Ref. [63]. This system exhibits chiral edge states in the presence of an effective magnetic field, which is induced by adding translation-invariant hopping phases $\pm\alpha/2$ to the outer edges of the ladder using the MZI. Figure 6.2(a) depicts the emulated ladder system; left and right nodes on each rung are mapped to pulses in even- and odd-indexed time bins. The band structures for the target and emulated Hamiltonians for this system are shown in Figure 6.2(b) for hopping phases $\alpha = 2\pi/3$ and $\alpha = 0$. Chiral edge states are clearly visible in the case of $\alpha = 2\pi/3$, indicating the presence of an effective gauge potential. The propagation of these chiral currents on the left leg of the ladder is shown Figure 6.2(c). In the presence of a gauge field, only one-way motion is allowed. The band structures for the synthetic case are computed by simulating one iteration of the propagator $\hat{G} = e^{-i\hat{H}(t=1)}$ in the device, taking the matrix logarithm $\hat{H} = \frac{\log \hat{G}}{-i}$, and then diagonalizing \hat{H} ; k values are computed using peak detection of the eigenstate Fourier transform (see Appendix D.4).

As shown in Fig. 6.2(b), the band structure from the emulated system (bottom row) closely

¹All simulation code for this chapter is available online at github.com/fancompute/synthetic-hamiltonians.

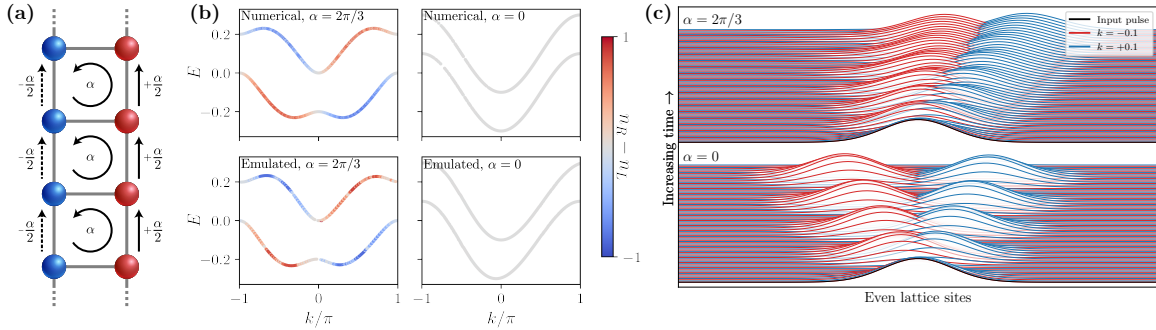


Figure 6.2: (a) Lattice diagram for a two-legged synthetic Hall ladder emulated with the device. By varying the inter-rung hopping phases α , an effective controllable magnetic field can be induced in the lattice. (b) Band structure of the system computed by diagonalizing the Hamiltonian for the exact case (top panels) and as emulated in the device (bottom) in the presence (left) and absence (right) of a synthetic magnetic field. Projection operators to the left and right nodes are color coded for each eigenstate. In all cases the Hamiltonian is represented in real space; for each eigenstate with eigenvalue E , we compute k with peak detection of its Fourier transform. This results in small numerical instabilities which are present in both the exact and emulated cases. Other parameters for this simulation: $\kappa = 0.1$, $\alpha = 2\pi/3$ or $\alpha = 0$, $\mu = U = 0$, number of lattice sites $D = 1000$, number of bosons $N = 1$. (c) Experimental signature for the propagation of chiral edge currents on the left leg of the ladder. A Gaussian input state is created with some initial $k = \pm 0.1$ by exciting multiple time bins with a phase difference between bins. When the gauge field is turned off ($\alpha = 0$), the pulses propagate in opposite directions, but when the field is turned on ($\alpha = 2\pi/3$), the motion in one direction is inhibited.

matches the desired band structure (top row, see also Ref. [99]), as well as the experimental results from very different platforms (Fig. 2 of Ref. [63]). This shows that the simulation of our device physics faithfully constructs the desired synthetic Hall Hamiltonian. Furthermore, because this demonstration uses only single boson, the single photon pulse could be substituted for a classical laser pulse which could be periodically re-amplified and reshaped, negating much of the experimental concerns related to attenuation and pulse deformation.

Next, to demonstrate the quantum capabilities of the device, we show how a two-photon state can be manipulated by introducing time-dependent hopping phases $\alpha(t)$ on a 1D lattice while using nonlinearity which is strong compared to the coupling constants $U \gg \kappa$. Figure 6.3 depicts the evolution of a two-photon state and a single-photon state under time-dependent hopping phases $\alpha(t)$. The energetic gap between $U \gg \kappa$ means that while $\alpha(t) = 0$, the two-photon state evolves the same as the single-photon state, but with a slower timescale for the evolution.² As $\alpha(t)$ is changed, $\partial\alpha/\partial t$ introduces an effective field, analogous to $\vec{E} = -\nabla V - \partial\vec{A}/\partial t$, which causes the two-photon state to look like it is “lensing” back to its original configuration. This field is maximized

²The two-photon state in panel (a) undergoes slower evolution because the Hamiltonian has no terms which can transport two photons simultaneously between lattice sites. Thus, evolution is only allowed via single-photon transport through an intermediate state which is lower in energy by U . This intermediate state never develops a sizable population because it is off-resonant from the initial and final states.

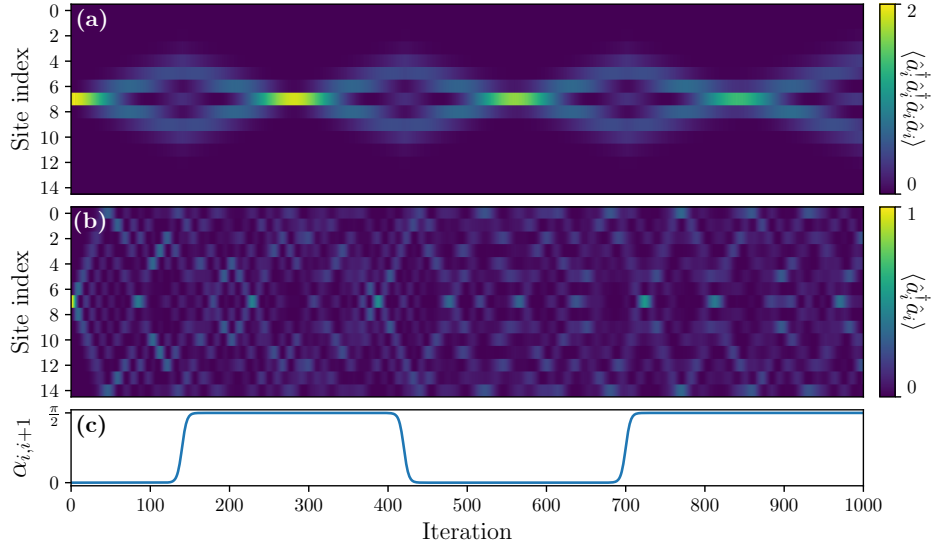


Figure 6.3: Emulated evolution of (a) a two-photon state and (b) a single-photon state in a 1D lattice as (c) time-dependent hopping phases are varied. The changing hopping phases introduce a changing gauge potential which causes the two-photon state to experience an effective electric field. The single-photon state is unaffected by this field.

at odd multiples of $\pi/2$, and by choosing suitable amplitude, duration, and periodicity of $\alpha(t)$, the two-photon state can effectively be trapped in the center of the lattice. The single-photon state is unaffected by the field, since we can perform a gauge transformation of the single-photon basis states as $\hat{a}_n^\dagger \mapsto \hat{b}_n^\dagger e^{in\alpha(t)}$ which eliminates the effect of $\alpha(t)$.

Finally, we demonstrate how the programmable nature of the device allows for emulation of complex, high-dimensional topologies. Figure 6.4 shows the evolution of a tight-binding Hamiltonian over a four-dimensional tesseract lattice emulated using the device. This demonstration uses a single degree of freedom (time) to emulate four independent physical synthetic dimensions. A projection of the non-planar graph defining the lattice is shown in Figure 6.4(a). The evolution of a two-photon state over this tesseract is shown in Figure 6.4(b): photons are initially placed in time bins 0 and 5, and oscillations across the tesseract are visible, with the photons oscillating between sites $0 \leftrightarrow 10$ and $5 \leftrightarrow 15$. (This is the expected behavior, representing the four-dimensional analogue of a boson oscillating between the corners of a 2×2 square lattice.) Two-photon correlation matrices are shown at different points in time in the upper panels. The state is plotted at the end of each iteration of the device; since photons have been swapped out of register time bin at the end of each iteration, it is shown to be empty at all times.

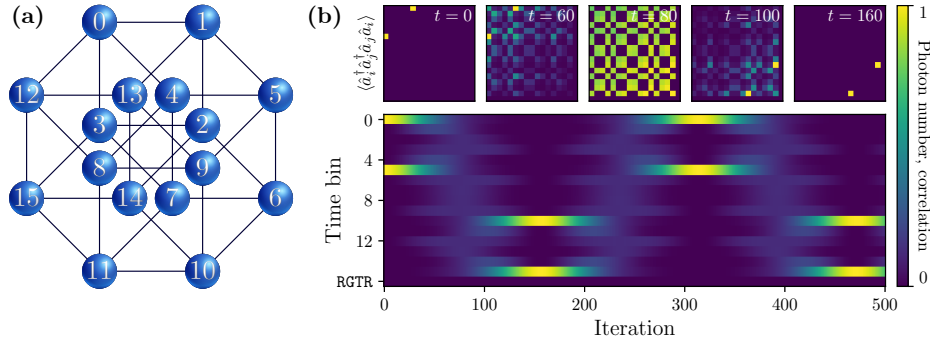


Figure 6.4: Emulation of a tight-binding Hamiltonian over a four-dimensional tesseract. (a) Projection of the tesseract graph which defines the lattice. (b) Evolution of a two-photon state exhibiting oscillations between time bins $0 \leftrightarrow 10$ and $5 \leftrightarrow 15$. Parameters: $\kappa = 0.01$, $\alpha = \mu = U = 0$.

6.5 Error analysis

Let us now briefly discuss the feasibility of this proposed device in the presence of experimental imperfections. The main limitations of the device are dispersion within the fiber loops³, optical attenuation in the fiber⁴, phase shifter actuation speed and insertion losses, and limitations on the nonlinear potential U .

For non-classically emulable cases with no nonlinearity ($U = 0$), single photon pulses must be used, which cannot be re-amplified, so attenuation and insertion losses will constrain the maximum lattice size which can be emulated with a given fidelity. If we take the tesseract graph with two photons from Figure 6.4 as an example, using a pulse width of 2 cm and a time bin size of 1 ns, the allowable cycle loss L to emulate a single iteration with 90% fidelity is $1 - L = 0.998$ or $L = -0.007\text{dB} / \text{cycle}$. Ignoring MZI insertion losses, this corresponds to -2.23 dB/km fiber attenuation, which is easily possible using commercially available fibers (with attenuation as low as -0.17 dB/km).

The most difficult cases to emulate are non-classical with large values of U . Highly nonlinear photonic crystal fibers filled with a high-density atomic gas can create nonlinearity up to $U/\hbar \sim 1 \text{ GHz}$ in the few-photon regime. [249, 264] To compare this to the numerical values of κ, U used in the simulations in this work, consider a time bin of size $\Delta t = n_{\text{fiber}} \Delta x / c$, where n_{fiber} is the refractive index. If there are N time bins, then the clock cycle time of the device is $N \Delta t$, so the frequency units for a numerical value of $\kappa = 1.0$ are $(2\pi N \Delta t)^{-1}$. For $\kappa = 0.2$ and nine time bins, as used in the lattice for Figure 6.1, this corresponds to $\kappa \approx 0.007 \text{ GHz}$. The value of U is independent of cycle time since it is distributed throughout the length of the fiber ring, and using current nonlinear fibers

³Dispersion is unlikely to have an important effect in emulation quality. Typical pulse parameters in a low-dispersion fiber allow for distinguishability over thousands of kilometers of distance (see Appendix D.5).

⁴For classically emulable cases (where the total boson number is $N = 1$ or where the initial state is well-approximated by a coherent state), single-photon pulses can be replaced by classical laser pulses with complex amplitudes, which can be re-amplified as needed, so attenuation and insertion loss is much less of a concern.

could be made several orders of magnitude larger than κ . Furthermore, recent improvements in the demonstrated nonlinearity-to-loss ratios for integrated photonic platforms approaching 1.5% [273, 141] (albeit in $\chi^{(2)}$ materials) show promise for achieving even higher values of U in the near future.

6.6 Conclusion

In this chapter we have presented a design for a programmable photonic device capable of emulating a broad class of classical and quantum Hamiltonians in lattices with arbitrary topologies. The device contains only a single actively controlled optical component – a phase-modulated MZI – and can be reprogrammed to emulate a wide variety of systems, such as chiral states in a Hall ladder, synthetic gauge potentials, and high-dimensional lattices. Our proposal opens new possibilities for studying fundamental topological and many-body physics, and it is experimentally feasible to build for a subset of these emulation scenarios.

Chapter 7

Conclusion and final remarks

The continual advancement of computing power has been the most important factor which has fueled technological progress in the 21st century. As electronic computing systems approach fundamental physical limits, photonics holds great promise for extending the development of computational capabilities. As we have shown in this thesis, light-based computing systems enable a wide range of new technologies, including low power and ultra-fast machine learning, scalable room-temperature quantum computation, and quantum emulation of exotic physics.

In Part I, we presented several new advancements in optical neural networks, which use lattices of programmable interferometers to physically process optically encoded information. We designed an electro-optic nonlinear activation for optical neural networks, and we subsequently demonstrated this design using a custom fabricated silicon nitride chip. The activation function operates by using a small portion of the input optical signal to modulate the optical signal intensity. This design eliminates the requirement of having additional optical sources between each layer of the ONN, and the activation function can be programmed or even trained to implement a variety of nonlinear responses. The proposed activation function and its experimental demonstration significantly improve the performance of optical neural networks on a variety of machine learning tasks, paving the way for much deeper ONNs.

Additionally, we proposed error-tolerant initialization routines and architectures for the programmable linear optical components which perform the bulk of the computation in optical neural networks. We simulate the gradient-based optimization of random unitary matrices on universal photonic devices composed of imperfect tunable interferometers. The locally-interacting nature of the optical devices biases the optimization search space towards banded unitary matrices, limiting convergence to random unitary matrices. To surpass this problem, we detailed a procedure for initializing the device by sampling from the distribution of random unitary matrices, which greatly improved convergence speed. We also proposed a number of architectural improvements, such as adding extra tunable beamsplitters or permuting waveguide layers, to which further improved the

training speed and scalability of these devices.

In Part II, we present three novel schemes for quantum computation and emulation. We extended the paradigm of programmable photonics explored in the first half of this thesis to the domain of quantum information with a design for a device called a quantum programmable gate array, which can be reprogrammed to prepare any quantum states without hardware modifications. This device parameterizes arbitrary quantum circuits using single-qubit rotations, performed by Mach-Zehnder interferometers, and controlled- σ_z gates, implemented using a deterministic two-photon scattering process mediated by strongly-coupled quantum emitters. Like an optical neural network, the device can be trained to implement desired operators, and we found that this gradient-based optimization can automatically implement compact, high-fidelity approximations to many important quantum circuits.

Next, we took the simplification of this design to its limits, designing a photonic quantum computer which can deterministically perform any computation using only a single coherently controlled atomic qubit. Borrowing methods from topological physics, our scheme endows a photonic quantum state with a synthetic time dimension by using optical switches to modulate photon-atom couplings. Quantum operations applied to the atomic qubit are teleported onto the photonic qubits via projective measurements, and arbitrary quantum circuits can be compiled into a sequence of these teleported operators. Our design has a machine size which is independent of quantum circuit depth, does not require single-photon detectors, operates deterministically, and is robust to experimental imperfections.

Finally, we explored the capabilities of programmable photonics for use in quantum emulation of exotic physics. We described a programmable photonic device capable of emulating the dynamics of a broad class of Hamiltonians in lattices with arbitrary topologies and dimensions. We simulated the physics of the device to observe a wide variety of physical phenomena, including chiral states in a Hall ladder, effective gauge potentials, and oscillations in high-dimensional lattices. This design opens new possibilities for studying topological and many-body physics in near-term experimental platforms.

In conclusion, the manipulation of light to process information is a natural and incredibly powerful computational paradigm, and optical computing may be a critical avenue to explore to continue to advance humanity's computational capabilities.

Part III

Supplementary materials

Appendix A

Matrix optimization on programmable photonics

This appendix contains supplementary calculations for the results presented in Chapter 3.

A.1 Derivation of beamsplitter errors

Unitary matrices generated by lossless MZIs are prone to errors in beamsplitter fabrication. We introduce the error ϵ to our expression derived in Equation 3.1, which is twice the displacement in beamsplitter split ratio from 50 : 50. Beamsplitter gates with error ϵ are defined as $B_\epsilon = \begin{bmatrix} \rho & i\tau \\ i\tau & \rho \end{bmatrix}$ where $\rho = \sqrt{\frac{1+\epsilon}{2}}$, $\tau = \sqrt{\frac{1-\epsilon}{2}}$ are transmissivity and reflectivity amplitudes that result in slight variations from a 50 : 50 beamsplitter. We use this error definition since it is a measurable quantity in the chip; in fact, there are strategies to minimize ϵ directly [158]. The unitary matrix that we implement in presence of beamsplitter errors becomes

$$\begin{aligned} U_\epsilon &:= R_\phi B_{\epsilon_2} R_\theta B_{\epsilon_1} \\ t_\epsilon &:= |U_{\epsilon,12}|^2 = |U_{\epsilon,21}|^2 \\ r_\epsilon &:= |U_{\epsilon,11}|^2 = |U_{\epsilon,22}|^2. \end{aligned} \tag{A.1}$$

If $\epsilon_1 = \epsilon_2 = \epsilon$, which is a reasonable practical assumption for nearby fabricated structures, then solving for t_ϵ in terms of t :

$$\begin{aligned} t_\epsilon &= 4|\rho|^2|\tau|^2t \\ &= 4t \left(\frac{1}{2} + \frac{\epsilon}{2} \right) \left(\frac{1}{2} - \frac{\epsilon}{2} \right) \\ &= t(1 - \epsilon^2). \end{aligned} \tag{A.2}$$

Similarly, we can solve for r_ϵ :

$$r_\epsilon = 1 - t_\epsilon = r + t \cdot \epsilon^2. \tag{A.3}$$

As we have discussed in this paper (and as we later show in Figure A.5), photonic errors ϵ (standard deviation of 0.1) can affect the optimized phase shifts for unitary matrices. The above constraints on r_ϵ and t_ϵ suggest that limited transmissivity is likely in the presence of fabrication errors, which can inhibit progressive setup of unitary meshes [158, 37]. However, we will later show through `tensorflow` simulation that *in situ* backpropagation updates can to some extent address this issue using a more sophisticated experimental protocol involving phase conjugation and interferometric measurements [102].

A.2 Derivation of the Haar measure

In this section, we outline a proof for the Haar measure of a unitary matrix in terms of the physical parameters of a photonic mesh to supplement our discussion of Haar phase and the proof in Ref. [212]. The Haar measure for $U(N)$ can be defined in two physical basis representations: the measurement basis represents measurements after each MZI and the transmissivity basis represents the transmissivity of each MZI.

To make our explanation simpler, we will adopt the orthogonalization protocol used by Ref. [202]. In this representation, we define the triangular mesh U_T as

$$\begin{aligned} U_T &= \prod_{m=0}^{N-1} U^{(N-m)} \\ U^{(m)} &= \prod_{n=1}^{m-1} U_{N-n}(\theta_{N-n}^{(m)}, \phi_{N-n}^{(m)}) \cdot D_m(\gamma_{N-m+1}), \end{aligned} \tag{A.4}$$

where D_m is a diagonal matrix representing a single mode phase shift at index $N - m + 1$.

The N operators $U^{(m)}$ represent the diagonal layers of the triangular mesh and their role is to project inputs from Hilbert space dimension from m to $m - 1$ recursively until we reach a single mode phase shift in $U^{(1)} = D_1(\gamma_N)$. Our proof moves the same direction as Reck’s orthogonalization procedure; starting from $m = N$, we progressively find the for each $U^{(m)}$ in decreasing order. For each layer m , there are $2m - 1$ complex hyperspherical coordinates ($m - 1$ “amplitude” coordinates

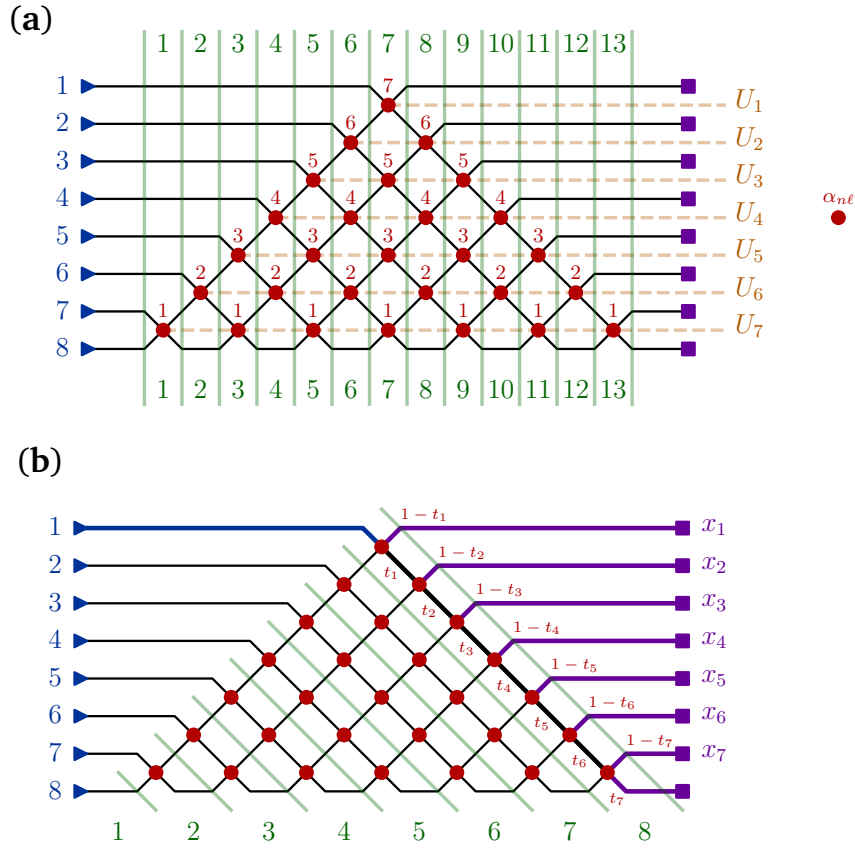


Figure A.1: Triangular mesh for $N = 8$ using (a) $2N - 3$ vertical layers ℓ showing the sensitivity index $\alpha_{n\ell}$ and (b) N diagonal layers m showing the transmissivity basis (t_n in red) and the measurement basis (x_n in purple).

and m “phase” coordinates). The first column vector of U can be recovered by shining light (using a unit power $P = 1$) through the top port of the layer (given by $n = N - m + 1$) and measuring the output fields in the triangular mesh generated by $U^{(m)}$, as shown in Figure A.1(b). As mentioned in Refs. [202, 157], progressive optimization moves in the opposite direction; the desired output fields are shined back into the device and the transmissivities $t_n^{(m)}$ and phases $\phi_n^{(m)}$ for each layer m (moving from N to 1) can be progressively tuned until all the power lies in the top input port for that layer.

The measurement basis is an unbiased Haar measure (as shown in Ref. [212] using Gaussian random vectors) and can be physically represented by the power x_n measured at waveguides $n \leq m - 1$ due to shining light through the top input port for that layer. Unlike the proof in Ref. [212], we choose our constraint that the input power $P = 1$ rather than $P \in \mathbb{R}^+$, which introduces a

normalization prefactor in our Haar measure by integration over all possible P .¹ This allows us to ignore the power in the final output port x_N because energy conservation ensures we have the constraint $x_N = 1 - \sum_{n=1}^{N-1} x_n$. Therefore, our simplified Cartesian basis for each m is (ignoring the normalization prefactor):

$$d\mu(U^{(m)}) \propto d\gamma_{N-m} \prod_{n=1}^{m-1} dx_n \prod_{n=1}^m d\phi_n. \quad (\text{A.5})$$

Now we represent the Cartesian power quantities x_n explicitly in terms of the component transmissivities, which we have defined already to be $t_n := \cos^2(\theta_n/2)$. Using the same convention as hyperspherical coordinates, we get the following recursive relation for x_n as shown diagrammatically by following the path of light from the top input port in Figure A.1(b):

$$x_n = (1 - t_n) \prod_{k=1}^{n-1} t_k. \quad (\text{A.6})$$

Intuitively, Equation A.6 implies that the power x_n measured at port n is given by light that is transmitted by the first $n - 1$ components along the path of light and then reflected by the n th component. In other words, x_n follows a geometric distribution.

We can use Equation A.6 to find the Jacobian $\mathcal{J} \in \mathbb{R}^{N-1 \times N-1}$ relating the x_n and the t_n . We find that we have a lower triangular matrix \mathcal{J} with diagonal elements for $n \leq N - 1$

$$\mathcal{J}_{nn} = \frac{\partial x_n}{\partial t_n} = - \prod_{k=1}^{n-1} t_k. \quad (\text{A.7})$$

We know \mathcal{J} is lower triangular since for all $n' > n$, $\mathcal{J}_{nn'} = \frac{\partial x_n}{\partial t_{n'}} = 0$ from Equation A.6.

Since the determinant of a lower triangular matrix is the same as the product of the diagonal, we can directly evaluate the unbiased measure (off by a normalization constant) as

$$\begin{aligned} d\mu(U^{(m)}) &\propto d\gamma_{N-m+1} \det \mathcal{J} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \\ &= d\gamma_{N-m+1} \prod_{n=1}^{m-1} \mathcal{J}_{nn} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \\ &\propto d\gamma_{N-m+1} \prod_{n=2}^{m-1} t_{n-1}^{m-n} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \end{aligned} \quad (\text{A.8})$$

To get the total Haar measure, we multiply the volume elements for the orthogonal components $d\mu(U^{(m)})$. We get from this procedure that the sensitivity index $\alpha_{nl} = N - n$ for a triangular mesh

¹This prefactor is exactly $\int_0^\infty dP e^{-P} P^{m-1} = (m-1)!$.

in Equation A.8 (independent of ℓ), which can be seen using Figure A.1. We can express this Haar measure in terms of $\mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell})$, the probability distribution for the transmissivity, and $\mathcal{P}_{\alpha_{n\ell}}(\theta_{n\ell}/2)$, the probability distribution for the phase shift corresponding to that same transmissivity, assuming appropriate choice n, ℓ for the triangular mesh:

$$\begin{aligned} d\mu(U) &= \prod_{n=1}^N d\mu(U^{(n)}) \\ &= \prod_n d\gamma_n \prod_{n,\ell} \mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell}) dt_{n\ell} d\phi_{n\ell} \\ &= \prod_n d\gamma_n \prod_{n,\ell} \mathcal{P}_{\alpha_{n\ell}}\left(\frac{\theta_{n\ell}}{2}\right) d\theta_{n\ell} d\phi_{n\ell} \end{aligned} \quad (\text{A.9})$$

We can now normalize Equation A.8 using the normalization factor for P to get $\mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell})$ and then substitute $t_{n\ell} = \cos^2(\theta_{n\ell}/2)$ to get our desired expression for $\mathcal{P}_{\alpha_{n\ell}}(\theta_{n\ell}/2)$:

$$\begin{aligned} \mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell}) &= \alpha_{n\ell} t_{n\ell}^{\alpha_{n\ell}-1} \\ \mathcal{P}_{\alpha_{n\ell}}\left(\frac{\theta_{n\ell}}{2}\right) &= \alpha_{n\ell} \sin\left(\frac{\theta_{n\ell}}{2}\right) \left[\cos\left(\frac{\theta_{n\ell}}{2}\right)\right]^{2\alpha_{n\ell}-1}. \end{aligned} \quad (\text{A.10})$$

Finally, we can recover the Haar phase parameter $\xi_{n\ell} \in [0, 1]$ (i.e. the cumulative density function) in terms of either $t_{n\ell}$ or $\theta_{n\ell}$:

$$\xi_{n\ell} = \left[\cos\left(\frac{\theta_{n\ell}}{2}\right)\right]^{2\alpha_{n\ell}} = t_{n\ell}^{\alpha_{n\ell}}. \quad (\text{A.11})$$

Finally, as explained in Ref. [212], we can use the Clements decomposition [50] to find another labelling for $\alpha_{n\ell}$ in a rectangular mesh that gives probability distributions and Haar phases in the same form as Equations A.10 and A.11 respectively.

A.3 Unitary matrix bandsizes

We would like to quantify the bandedness of matrices implemented by the meshes with randomly initialized phases. We define the η -bandsize as the minimum number of matrix elements whose absolute value squared sums to $(1 - \eta)N$. Note that our η -bandsize measurement is agnostic of the ordering of the inputs and outputs, and is therefore agnostic to any permutations that may be applied at the end of the decomposition. In photonics terms, if $\eta = 0.001$, let r_i measure the fraction of output waveguides over which 99.9% of the power is distributed when light is input into waveguide i . The η -bandsize is r_i averaged over all i . Sampling from our matrix distributions, we observe the relationship between the bandsize (given $\eta = 0.001$) and the dimension N in Figure A.2.

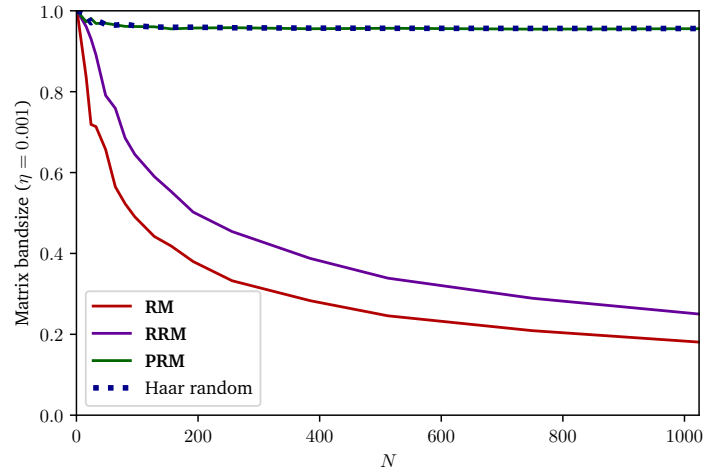


Figure A.2: Given $\eta = 0.001$, we compare bandsizes for rectangular ($U \sim \mathcal{U}_R(N, N)$), permuting rectangular ($U \sim \mathcal{U}_{PR}(N)$), and redundant meshes ($U \sim \mathcal{U}_R(N, 2N)$). Permuting rectangular meshes match the bandsize of Haar random matrices.

A.4 Fabrication imperfections in a redundant mesh

When photonic errors are added to the redundant mesh, specifically the 256-layer mesh, we observe a slight decrease in optimization performance in Figure A.3, similar to what we observed for the rectangular and permuting rectangular meshes in Figure 3.6. This decrease in performance, however, is less concerning considering that we still achieve a mean square error of around 10^{-5} , suggesting that RRM might be more robust to photonic errors even during on-chip optimization.

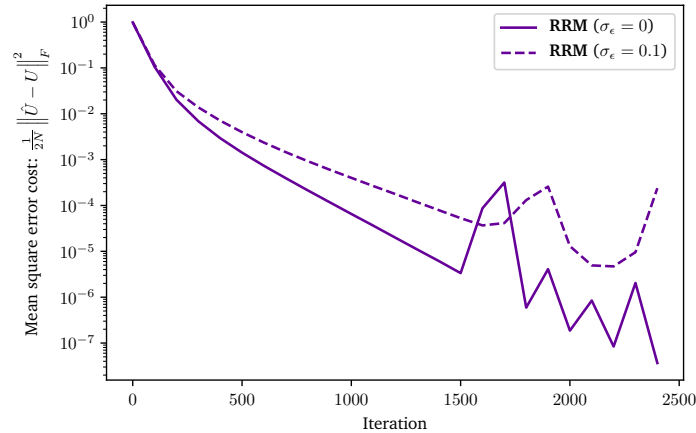


Figure A.3: A comparison of test mean square error for $N = 128$ between redundant rectangular meshes with error ϵ for 256-layer mesh for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256, simulated in `tensorflow`.

A.5 Optical singular value decomposition simulations

We compare the simulated performance of such rectangular and permuting rectangular architectures in the singular value decomposition (SVD) configuration discussed in Refs. [157, 225]. Such architectures would allow one to perform arbitrary linear operations with a relatively small footprint, and may have some other useful dimensionality-reduction properties in machine learning contexts.

In SVD, we represent complex matrix $\hat{A} \in \mathbb{C}^M \times \mathbb{C}^N$ as $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^\dagger$, where $\hat{\Sigma}$ is a diagonal matrix implemented on-chip with $\min(M, N)$ single-mode gain or attenuating elements and \hat{U}, \hat{V}^\dagger are unitary matrices implemented in a photonic mesh. While \hat{A} has $2MN$ free parameters, any global optimization for a photonic SVD implementation using rectangular meshes can have at most $D = N(N - 1) + M(M - 1) + 2\min(N, M) \geq 2MN$ free parameters, with equality when $M = N$. In the triangular architecture discussed in Ref. [157], the total complexity of parameters can be exactly $D = 2MN$ when setting a subset of the beamsplitters to bar state. In the case where the total number of singular values for \hat{A} is $S < \min(M, N)$, we get $D = 2S(M + N - S)$ tunable elements. Additionally, there is an “effective redundancy” in that some vectors in U, V are more important than others due to the singular values.

In our simulations, we investigate an SVD architecture for $A = U\Sigma V^\dagger$ for $A \in \mathbb{C}^M \times \mathbb{C}^N$ composed of the unitaries $U \in \mathbb{C}^M \times \mathbb{C}^M$ and $V \in \mathbb{C}^N \times \mathbb{C}^N$. Note that such an architecture is redundant when $M \neq N$, so we focus on the simple case of $M = N = 64$.

We define our train and test cost functions analogous to the unitary mean-squared error cost functions as

$$\begin{aligned} \mathcal{L}_{\text{test}} &= \frac{N\|\hat{A} - A\|_F^2}{2\|A\|_F^2} \\ \mathcal{L}_{\text{train}} &= \|\hat{A}X - AX\|_F^2, \end{aligned} \tag{A.12}$$

where $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^\dagger$ is defined in Section 3.5.

We randomly generate $A \in \mathbb{C}^N \times \mathbb{C}^M$ by expressing $A_{jk} = a + ib$, where $a, b \sim \mathcal{N}(0, 1)$. The synthetic training batches of unit-norm complex vectors are represented by $X \in \mathbb{C}^{N \times 2N}$.

Assuming a procedure similar to [102] can be used in presence of gains and optimization, the permuting rectangular mesh converges slightly faster but is significantly more resilient to uniform random phase initialization compared to the rectangular mesh as shown in Figure A.4. Both optimizations are minimally affected by beamsplitter error, unlike what is seen in the unitary optimization case.

A.6 Periodic parameters

We comment on our reported values of $\theta_{n\ell}$ in the checkerboard plots in Figures 3.3 (of the main text) and A.5. Since our simulated optimization does not have the explicit constraint that $\theta_{n\ell} \in [0, \pi)$, we report the “absolute $\theta_{n\ell}$,” where we map all values of $\theta_{n\ell}/2$ to some value in $[0, \pi/2]$. This

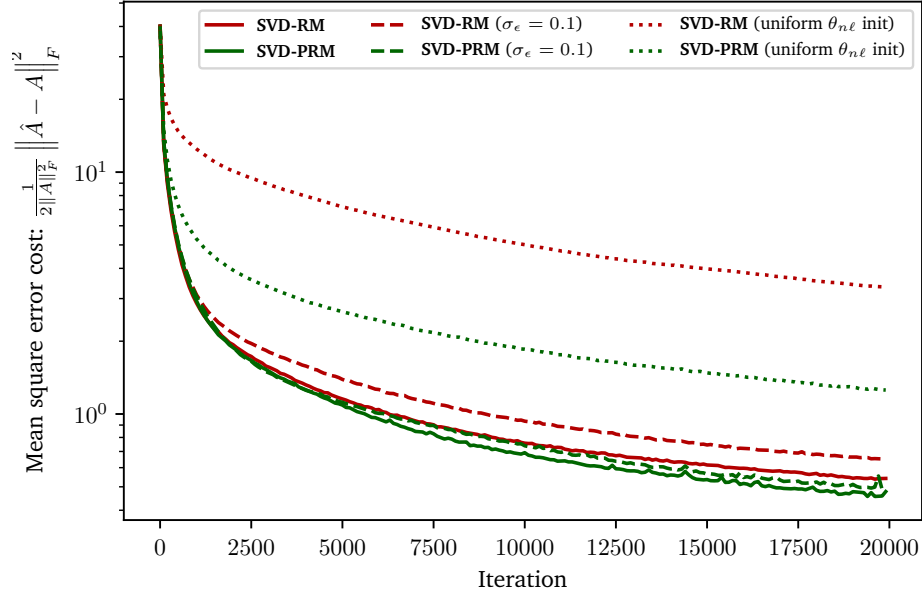


Figure A.4: A comparison of test mean square error for $N = 64$ between SVD devices using rectangular (SVD-RM) and permuting rectangular (SVD-PRM) meshes for: 20000 iterations, Adam update, learning rate of 0.005, batch size of 128, simulated in `tensorflow`. Unless otherwise noted, the default setting is Haar random initialized θ_{nl} with $\sigma_\epsilon = 0$.

corresponds to the transformation (assuming θ_{nl} is originally between 0 and 2π):

$$\theta_{nl} \rightarrow \begin{cases} \theta_{nl} & \theta_{nl} \leq \pi \\ 2\pi - \theta_{nl} & \theta_{nl} > \pi \end{cases}. \quad (\text{A.13})$$

Note a similar treatment as Equation A.13 can be used to represent the Haar phase $\xi \in [0, 1]$ in terms of a “periodic” Haar phase $\tilde{\xi} \in [0, 2]$ with period 2:

$$\xi(\tilde{\xi}) = \begin{cases} \tilde{\xi} & \tilde{\xi} \leq 1 \\ 2 - \tilde{\xi} & \tilde{\xi} > 1 \end{cases}. \quad (\text{A.14})$$

Note both $\tilde{\xi}$ and $\tilde{\theta}$ can therefore be made to vary continuously from $(-\infty, \infty)$ with $\tilde{\xi}$ having a period of 2 and $\tilde{\theta}$ having a period of 2π . We map these periodic parameters to their half-periods according to Equations A.13 and A.14 based on symmetry arguments.

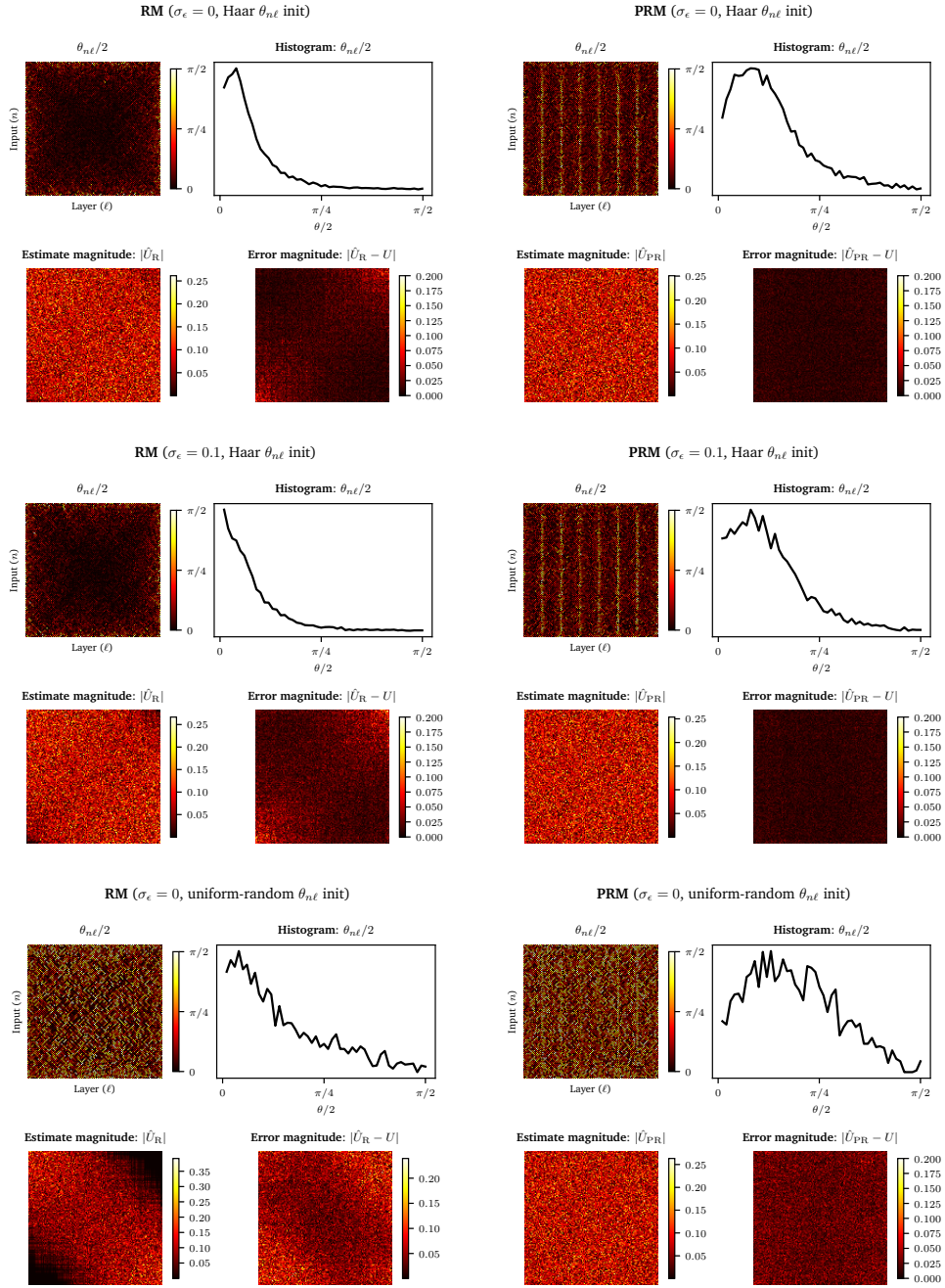


Figure A.5: Comparison of learned matrix errors and learned $\theta_{n\ell}$ weights after 20000 iterations for the Adam update at learning rate 0.0025 and batch size 256 for the simple unitary network. We consider two meshes: (1) rectangular mesh (RM), and (2) permuting rectangular mesh (PRM). We consider three conditions for each mesh: (1) ideal (with Haar random unitary initialization), (2) photonic beamsplitter error displacement $\epsilon \sim \mathcal{N}(0, 0.01)$, (3) random initialization.

A.7 Training simulation comparisons

In Figure A.5, we compare the performance for our unitary network experiment over our aforementioned conditions in Section 3.5. For each plot, we also have an associated video, showing how the parameter distributions, estimates, and errors vary during the course of the optimization, available online.²

There are several takeaways from these plots. First, the reflectivity of the MZIs near the center of the mesh are much smaller in the optimized rectangular meshes than in the permuting rectangular meshes. Second, the gradient descent algorithm has a hard time finding the regime of Haar random matrices after a uniform random phase initialization. The values of $\theta_{n\ell}$ are much larger than they need to be even 100 iterations into the optimization. This is likely evidence of a “vanishing gradient” problem when the mesh is not Haar-initialized. Finally, an important observation for the meshes with beamsplitter error is that the $\theta_{n\ell}/2$ distribution shifts slightly towards 0 in the rectangular mesh. This is a consequence of the limits in reflectivity and transmissivity in each MZI due to beamsplitter fabrication error as discussed in Section 3.2.

Our simulated permuting rectangular implementation uses the same layer definitions as defined in Equation 3.11 except the P_k with the most layers are in the center of the mesh, and the P_k with the fewest layers are near the inputs and outputs of the mesh. In Figure 3.4, P_2 and P_3 would be switched, and for $N = 128$, the order is $[P_2, P_4, P_6, P_5, P_3, P_1]$. We find this configuration to be the best permuting rectangular mesh so far in our experiments, although the architecture in Equation 3.11 gives improvements over the rectangular mesh.

A.8 An equivalent definition for $\alpha_{n\ell}$

Let $\alpha_{n\ell}$ be the sensitivity index for an MZI (“node”) at (waveguide, layer) coordinates (n, ℓ) in a local decomposition for an $N \times N$ unitary operator. We define the “row coordinate” or waveguide index n from the MZI’s operator U_n coupling waveguides n and $n + 1$, and we define the “column coordinate” or layer index m to be $\ell = k + 1$, where k is the maximum number of operators applied to a reachable input (This is equivalent to the vertical layers definition in Figure 3.1.). The reachable inputs $I_{n\ell}$ are the subset of input modes affecting the immediate inputs of the MZI at (n, ℓ) , and the reachable outputs $O_{n\ell}$ are the subset of output modes affected by the immediate outputs of the MZI.

Following the definitions in Ref. [212], in the triangular scheme, $\alpha_{n\ell} := N - n$, and in the rectangular scheme, $\alpha_{n\ell} := d(n, \ell) + 1 - s_{n\ell}[\ell]$ where $d(n, \ell)$ is the number of nodes on the diagonal (measured along paths of constant $n + \ell$) containing a rotation parameterized by $\theta_{n\ell}$, and $s_{n\ell}$ is a sequence of decreasing odd integers $d(n, \ell) \geq k_{\text{odd}} \geq 1$, followed by increasing even integers

²See <https://av.tib.eu/series/520/photonic+optimization>.

$2 \leq k_{\text{even}} \leq d(n, \ell)$, as defined in [212]. We prove below that for both the triangular and rectangular meshes, $\alpha_{n\ell} = |I_{n\ell}| + |O_{n\ell}| - N - 1$.

Lemma 1. *In the triangular mesh, $\alpha_{n\ell} = |I_{n\ell}| + |O_{n\ell}| - N - 1$.*

Proof. In the triangular mesh (shown for $N = 8$ in Figure A.1) $\alpha_{n\ell} := N - n$, so we wish to show that $N - n = |I_{n\ell}| + |O_{n\ell}| - N - 1$, or:

$$2N + 1 = |I_{n\ell}| + |O_{n\ell}| + n. \quad (\text{A.15})$$

Suppose Equation A.15 holds for some arbitrary n', ℓ' in the mesh, such that $2N + 1 = |I_{n'\ell'}| + |O_{n'\ell'}| + n'$. First, induct on n : if we take $n = n' + 2$ and $\ell = \ell'$, then $|I_{n\ell}| = |I_{n'\ell'}| - 1$ and $|O_{n\ell}| = |O_{n'\ell'}| - 1$. Next, induct on ℓ : if we take $n = n'$ and $\ell = \ell' + 2$, then $|I_{n\ell}| = |I_{n'\ell'}| + 1$ and $|O_{n\ell}| = |O_{n'\ell'}| - 1$. In both cases, Equation A.15 holds.

Traversals by 2 along n or ℓ from a starting node can reach all nodes with the same parity of n and ℓ , so we need two base cases. Consider the apex node at $n = 1, \ell = N - 1$ and one of its neighbors at $n = 2, \ell = N$. The former has $|I_{n\ell}| = |O_{n\ell}| = N$ and the latter has $|I_{n\ell}| = N$ and $|O_{n\ell}| = N - 1$. In both cases, Equation A.15 is satisfied, so the lemma holds by induction. \square

Lemma 2. *In the rectangular mesh, $\alpha_{n\ell} = |I_{n\ell}| + |O_{n\ell}| - N - 1$.*

Proof. In the rectangular mesh, $\alpha_{n\ell} := d(n, \ell) + 1 - s_{n\ell}[\ell]$, as defined in Ref. [212]. Define orthogonal axes x and y on the lattice such that for a node at (n, ℓ) , traveling in the $+x$ direction gives the neighboring node at $(n + 1, \ell + 1)$ and traveling in the $+y$ direction gives the neighboring node at $(n - 1, \ell + 1)$, as depicted in Figure A.6. For even $\{\text{odd}\} N$, let the node at $(n, \ell) = (1, 1)$ have $x = 1$ and the node at $(n, \ell) = (N - 1, 1\{2\})$ have $y = 1$. Then there is a one-to-one mapping such that $(x, y) = (\frac{n+\ell}{2}, \frac{\ell-n}{2} + \lfloor \frac{N}{2} \rfloor)$, as shown in Figure A.6, and it suffices to prove the lemma by induction in this diagonal basis.

Since $d(n, \ell)$ is defined to be the length of a diagonal along paths of constant $n + \ell$, it depends only on x , so we rewrite $d(n, \ell) \mapsto d(x)$ explicitly:

$$d(x) = \begin{cases} 2x - 1 & x \leq \lfloor \frac{N}{2} \rfloor \\ 2(N - x) & x > \lfloor \frac{N}{2} \rfloor \end{cases}. \quad (\text{A.16})$$

Similarly, since $s_{n\ell}[\ell]$ is enumerated along a diagonal, it depends only on y , and we convert $s_{n\ell}[\ell] \rightarrow s_x[y]$ from the sequence definition of Ref. [212] to an explicit lattice form:

$$s_x[y] = \begin{cases} 2(\lfloor \frac{N}{2} \rfloor - y) + 1 & y \leq \lfloor \frac{N}{2} \rfloor \\ 2(y - \lfloor \frac{N}{2} \rfloor) & y > \lfloor \frac{N}{2} \rfloor \end{cases}. \quad (\text{A.17})$$

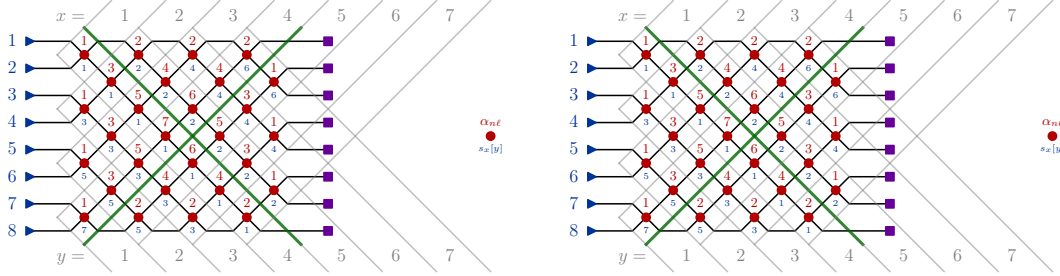


Figure A.6: Rectangular decomposition for even ($N = 8$) and odd ($N = 7$) meshes, showing the diagonal x, y basis. Values for $\alpha_{n\ell}$ are shown in red above each MZI, with values for $s_x[y]$ shown in blue below. The critical boundaries of $x, y = \lfloor \frac{N}{2} \rfloor$ separating the different quadrants are drawn in green. (Boundaries are offset for visual clarity.)

In this diagonal basis, we want to show that

$$d(x) + 1 - s_x[y] = |I_{xy}| + |O_{xy}| - N - 1. \quad (\text{A.18})$$

There are two boundaries at $x, y = \lfloor \frac{N}{2} \rfloor$ which separate four quadrants that must be considered, depicted by gray lines in Figure A.6. We will induct on x and y within each quadrant, then induct on x or y across each of the two boundaries.

Suppose that Equation A.18 holds for some arbitrary $x'y'$ in the mesh, such that $d(x') + 1 - s_{x'}[y'] = |I_{x'y'}| + |O_{x'y'}| - N - 1$. First, we induct on x and y within each quadrant; the results are tabulated in Table A.1. In every case, $d(x) - s_x[y] - |I_{xy}| - |O_{xy}| = d(n, \ell) - s_{x'}[y'] - |I_{x'y'}| - |O_{x'y'}|$, so Equation A.18 remains satisfied.

Quadrant	Induction	$d(x) = \dots$	$s_x[y] = \dots$	$ I_{xy} = \dots$	$ O_{xy} = \dots$
$x' \leq \lfloor \frac{N}{2} \rfloor, y' \leq \lfloor \frac{N}{2} \rfloor$	$x = x' - 1$	$d(n, \ell) - 2$	$s_{x'}[y']$	$ I_{x'y'} - 2$	$ O_{x'y'} $
	$y = y' - 1$	$d(n, \ell)$	$s_{x'}[y'] + 2$	$ I_{x'y'} - 2$	$ O_{x'y'} $
$x' \leq \lfloor \frac{N}{2} \rfloor, y' > \lfloor \frac{N}{2} \rfloor$	$x = x' - 1$	$d(n, \ell) - 2$	$s_{x'}[y']$	$ I_{x'y'} - 2$	$ O_{x'y'} $
	$y = y' + 1$	$d(n, \ell)$	$s_{x'}[y'] + 2$	$ I_{x'y'} $	$ O_{x'y'} - 2$
$x' > \lfloor \frac{N}{2} \rfloor, y' \leq \lfloor \frac{N}{2} \rfloor$	$x = x' + 1$	$d(n, \ell) - 2$	$s_{x'}[y']$	$ I_{x'y'} $	$ O_{x'y'} - 2$
	$y = y' - 1$	$d(n, \ell)$	$s_{x'}[y'] + 2$	$ I_{x'y'} - 2$	$ O_{x'y'} $
$x' > \lfloor \frac{N}{2} \rfloor, y' > \lfloor \frac{N}{2} \rfloor$	$x = x' + 1$	$d(n, \ell) - 2$	$s_{x'}[y']$	$ I_{x'y'} $	$ O_{x'y'} - 2$
	$y = y' + 1$	$d(n, \ell)$	$s_{x'}[y'] + 2$	$ I_{x'y'} $	$ O_{x'y'} - 2$

Table A.1: Induction on x and y within each of the quadrants in the mesh.

Next, we induct across the $x, y = \lfloor \frac{N}{2} \rfloor$ boundaries, shown in Table A.2. Again, in every case,

$d(x) - s_x[y] - |I_{xy}| - |O_{xy}| = d(n, \ell) - s_{x'}[y'] - |I_{x'y'}| - |O_{x'y'}|$, satisfying Equation A.18.

x'	y'	Induction	$d(x) = \dots$	$s_x[y] = \dots$	$ I_{xy} = \dots$	$ O_{xy} = \dots$
$x' = \lfloor \frac{N}{2} \rfloor$	any	$x = x' + 1$	$d(n, \ell) - \{+\}1$	$s_{x'}[y']$	$ I_{x'y'} + 0\{1\}$	$ O_{x'y'} - 1\{0\}$
any	$y' = \lfloor \frac{N}{2} \rfloor$	$y = y' + 1$	$d(n, \ell)$	$s_{x'}[y'] + 1$	$ I_{x'y'} $	$ O_{x'y'} - 1$

Table A.2: Induction on x or y across each of the borders of $x, y = \lfloor \frac{N}{2} \rfloor$.

Finally, note that the base case of the top left MZI at $(n, \ell) = (1, 1)$, $(x, y) = (1, \lfloor \frac{N}{2} \rfloor)$ holds, with $d(x) + 1 - s_x[y] = 1 = 2 + N - N - 1 = |I_{xy}| + |O_{xy}| - N - 1$. This completes the proof in the (x, y) basis, and since there is a one-to-one mapping between $(x, y) \leftrightarrow (n, \ell)$, $\alpha_{n\ell} = |I_{n\ell}| + |O_{n\ell}| - N - 1$ holds by induction. \square

Appendix B

Quantum programmable gate arrays

This appendix contains more detailed supplementary calculations for the results presented in Chapter 4. Section B.1 derives effective transfer operations for photons with arbitrary spectra, Section B.2 contains a detailed derivation for the reflection coefficients of the strongly-coupled quantum emitter, and Section B.3 contains phase shifter parameters to implement a variety of common single-qubit quantum gates, as well as embeddings of multi-qubit gates within the QPGA architecture.

B.1 Phase-modulated interference for photons with arbitrary spectra

Consider a Mach-Zehnder interferometer with four phase shifters in the arrangement presented in Figure 4.1c. Let the operators $\hat{a}_1^\dagger(\omega), \hat{a}_2^\dagger(\omega)$ represent creation operators for the top and bottom waveguides, respectively, acting on a single frequency mode ω . Consider an input state to the MZI representing a single logical qubit in the state $\alpha|0_L\rangle + \beta|1_L\rangle$:

$$|\psi^{\text{in}}\rangle = \int d\omega \phi(\omega) \left(\alpha \hat{a}_1^\dagger(\omega) + \beta \hat{a}_2^\dagger(\omega) \right) |\emptyset\rangle. \quad (\text{B.1})$$

The phase shifters in the MZI act by imparting a time delay τ on the creation operators, mapping $\hat{a}^\dagger(\omega) \mapsto \hat{a}^\dagger(\omega)e^{i\omega\tau}$. (Here we make the approximation that the phase shifter imparts an equal time delay across the range of frequencies of the photon, e.g. has a constant refractive index.) Let $\{\tau_\zeta, \tau_\xi, \tau_\theta, \tau_\phi\} \equiv \{\zeta, \xi, \theta, \phi\}/\omega_0$ denote the effective time delays imparted by the four phase shifters, where ω_0 denotes the 4LS resonance frequency ω in the main text. The idealized action of the MZI on photons of zero spectral width described in Eq. 4.1 is $R_\beta^\alpha H R^\theta H R^\phi$. In the case of finite spectral

width, the transformation maps:

$$\begin{bmatrix} \hat{a}_1^\dagger(\omega) \\ \hat{a}_2^\dagger(\omega) \end{bmatrix} \leftarrow \frac{1}{2} \begin{bmatrix} e^{i(\tau_\zeta + \tau_\phi)\omega} (e^{i\tau_\theta\omega} + 1) & e^{i(\tau_\xi + \tau_\phi)\omega} (e^{i\tau_\theta\omega} - 1) \\ e^{i\tau_\zeta\omega} (e^{i\tau_\theta\omega} - 1) & e^{i\tau_\xi\omega} (e^{i\tau_\theta\omega} + 1) \end{bmatrix} \begin{bmatrix} \hat{a}_1^\dagger(\omega) \\ \hat{a}_2^\dagger(\omega) \end{bmatrix}. \quad (\text{B.2})$$

Thus, the output state of the MZI is:

$$|\psi^{\text{out}}\rangle = \frac{1}{2} \int d\omega \phi(\omega) \left[\left(\alpha e^{i(\tau_\zeta + \tau_\phi)\omega} (e^{i\tau_\theta\omega} + 1) + \beta e^{i(\tau_\xi + \tau_\phi)\omega} (e^{i\tau_\theta\omega} - 1) \right) \hat{a}_1^\dagger(\omega) \right. \\ \left. + \left(\alpha e^{i\tau_\zeta\omega} (e^{i\tau_\theta\omega} - 1) + \beta e^{i\tau_\xi\omega} (e^{i\tau_\theta\omega} + 1) \right) \hat{a}_2^\dagger(\omega) \right] |\emptyset\rangle. \quad (\text{B.3})$$

Define coefficients $C_0(\omega) \equiv \frac{1}{2}(\alpha e^{i(\tau_\zeta + \tau_\phi)\omega} (e^{i\tau_\theta\omega} + 1) + \beta e^{i(\tau_\xi + \tau_\phi)\omega} (e^{i\tau_\theta\omega} - 1))$ and $C_1(\omega) \equiv \frac{1}{2}(\alpha e^{i\tau_\zeta\omega} (e^{i\tau_\theta\omega} - 1) + \beta e^{i\tau_\xi\omega} (e^{i\tau_\theta\omega} + 1))$. Then the output state is $|\psi^{\text{out}}\rangle = \int d\omega \phi(\omega) (C_0(\omega) \hat{a}_1^\dagger(\omega) + C_1(\omega) \hat{a}_2^\dagger(\omega)) |\emptyset\rangle$. Define projection operators \hat{P}_0, \hat{P}_1 which map physical wavefunctions to logical state vectors:

$$\hat{P}_0 = \int d\omega |0_L\rangle \langle \emptyset| \hat{a}_1(\omega) \quad (\text{B.4})$$

$$\hat{P}_1 = \int d\omega |1_L\rangle \langle \emptyset| \hat{a}_2(\omega) \quad (\text{B.5})$$

To obtain the fidelity of the physical output state against the target logical output state $|\psi^{\text{targ}}\rangle = C_0^L |0\rangle + C_1^L |1\rangle$, we evaluate the inner product between the states by expanding in terms of the complete basis $\mathbf{1} = \hat{P}_0 + \hat{P}_1$:

$$\begin{aligned} \mathcal{F} &= |\langle \psi^{\text{targ}} | \psi^{\text{out}} \rangle|^2 \\ &= \left| (C_0^{L*} \langle 0_L | + C_1^{L*} \langle 1_L |) \int d\omega \phi(\omega) (C_0(\omega) \hat{a}_1^\dagger(\omega) + C_1(\omega) \hat{a}_2^\dagger(\omega)) |\emptyset\rangle \right|^2 \\ &= \left| (C_0^{L*} \langle 0_L | + C_1^{L*} \langle 1_L |) (\hat{P}_0 + \hat{P}_1) \int d\omega \phi(\omega) (C_0(\omega) \hat{a}_1^\dagger(\omega) + C_1(\omega) \hat{a}_2^\dagger(\omega)) |\emptyset\rangle \right|^2 \\ &= \left| \int d\omega \phi(\omega) (C_0^{L*} C_0(\omega) + C_1^{L*} C_1(\omega)) \right|^2. \end{aligned} \quad (\text{B.6})$$

The fidelity of the output state will depend on the phase shifter values. We numerically simulate the output wavefunctions for a large sample of ζ, ξ, θ, ϕ across a range of spectral widths and plot the results in Figure B.1.

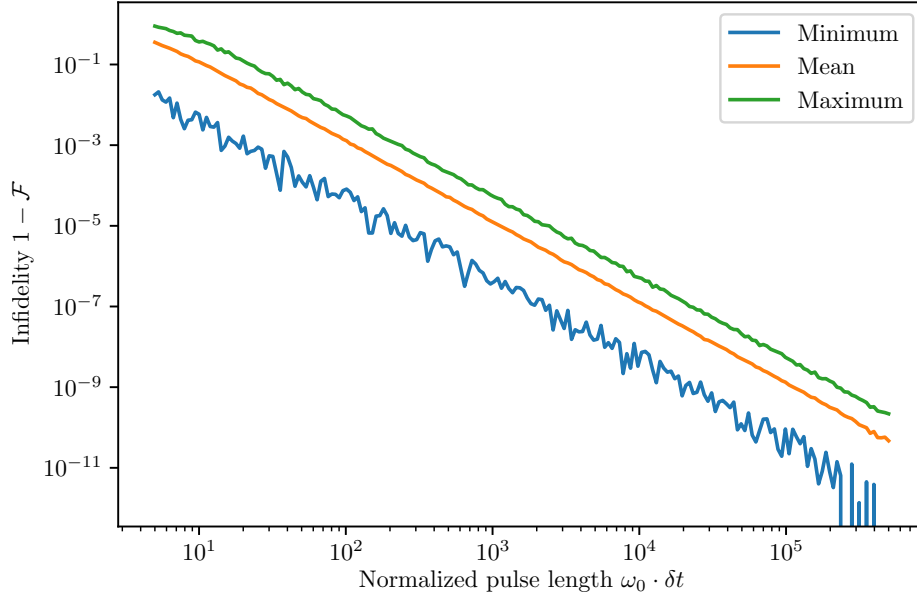


Figure B.1: Infidelities of the output state of a Mach-Zehnder interferometer for a range of spectral distributions. We assume an input wavefunction of $|\psi^{\text{in}}\rangle = \int d\omega g_{\delta t}(\omega) \frac{1}{\sqrt{2}}(\hat{a}_1^\dagger(\omega) + \hat{a}_2^\dagger(\omega)) |\emptyset\rangle$, where $g_\sigma(\omega)$ is a Gaussian with a spectral width of $\sigma = \frac{\delta\omega}{\omega_0}$ and a pulse length of $\delta t = \frac{1}{2\delta\omega}$ periods of the central frequency ω_0 . We compute the output wavefunction for an ensemble of 1000 values of ζ, ξ, θ, ϕ sampled uniformly from $[0, 2\pi)$ across 250 values of σ and plot the maximum, minimum, and average infidelity (defined as $1 - \mathcal{F}$) for each case, depicted as the bottom, middle, and top lines, respectively.

B.2 Derivation of reflection coefficients

In this section, we derive the reflection coefficients presented in Section 4.2.2, using a similar treatment of the problem as in Ref. [274]. To simplify the derivation, we replace the Hamiltonian in Eq. 4.6 with an *ad-hoc* Hamiltonian:

$$\begin{aligned} \mathcal{H}_{\text{ad-hoc}} = & \frac{\hbar}{i} \int dx \left[\hat{b}_R^\dagger(x) \frac{\partial}{\partial x} \hat{b}_R(x) - \hat{b}_L^\dagger(x) \frac{\partial}{\partial x} \hat{b}_L(x) \right] + \hbar \sum_{n=2}^4 \left(\Omega_n - \frac{i\Gamma'}{2} \right) |n\rangle\langle n| \\ & + \hbar \int dx \sqrt{\Gamma/2} \delta(x) \left[(\hat{b}_R^\dagger(x) + \hat{b}_L^\dagger(x)) (|1\rangle\langle 2| + |3\rangle\langle 2| + |3\rangle\langle 4|) + \text{H.c.} \right], \quad (\text{B.7}) \end{aligned}$$

where we have also set $v_g = v_r = 1$. With this approach, the Hilbert space contains only waveguide and atom states, without the environmental reservoir. This *ad-hoc* approach is known to produce correct scattering matrices for single-photon (and temporally-separated multi-photon) interactions, and is thus suitable for our purposes, but it should be noted that the direct substitution of $\Omega \rightarrow \Omega - i\Gamma'/2$ in the Hamiltonian rather than in the scattering matrix will yield incorrect results for temporally overlapping two-photon scattering. [207]

Step 1. Consider the dynamics of a single quantum emitter in the device, from sites (4a) to (6a) in Fig. 4.1c. Photon A at frequency $\omega = \Omega_{12} = \Omega_{34}$ is incident on the atom, which is initialized in state $|1\rangle$. The stationary state of the system is:

$$|\psi_1\rangle = \int dx \left[\phi_{1R}(x) \hat{b}_R^\dagger(x) + \phi_{1L}(x) \hat{b}_L^\dagger(x) \right] |\emptyset\rangle \otimes |1\rangle + e_2 |\emptyset\rangle \otimes |2\rangle \\ + \int dx \left[\phi_{3R}(x) \hat{b}_R^\dagger(x) + \phi_{3L}(x) \hat{b}_L^\dagger(x) \right] |\emptyset\rangle \otimes |3\rangle, \quad (\text{B.8})$$

where the amplitude of the ϕ wavepackets correspond to the component of the photon which is in the spatial mode being considered. [221, 274] Using the Schrodinger equation $\mathcal{H}|\psi_1\rangle = \hbar\omega|\psi_1\rangle$, where \mathcal{H} is given in Eq. B.7, and defining a coupling constant $V \equiv \sqrt{v_g\Gamma/2}$ we obtain:

$$\left(-i \frac{d}{dx} - \omega \right) \phi_{1R}(x) + V\delta(x)e_2 = 0, \quad (\text{B.9a})$$

$$\left(+i \frac{d}{dx} - \omega \right) \phi_{1L}(x) + V\delta(x)e_2 = 0, \quad (\text{B.9b})$$

$$\left(-i \frac{d}{dx} - \omega' \right) \phi_{3R}(x) + V\delta(x)e_2 = 0, \quad (\text{B.9c})$$

$$\left(+i \frac{d}{dx} - \omega' \right) \phi_{3L}(x) + V\delta(x)e_2 = 0, \quad (\text{B.9d})$$

$$-\frac{i\Gamma'}{2}e_2 + V(\phi_{1R}(0) + \phi_{1L}(0) + \phi_{3R}(0) + \phi_{3L}(0)) = 0. \quad (\text{B.9e})$$

Defining $k \equiv \omega/c$ and $k' \equiv \omega'/c = \Omega_{32}/c$, and following the treatment in Ref. [221] and [274], we assume a solution ansatz of:

$$\phi_{1R}(x) = e^{+ikx} (\theta(-x) + \beta_{1R}\theta(x)), \quad (\text{B.10a})$$

$$\phi_{1L}(x) = e^{-ikx} (\alpha_{1L}\theta(-x) + \beta_{1L}\theta(x)), \quad (\text{B.10b})$$

$$\phi_{3L}(x) = e^{-ik'x} (\beta_{3L}\theta(-x)), \quad (\text{B.10c})$$

$$\phi_{3R}(x) = e^{+ik'x} (\beta_{3L}\theta(-x) + \alpha_{3R}\theta(x)), \quad (\text{B.10d})$$

where θ is the Heaviside function with $\theta(0) \equiv \frac{1}{2}$. Here, β coefficients describe parts of the wavefunction between the relevant reflector and the 4LS, while α coefficients describe parts which are outside the 4LS (the input/output waveguide for the ω photon and the delay line for the ω' photon). The reversal of direction of x for ϕ_1 and ϕ_3 is due to the opposite orientation of the reflectors for ω and ω' , respectively. The reflective boundary conditions at $x = \pm a$ means that:

$$\phi_{1R}(a) + \phi_{1L}(a) = 0 = \phi_{3L}(-a) + \phi_{3R}(-a). \quad (\text{B.11})$$

Using this and substituting equations B.10 into B.9 gives us the solution:

$$r_{11} = \alpha_{1L} = e^{2i\omega a} \frac{\frac{i\Gamma'}{2} - \frac{i\Gamma}{2} (e^{2i\omega' a} - e^{-2i\omega a})}{-\frac{i\Gamma'}{2} + \frac{i\Gamma}{2} (e^{2i\omega' a} + e^{2i\omega a} - 2)}, \quad (\text{B.12})$$

$$r_{13} = \alpha_{3R} = \frac{\frac{i\Gamma}{2} (e^{2i\omega a} - 1) (e^{2i\omega' a} - 1)}{-\frac{i\Gamma'}{2} + \frac{i\Gamma}{2} (e^{2i\omega' a} + e^{2i\omega a} - 2)}. \quad (\text{B.13})$$

Step 2. We now send in the second photon B , also of frequency ω , which will scatter off of the $|1\rangle$ component of the 4LS state in the same manner as the first photon. We assume that the temporal separation of photons A and B is much greater than the decay timescale of the excited $|2\rangle, |4\rangle$ states, and since ω is off resonance from the $|3\rangle \leftrightarrow |2\rangle$ transition at ω' , then B will interact with the $|3\rangle \leftrightarrow |4\rangle$ transition only. The single photon scattering eigenstate for the $|3\rangle$ component of the 4LS state then takes the form:

$$|\psi_2\rangle = \int dx \left[\phi_{3R}(x) \hat{b}_R^\dagger(x) + \phi_{3L}(x) \hat{b}_L^\dagger(x) \right] |\emptyset\rangle \otimes |3\rangle + e_4 |\emptyset\rangle \otimes |4\rangle. \quad (\text{B.14})$$

As before, applying the *ad-hoc* Hamiltonian to $\mathcal{H}|\psi_2\rangle = \hbar\omega|\psi_2\rangle$, we obtain equations of motion:

$$\left(-i \frac{d}{dx} - \omega \right) \phi_{3R}(x) + V\delta(x)e_4 = 0, \quad (\text{B.15a})$$

$$\left(+i \frac{d}{dx} - \omega \right) \phi_{3L}(x) + V\delta(x)e_4 = 0, \quad (\text{B.15b})$$

$$-\frac{i\Gamma'}{2}e_4 + V(\phi_{3R}(0) + \phi_{3L}(0)) = 0. \quad (\text{B.15c})$$

Assuming a solution ansatz of

$$\phi_{3R}(x) = e^{+ikx} (\theta(-x) + \beta_{3R}\theta(x)), \quad (\text{B.16a})$$

$$\phi_{3L}(x) = e^{-ikx} (\alpha_{3L}\theta(-x) + \beta_{3L}\theta(x)), \quad (\text{B.16b})$$

where k is defined as before, and imposing reflective boundary conditions that $\phi_{3R}(a) + \phi_{3L}(a) = 0$, we obtain the reflected amplitude to be:

$$R_3 = \alpha_{3L} = \frac{\frac{i\Gamma'}{2}e^{2i\omega a} + \frac{i\Gamma}{2}(1 - e^{2i\omega a})}{-\frac{i\Gamma'}{2} - \frac{i\Gamma}{2}(1 - e^{2i\omega a})}. \quad (\text{B.17})$$

Step 3. The A' photon of frequency ω' has traveled down the delay line and back and is incident on the 4LS, which is in some superposition of $|1\rangle$ and $|3\rangle$. The photon is far off-resonance from the $|1\rangle \leftrightarrow |2\rangle$ transition, so will only interact with the $|3\rangle \leftrightarrow |2\rangle$ transition. Using the same approach as before, we obtain reflection amplitudes which are analogous to Eqs. B.12 and B.13, except with ω

and ω' switched:

$$r_{33} = e^{2i\omega'a} \frac{\frac{i\Gamma'}{2} - \frac{i\Gamma}{2} (e^{2i\omega a} - e^{-2i\omega'a})}{-\frac{i\Gamma'}{2} + \frac{i\Gamma}{2} (e^{2i\omega a} + e^{2i\omega'a} - 2)}, \quad (\text{B.18})$$

$$r_{31} = \frac{\frac{i\Gamma}{2} (e^{2i\omega'a} - 1) (e^{2i\omega a} - 1)}{-\frac{i\Gamma'}{2} + \frac{i\Gamma}{2} (e^{2i\omega a} + e^{2i\omega'a} - 2)}. \quad (\text{B.19})$$

Step 4. The B' photon of frequency ω' has returned to the 4LS, which is in some different superposition of $|1\rangle$ and $|3\rangle$. As before, the photon only interacts with the $|3\rangle \leftrightarrow |2\rangle$ transition, and has identical reflection coefficients as step 3.

B.3 Implementations of common quantum gates

Operator	Matrix representation	ζ	ξ	θ	ϕ
Identity	$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	0	0	0	0
Hadamard	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\frac{5\pi}{4}$	$\frac{3\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$
Pauli-X	$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	π	π	π	0
Pauli-Y	$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	π	0
Pauli-Z	$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	0	π	0	0
Rotation-X	$R_x(\theta') = \cos \frac{\theta'}{2} \mathbb{1} - i \sin \frac{\theta'}{2} \sigma_x$	$-2\theta'$	$-2\theta'$	θ'	0
Rotation-Y	$R_y(\theta') = \cos \frac{\theta'}{2} \mathbb{1} - i \sin \frac{\theta'}{2} \sigma_y$	$-2\theta' - \frac{\pi}{2}$	$-2\theta'$	θ'	$\frac{\pi}{2}$
Rotation-Z	$R_z(\theta') = \cos \frac{\theta'}{2} \mathbb{1} - i \sin \frac{\theta'}{2} \sigma_z$	$-\frac{\theta'}{2}$	$\frac{\theta'}{2}$	0	0
Phase shift	$R_{\phi'} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$	0	ϕ'	0	0

Table B.1: A table of phase shifter parameters which implement various common single-qubit gates on the phase-modulated MZIs depicted in Figure 4.1c.

Operator	Symbol	Decomposition	Cell depth
Identity			0, 2, 4, ...
Controlled-NOT			1
Controlled-phase			2
Controlled- U^1			2
SWAP			3

Table B.2: Construction of common multi-qubit gates by embedding single-qubit operations in a lattice of $c\sigma_z$ gates. Because the phase-modulated MZIs can implement any single-qubit operator, gate decompositions may be terminated with either with $c\sigma_z$ gates or with single-qubit gates, as the first layer of single-qubit operators of subsequent gates can implicitly include the final single-qubit operators of the previous logical gate. All quantum circuit diagrams in this paper were typeset using the `QCircuit` L^AT_EX package. [67]

Appendix C

Quantum computing in a synthetic time dimension

In this appendix, we give more detailed presentations of the results described in Chapter 5. In Section C.1 we present a derivation of the gate teleportation mechanism; in Section C.2 we derive a method to construct arbitrary single-qubit operations from the teleported gates; in Section C.3 we construct a photon-atom SWAP operation from scattering sequences and measurement; in Section C.4 we describe constructions for a two-photon $c\sigma_z$ gate; in Section C.5 we give more detail of the circuit compilation process and provide an example of a compiled instruction sequence to implement a quantum Fourier transform on our proposed device; and in Section C.7 we discuss in greater detail the imperfection analysis described Section 5.3.

C.1 Derivation of gate teleportation mechanism

Consider a photon which is circulating in the storage ring in the storage ring in a state $|\psi_{\text{in}}\rangle = \alpha|\circlearrowleft\rangle + \beta|\circlearrowright\rangle$, where $|\circlearrowleft\rangle$ and $|\circlearrowright\rangle$ denote the two counter-circulating states. Referring to Figure C.1, define bosonic operators $\hat{a}_{\circlearrowleft}^\dagger(t)$, $\hat{a}_{\circlearrowright}^\dagger(t)$ which create at time t a clockwise- or counterclockwise-propagating photon in the ring at the points P_1 , P_2 , respectively, just before the switches. The physical state of the photon in the ring can be written as

$$|\psi_{\text{in}}\rangle = \int dt \phi(t) \left[\alpha \hat{a}_{\circlearrowleft}^\dagger(t) + \beta \hat{a}_{\circlearrowright}^\dagger(t) \right] |\emptyset\rangle, \quad (\text{C.1})$$

where $|\emptyset\rangle$ denotes the vacuum state and $\phi(t)$ describes the pulse envelope. Here we assume that the photon was originally injected in the $|\circlearrowleft\rangle$ state as shown in Figure C.1 and has undergone at most a small number of scattering interactions with the atom-cavity system, such that the clockwise and

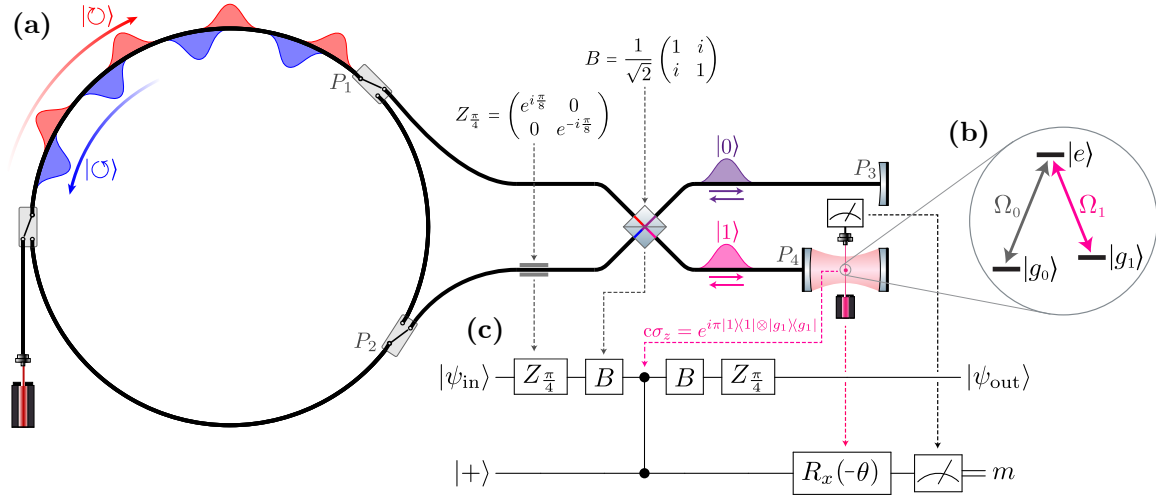


Figure C.1: An annotated figure depicting the architecture described in the main text and the correspondence of physical and logical circuit elements. **(a)** The physical design of the device, with annotations indicating quantum operations implemented by physical circuit elements. **(b)** The energy structure of the atom: Ω_1 is resonant with the cavity mode and photon carrier frequency, while Ω_0 is far-detuned. **(c)** Gate diagram of the quantum circuit applied in a single pass of a photonic qubit through the scattering unit. The top rail denotes the state of the photonic qubit and the bottom rail denotes the atomic qubit. After the photon returns to the storage ring, $R_x(-\theta)$ is applied to the atomic qubit and a projective measurement of the atomic state is performed. The final output state $|\psi_{\text{out}}\rangle$ is $Z_{\frac{\pi}{4}} \sigma_z (-\sigma_y)^{m \oplus 1} R_y(\theta) Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle$, as described in Eq. 2 of the main text.

counterclockwise pulses have not independently deformed significantly and can be described by a single envelope.

We also define bosonic operators $\hat{b}_{0,d}^\dagger(t)$, $\hat{b}_{1,d}^\dagger(t)$ which respectively create a photon in the top or bottom waveguides at points P_1, P_2 at time t propagating with direction $d \in \{L, R\}$. As the photon is injected by the switches from the ring into the waveguides, the fixed $\pi/4$ phase shifter applies (up to a global phase) a rotation $Z_{\frac{\pi}{4}} \equiv R_z(\frac{\pi}{4}) = \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$ to the photon state, and the beamsplitter applies the operation $B = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$. Finally, let operators $\hat{c}_{0,d}^\dagger(t)$, $\hat{c}_{1,d}^\dagger(t)$ with $d \in \{L, R\}$ create a photon at time t in the top or bottom waveguides just before the mirror or cavity at points P_3 and P_4 .

The round trip distance from points P_1, P_2 to P_3, P_4 and back is equal to the ring circumference $L = n\Delta t$, where the speed of light in the waveguides is set to unity and where n is the number of time bins. This matching path length ensures that a photon which leaves the ring to scatter against the atom will return to its original time bin. Let time t_0 denote the point at which the clockwise and counterclockwise components of the photon in the ring pass their respective switches and may be injected into the scattering unit. When the switches are set to the open state, we can relate the

$\hat{a}^\dagger, \hat{b}^\dagger, \hat{c}^\dagger$ operators on the outgoing pass of the photon with:

$$\begin{bmatrix} \hat{b}_{0,R}^\dagger(t_0) \\ \hat{b}_{1,R}^\dagger(t_0) \end{bmatrix} = \begin{bmatrix} \hat{a}_\odot^\dagger(t) \\ \hat{a}_\ominus^\dagger(t) \end{bmatrix}, \quad \begin{bmatrix} \hat{c}_{0,R}^\dagger(t_0 + \frac{n\Delta t}{2}) \\ \hat{c}_{1,R}^\dagger(t_0 + \frac{n\Delta t}{2}) \end{bmatrix} = BZ_{\frac{\pi}{4}} \begin{bmatrix} \hat{b}_{0,R}^\dagger(t) \\ \hat{b}_{1,R}^\dagger(t) \end{bmatrix}. \quad (\text{C.2})$$

The $\hat{c}_{1,R}^\dagger$ component of the photon interacts at time $t_1 = t_0 + \frac{n\Delta t}{2}$ with the $|g_1\rangle$ component of the atomic state that is resonant with the photon frequency, applying the unitary transformation onto the joint photon-atom state $c\sigma_z = e^{i\pi|1\rangle\langle 1| \otimes |g_1\rangle\langle g_1|} = \exp\left(i\pi \hat{c}_{1,R}^\dagger |\emptyset\rangle\langle\emptyset| \hat{c}_{1,R} \otimes |g_1\rangle\langle g_1|\right)$. Thus, we can relate the operators before and after reflection/scattering as:

$$\left(\begin{bmatrix} \hat{c}_{0,L}^\dagger(t_1) \\ \hat{c}_{1,L}^\dagger(t_1) \end{bmatrix} \otimes \begin{bmatrix} |g_0\rangle\langle g_0| \\ |g_1\rangle\langle g_1| \end{bmatrix} \right) = \exp\left(i\pi \hat{c}_{1,R}^\dagger(t_1) \hat{c}_{1,R}(t_1) \otimes |g_1\rangle\langle g_1|\right) \left(\begin{bmatrix} \hat{c}_{0,R}^\dagger(t_1) \\ \hat{c}_{1,R}^\dagger(t_1) \end{bmatrix} \otimes \begin{bmatrix} |g_0\rangle\langle g_0| \\ |g_1\rangle\langle g_1| \end{bmatrix} \right), \quad (\text{C.3})$$

where we assume that the interaction timescale (usually set by the cavity lifetime) is negligible compared to the time bin size Δt (the long pulse limit). Eq. C.3 is derived for scattering in the single-photon subspace, but is applicable to multi-photon states as long as the photon wavefunctions do not overlap in the scattering unit.

On the return trip, after scattering against the atom, the photon passes through the beamsplitter and phase shifter in reverse order before being re-injected at time $t_2 = t_1 + \frac{n\Delta t}{2}$ into the ring at points P_1, P_2 , allowing us to relate the final set of operators:

$$\begin{bmatrix} \hat{b}_{0,L}^\dagger(t_2) \\ \hat{b}_{1,L}^\dagger(t_2) \end{bmatrix} = Z_{\frac{\pi}{4}}^\dagger B^\dagger \begin{bmatrix} \hat{c}_{0,L}^\dagger(t_1) \\ \hat{c}_{1,L}^\dagger(t_1) \end{bmatrix}, \quad \begin{bmatrix} \hat{a}_\odot^\dagger(t_2) \\ \hat{a}_\ominus^\dagger(t_2) \end{bmatrix} = \begin{bmatrix} \hat{b}_{0,L}^\dagger(t_2) \\ \hat{b}_{1,L}^\dagger(t_2) \end{bmatrix}. \quad (\text{C.4})$$

Note that the \hat{a}^\dagger and \hat{b}^\dagger operators have opposite couplings on the photon's return trip; e.g. the clockwise \hat{a}_\odot^\dagger operator couples to the top waveguide $\hat{b}_{0,R}^\dagger$ on the outgoing direction, while on the return trip, the top waveguide $\hat{b}_{0,L}^\dagger$ couples to the counterclockwise mode \hat{a}_\ominus^\dagger . One can combine the equations above to obtain that, if the atom is in the non-interacting state $|g_0\rangle$, the total transformation performed on the photon by a round trip through the scattering unit is $Z_{\frac{\pi}{4}} B B Z_{\frac{\pi}{4}}$, and the photon state in the ring is unchanged up to a factor of i : $\hat{a}_\odot^\dagger(t + n\Delta t) = i\hat{a}_\odot^\dagger(t)$ and $\hat{a}_\ominus^\dagger(t + n\Delta t) = i\hat{a}_\ominus^\dagger(t)$.

For the purpose of the gate teleportation, we initialize the atom in the $|g_0\rangle$ state and use a $R_y(\pi/2)$ rotation to change the state to $|+\rangle \equiv \frac{1}{\sqrt{2}}(|g_0\rangle + |g_1\rangle)$. The scattering interaction applies a π phase shift to the $|1\rangle \otimes |g_1\rangle$ component of the joint quantum state, implementing a $c\sigma_z$ gate. After the photon has interacted with the atom, an $R_x(-\theta)$ rotation is applied to the atom as the photon passes back through the beamsplitter and phase shifter and is injected back into the ring. Thus, the joint photon-atom state after scattering is:

$$|\Phi\rangle = ((Z_{\frac{\pi}{4}} B) \otimes R_x(-\theta)) c\sigma_z ((BZ_{\frac{\pi}{4}}) \otimes R_y(\pi/2)) (|\psi_{\text{in}}\rangle \otimes |g_0\rangle). \quad (\text{C.5})$$

Finally, a projective measurement of the atom's state in the $\{|g_0\rangle, |g_1\rangle\}$ basis is performed, obtaining a bit $m \in \{0, 1\}$. If the atomic state collapses to state $|g_m\rangle$, then we obtain a disentangled output photon-atom state:

$$|\psi_{\text{out}}\rangle \otimes |g_m\rangle = \frac{1}{\sqrt{P_m}} [\mathbb{1} \otimes |g_m\rangle\langle g_m|] |\Phi\rangle, \quad (\text{C.6})$$

where $P_m = \text{tr}[(\mathbb{1} \otimes |g_m\rangle\langle g_m|) |\Phi\rangle\langle\Phi|]$. Working in the long pulse, high cooperativity limit where pulse shape deformation from the scattering interaction is negligible¹, we obtain respective output states for $m = 0, 1$ of:

$$|\psi_{\text{out}}\rangle \otimes |g_0\rangle = \int dt \phi(t) \left[\left(i\beta \cos \frac{\theta}{2} + e^{\frac{i\pi}{4}} \alpha \sin \frac{\theta}{2} \right) \hat{a}_{\text{c}}^\dagger(t) + \left(i\alpha \cos \frac{\theta}{2} + e^{-\frac{i\pi}{4}} \beta \sin \frac{\theta}{2} \right) \hat{a}_{\text{c}}^\dagger(t) \right] |\emptyset\rangle \otimes |g_0\rangle \quad (\text{C.7})$$

$$|\psi_{\text{out}}\rangle \otimes |g_1\rangle = \int dt \phi(t) \left[\left(e^{-\frac{i\pi}{4}} \alpha \cos \frac{\theta}{2} - \beta \sin \frac{\theta}{2} \right) \hat{a}_{\text{c}}^\dagger(t) - \left(e^{\frac{i\pi}{4}} \beta \cos \frac{\theta}{2} + \alpha \sin \frac{\theta}{2} \right) \hat{a}_{\text{c}}^\dagger(t) \right] |\emptyset\rangle \otimes |g_1\rangle, \quad (\text{C.8})$$

with α, β the coefficients from the input state of Eq. C.1. Thus, the output photon state $|\psi_{\text{out}}\rangle$, depending on the outcome of the atomic measurement m , is:

$$\begin{aligned} |\psi_{\text{out}}\rangle &= \begin{cases} -iZ_{\frac{\pi}{4}} \sigma_z R_y(\theta + \pi) Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle & \text{if } m = 0 \\ Z_{\frac{\pi}{4}} \sigma_z R_y(\theta) Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle & \text{if } m = 1 \end{cases} \\ &= Z_{\frac{\pi}{4}} \sigma_z (-\sigma_y)^{m \oplus 1} R_y(\theta) Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle, \end{aligned} \quad (\text{C.9})$$

where $m \oplus 1$ denotes addition modulo 2.

C.2 Constructing arbitrary single-qubit rotations

To construct arbitrary single-qubit gates, we compose a sequence of teleported gates of the form in Eq. C.9 with a sequence of “non-entangling” scattering process which correct for local Pauli errors introduced depending on the atomic measurement outcomes. If the atom is initialized to the off-resonant $|g_0\rangle$ state, then the atom-cavity system is on resonance with the incident photon and behaves as a mirror. In this case, the π phase shifts imparted by the cavity and by the mirror in the top waveguide cancel, and the photon state is transformed as $|\psi_{\text{out}}\rangle = Z_{\frac{\pi}{4}} B B Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle = i\sigma_x |\psi_{\text{in}}\rangle$. If the atom is initialized to $|g_1\rangle$, then the atom-cavity system is off resonance with the incident

¹Here we assume that the temporal pulse length τ is much less than the time bin spacing Δt but significantly larger than the cavity decay rate, such that the pulse shape changes slowly compared to the cavity decay rate. This means that the pulse shapes for the clockwise and counterclockwise components of the photon state do not change independently. A more realistic treatment of the pulse deformation is given in the imperfection analysis presented here and in the main text.

photon. In this case, the phase shift from the mirror in the top waveguide is not matched and a relative π phase shift is imparted between the top and bottom modes, transforming the photon state as $|\psi_{\text{out}}\rangle = Z_{\frac{\pi}{4}} B \sigma_z B Z_{\frac{\pi}{4}} |\psi_{\text{in}}\rangle = -i \sigma_z Z_{\pi/2} |\psi_{\text{in}}\rangle$.

Now consider a sequence of three successive teleported rotation gates $R_y(\theta_1), R_y(\theta_2), R_y(\theta_3)$, with atomic measurement results m_1, m_2, m_3 . The goal here is to create a sequence of scattering operations which result in a gate of the form $U = R_y(\theta_3) R_x(\theta_2) R_y(\theta_1)$, which is sufficient to implement any single-qubit gate up to an overall phase decomposed as Euler angles. [110] The total operation U applied to the initial input state $|\psi_{\text{in}}\rangle$ from the three scattering operations is:

$$U = (-1)^{m_1 \oplus m_2 \oplus m_3 \oplus 1} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_2 \oplus 1} R_y(\theta_2) Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_1 \oplus 1} R_y(\theta_1) Z_{\frac{\pi}{4}}. \quad (\text{C.10})$$

We can simplify this expression using $Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m \oplus 1} = -i(-i \sigma_y \sigma_z)^{m \oplus 1} Z_{-\frac{\pi}{2}} = -i(\sigma_x)^{m \oplus 1} Z_{-\frac{\pi}{2}}$, which reduces Eq. C.10 to:

$$U = (-1)^{m_3 \oplus m_2 \oplus m_1} (-i)^{m_2 \oplus m_1} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_2) (\sigma_y \sigma_z)^{m_1 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_1) Z_{\frac{\pi}{4}}. \quad (\text{C.11})$$

Since the results of previous measurements can add extraneous Pauli gates which affect future rotations, we wish to perform adaptive operations based on the measured outcomes. After the first measurement m_1 is performed, the gate operation is:

$$U = \begin{cases} (-1)^{m_3 \oplus m_2} (-i)^{m_2} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_2) \sigma_y \sigma_z Z_{-\frac{\pi}{2}} R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0 \\ (-1)^{m_3 \oplus m_2 \oplus 1} (-i)^{m_2 \oplus 1} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_2) Z_{-\frac{\pi}{2}} R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1. \end{cases} \quad (\text{C.12})$$

Using the identities that $\sigma_z Z_{-\frac{\pi}{2}} = i Z_{+\frac{\pi}{2}}$ and that $R_i(\theta) \sigma_i = -i R_i(\theta + \pi)$ for $i = x, y, z$, we can rewrite this as:

$$U = \begin{cases} (-1)^{m_3 \oplus m_2} (-i)^{m_2} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_2 + \pi) Z_{+\frac{\pi}{2}} R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0 \\ (-1)^{m_3 \oplus m_2 \oplus 1} (-i)^{m_2 \oplus 1} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} Z_{-\frac{\pi}{2}} R_y(\theta_2) Z_{-\frac{\pi}{2}} R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1. \end{cases} \quad (\text{C.13})$$

Substituting $Z_{-\frac{\pi}{2}} R_y(\theta) Z_{+\frac{\pi}{2}} = R_x(\theta)$ and $Z_{-\frac{\pi}{2}} R_y(\theta) Z_{-\frac{\pi}{2}} = i \sigma_z R_x(-\theta)$, we rearrange the equation to turn the second rotation gate into a $R_x(\pm\theta)$ gate, where the sign depends on the outcome of m_1 ,

which is already known:

$$\begin{aligned}
U &= \begin{cases} (-1)^{m_3 \oplus m_2} (-i)^{m_2} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} R_x(\theta_2 + \pi) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0 \\ i (-1)^{m_3 \oplus m_2 \oplus 1} (-i)^{m_2 \oplus 1} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (\sigma_y \sigma_z)^{m_2 \oplus 1} \sigma_z R_x(-\theta_2) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1 \end{cases} \\
&= (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (i \sigma_y \sigma_z)^{m_2 \oplus 1} \times \begin{cases} R_x(\theta_2 + \pi) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0 \\ \sigma_z R_x(-\theta_2) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1. \end{cases}
\end{aligned} \tag{C.14}$$

Importantly, the decision for which adaptive changes to apply to the θ_2 operation (adding π or inverting the angle) can be made knowing only the outcome of the previous measurement m_1 . Let $\theta_2(m_1) = \theta_2 + \pi$ if $m_1 = 0$ and $-\theta_2$ if $m_1 = 1$ denote the adaptive angle to implement the desired rotation $R_x(\theta_2)$. Then we can rewrite Eq. C.14 as:

$$U = (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) (i \sigma_y \sigma_z)^{m_2 \oplus 1} \sigma_z^{m_1} R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}}. \tag{C.15}$$

We repeat this process of performing a measurement and commuting the error terms to the front of the equation for measurement m_2 . After performing the second measurement, we use $R_y(\theta) \sigma_z = \sigma_z R_y(-\theta)$ and the above identities to obtain:

$$\begin{aligned}
U &= (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3) \times \begin{cases} i \sigma_y \sigma_z R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 0 \\ R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 1 \\ i \sigma_y \sigma_z \sigma_z R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 0 \\ \sigma_z R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 1 \end{cases} \\
&= (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} \times \begin{cases} -R_y(\theta_3 + \pi) \sigma_z R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 0 \\ R_y(\theta_3) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 1 \\ -R_y(\theta_3 + \pi) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 0 \\ R_y(\theta_3) \sigma_z R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 1 \end{cases} \\
&= (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} \times \begin{cases} -\sigma_z R_y(-\theta_3 - \pi) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 0 \\ R_y(\theta_3) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 0, m_2 = 1 \\ -R_y(\theta_3 + \pi) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 0 \\ \sigma_z R_y(-\theta_3) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} & \text{if } m_1 = 1, m_2 = 1. \end{cases}
\end{aligned} \tag{C.16}$$

As before, the modifications to θ_3 can be performed with only knowledge of m_1 and m_2 . Let $\theta_3(m_2, m_1)$ be defined as in Eq. C.16, such that $\theta_3(m_2, m_1) = (-1)^{m_2 \oplus m_2 \oplus 1} (\theta_3 + \pi(1 - m_2))$. We

perform the final measurement m_3 using this adaptive θ_3 . We obtain an equation of the desired form with a possible Pauli error term $\varepsilon(m_1, m_2, m_3)$ at the front:

$$\begin{aligned}
U &= (-1)^{m_3} Z_{\frac{\pi}{4}} \sigma_z (\sigma_y)^{m_3 \oplus 1} (-1)^{m_2} \sigma_z^{m_2 \oplus m_1 \oplus 1} R_y(\theta_3(m_2, m_1)) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} \\
&= (-1)^{m_3 \oplus m_2 \oplus 1} Z_{\frac{\pi}{4}} \sigma_z^{m_2 \oplus m_1} (\sigma_y)^{m_3 \oplus 1} R_y(\theta_3(m_2, m_1)) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} \\
&= (-1)^{m_3 \oplus m_2 \oplus 1} \sigma_z^{m_2 \oplus m_1} (-\sigma_y)^{m_3 \oplus 1} Z_{\frac{\pi}{4}} R_y(\theta_3(m_2, m_1)) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}} \\
&\equiv \varepsilon(m_1, m_2, m_3) Z_{\frac{\pi}{4}} R_y(\theta_3(m_2, m_1)) R_x(\theta_2(m_1)) R_y(\theta_1) Z_{\frac{\pi}{4}},
\end{aligned} \tag{C.17}$$

where the error term $\varepsilon(m_1, m_2, m_3)$ is:

$$\begin{aligned}
\varepsilon(0, 0, 0) &= -\sigma_y \\
\varepsilon(0, 0, 1) &= -\mathbb{1} \\
\varepsilon(0, 1, 0) &= -i\sigma_x \\
\varepsilon(0, 1, 1) &= \sigma_z \\
\varepsilon(1, 0, 0) &= -i\sigma_x \\
\varepsilon(1, 0, 1) &= \sigma_z \\
\varepsilon(1, 1, 0) &= -\sigma_y \\
\varepsilon(1, 1, 1) &= -\mathbb{1}.
\end{aligned} \tag{C.18}$$

We can remove any of these errors up to a global phase by using a sequence of non-interacting passes, where the atom is initialized to $|g_0\rangle$ or $|g_1\rangle$ rather than $|+\rangle$. To remove $-i\sigma_x$, we use a $|g_0\rangle$ initialization to apply $Z_{\frac{\pi}{4}} B B Z_{\frac{\pi}{4}} = i\sigma_x$. To remove σ_z , we use two $|g_1\rangle$ -initialized scatterings to apply $Z_{\frac{\pi}{4}} B \sigma_z B Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} B \sigma_z B Z_{\frac{\pi}{4}} = -i\sigma_z$. To remove σ_y , we apply two $|g_1\rangle$ -initialized scatterings and one $|g_0\rangle$ -initialized scatterings to apply $Z_{\frac{\pi}{4}} B \sigma_z B Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} B \sigma_z B Z_{\frac{\pi}{4}} Z_{\frac{\pi}{4}} B B Z_{\frac{\pi}{4}} = -i\sigma_y$. Thus, one can apply arbitrary single-qubit operations parameterized via YXY Euler angles using this gate construction method.

C.3 Photonic qubit readout

To measure the state of a photonic qubit, we construct a SWAP gate from a sequence of three scattering operations. We may initialize the atom to any state, and we then perform the sequence of scattering interactions shown in Figure C.2.

Let $|\psi\rangle = \left(\alpha \hat{b}_{0L}^\dagger + \beta \hat{b}_{1L}^\dagger \right) |\emptyset\rangle$ be the state of the photon at points P_3, P_4 in the device. By scattering the photon against the atom three times and applying the rotation $R_y(\pi/2)R_x(\pi)$ to the atomic states in between scattering, one can swap the states of the photon and atom, such that the final atomic state is $\alpha |g_0\rangle + \beta |g_1\rangle$. It is straightforward to verify that this sequence of operations

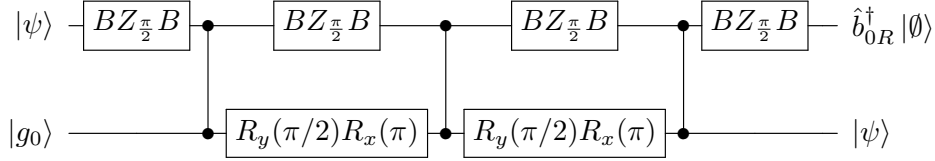


Figure C.2: Construction of a SWAP gate from three scattering interactions. The top rail denotes the photonic qubit and the bottom rail denotes the atom. The $BZ_{\frac{\pi}{2}}B$ operations correspond to a return trip of the photon from the scattering site to the ring and back, passing through the beamsplitter and phase shifter twice.

implements the SWAP gate up to a phase of -1 :

$$(BZ_{\frac{\pi}{2}}B \otimes \mathbf{1}) c\sigma_z (BZ_{\frac{\pi}{2}}B \otimes Y_{\frac{\pi}{2}}X_{\pi}) c\sigma_z (BZ_{\frac{\pi}{2}}B \otimes Y_{\frac{\pi}{2}}X_{\pi}) c\sigma_z (BZ_{\frac{\pi}{2}}B \otimes \mathbf{1}) = -1 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{C.19})$$

Once the states of the photonic and atomic qubits are swapped, the atomic state can be measured with near 100% efficiency using the quantum jump technique [162, 59] while the photonic qubit is discarded by allowing it to gradually dissipate through leakage to the environment. This SWAP-and-measure protocol can be repeated for the rest of the photonic qubits to read out the entire photonic quantum state.

C.4 Implementing a two-photon $c\sigma_z$ gate

In addition to implementing single-qubit gates, constructing a two-photon entangling gate is necessary for universal computation. A controlled phase-flip gate $c\sigma_z$ between two photonic qubits can be constructed through a sequence of three scattering interactions in a somewhat similar manner as in Ref. [59]. However, the fixed beamsplitter and phase shifter, which are required for implementation of single-qubit gates in our scheme, only allow us to apply operations of the form $((Z_{\frac{\pi}{4}}B) \otimes \mathbf{1}) c\sigma_z ((BZ_{\frac{\pi}{4}}) \otimes \mathbf{1})$ to the $|\text{photon}\rangle \otimes |\text{atom}\rangle$ system with each scattering interaction. This prevents us from performing the exact protocol described in Ref. [59], which requires photons to undergo three successive $c\sigma_z$ operations without any gates between them.

Here we describe two possible implementations of this $c\sigma_z$ gate between two photons A and B in states $|\psi_A\rangle$ and $|\psi_B\rangle$ which work with the design of our proposed device. The first solution is to use a SWAP gate as described in Section C.3 to swap the states of photon A and the atom, then perform a scattering of photon B against the atom, then to swap the atomic state back to photon A .

Although the construction of $c\sigma_z$ through SWAP gates allows for direct interaction of $|\psi_A\rangle$ with $|\psi_B\rangle$, it involves a total of $3 + 1 + 3 = 7$ scattering interactions, which is significantly less compact than the three scatterings used in the construction from Ref. [59].

We can implement a more compact construction of $c\sigma_z$ which also only requires three scatterings by using a measurement based scheme shown in Figure C.3. This second possible construction implements a $c\sigma_z$ gate between photons A and B which is sandwiched between single-qubit gates. These extra gates can be implicitly removed by programming the single-qubit gate U_{before} which immediately precedes this operation to instead implement $(Z_{\frac{\pi}{4}}B)^{-1}U_{\text{before}}$ and the gate U_{after} following $c\sigma_z$ to implement $U_{\text{after}}(BZ_{\frac{\pi}{4}})^{-1}$.

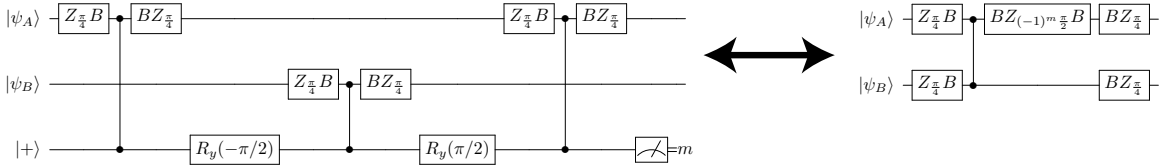


Figure C.3: Construction of a $c\sigma_z$ gate with three scattering interactions using a measurement-based approach. After measurement, the left and right circuits are equivalent. The single-qubit gates on either side of $c\sigma_z$ can be removed by absorbing them into the preceding/subsequent single-qubit gates as described above.

C.5 Circuit compilation

An arbitrary n -qubit quantum operator $U \in U(2^n)$, can be compiled into a sequence of physical instructions on the proposed device using a three-step process shown in Figure 4 of the main text, and shown in greater detail in Figure C.4 of this document. The first step is to decompose U into a sequence of single-qubit gates and $c\sigma_z$ operations, a process described in our previous work [21]. The second step is to decompose each single-qubit gate via Euler angles as three R_y rotations which may be teleported onto the photonic qubits by a sequence of scatter-rotate-measure operations. The third step is to use a high-speed classical control system to modify the adaptive rotations which are applied to the atomic qubit based on the measurement outcomes during operation. Pauli errors which are accumulated during the course of the circuit operation can either be removed explicitly by scattering against $|g_0\rangle$ or $|g_1\rangle$, as described at the end of Section C.2, or can be removed implicitly (resulting in a more compact circuit) by programming the inverse of the error term into subsequent single-qubit operators. An example program for implementing a three-qubit quantum Fourier transform is shown in Program C.1 at the end of this Supplementary Information document.

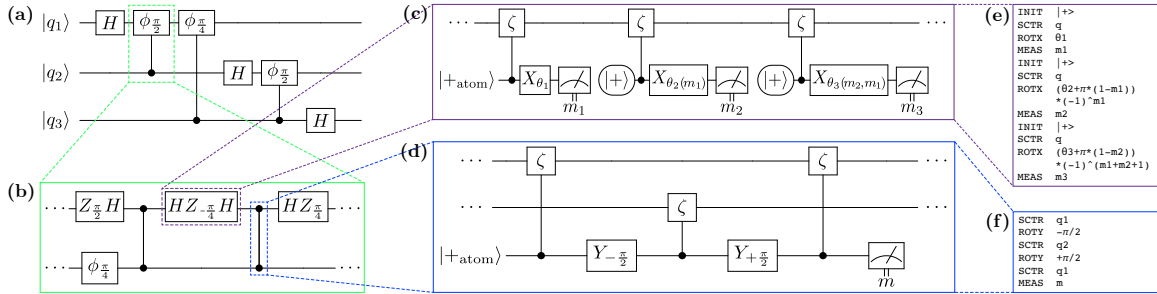


Figure C.4: Graphical depiction of the circuit compilation process. (a) The target quantum circuit we wish to implement in the device, in this case a three-qubit quantum Fourier transform. (b) The first step of the compilation process is to decompose complex circuit elements into single-qubit and $c\sigma_z$ gates. The subcircuit depicted here implements the first controlled- $\phi_{\pi/2}$ gate between photonic qubits q_1 and q_2 . (c, d) The second step is to decompose each single-qubit gate (c) via Euler angles into a sequence of rotations which can be teleported from the atom to the photonic qubits, and to decompose each $c\sigma_z$ gate (d) using the scattering sequence shown in Figure C.3. (e, f) Programmatic representation of the instructions sent to the device to implement subroutines (c,d), respectively. The full code for implementing the target quantum circuit depicted in (a) is shown in Program C.1.

C.6 Compiled example: quantum Fourier transform

```

1 # Instruction set
2 # -----
3 # OPEN t ... open the switches at time t
4 # CLOS t ... close the switches at time t
5 # ROTX  $\theta$  ... laser pulse rotates atom state, Rx( $\theta$ )
6 # ROTY  $\theta$  ... laser pulse rotates atom state, Ry( $\theta$ )
7 # MEAS m ... measure atom state and store bit as m
8 # INIT  $\Psi$  ... initialize atom to  $|\Psi\rangle = |g_0\rangle, |g_1\rangle, |+\rangle$ 
9
10
11 # Scatter photon q and return it to ring
12 define SCTR q:
13     OPEN t_q -  $\Delta t/2$       # t_q: time bin for |q>
14     CLOS t_q +  $\Delta t/2$     #  $\Delta t$ : temporal bin size
15     OPEN N *  $\Delta t$  + t_q -  $\Delta t/2$  # N: number of time bins
16     CLOS N *  $\Delta t$  + t_q +  $\Delta t/2$  # N *  $\Delta t$ : time around ring
17
18

```

```

19 # Explicitly correct Pauli errors after a gate
20 define CORR q m1 m2 m3:
21     if m3 == 0:
22         INIT |g1>
23         SCTR q
24         SCTR q
25         INIT |g0>
26         SCTR q
27     if m1 != m2:
28         INIT |g1>
29         SCTR q
30         SCTR q
31
32
33 # Single-qubit gate via Euler angles
34 define GATE q  $\theta_1$   $\theta_2$   $\theta_3$ :
35     INIT |>
36     SCTR q
37     ROTX  $\theta_1$ 
38     MEAS m1
39     INIT |>
40     SCTR q
41     ROTX  $(\theta_2 + \pi(1-m_1)) * (-1)^{m_1}$  # adaptive  $\theta_2$ 
42     MEAS m2
43     INIT |>
44     SCTR q
45     ROTX  $(\theta_3 + \pi(1-m_2)) * (-1)^{(m_1+m_2+1)}$ 
46     MEAS m3
47     CORR q m1 m2 m3 # remove Pauli  $\epsilon(m_1, m_2, m_3)$ 
48
49
50 # Swap photon q with atom state
51 define LOAD q:
52     SCTR q
53     ROTX  $\pi$ 
54     ROTY  $\pi/2$ 
55     SCTR q

```

```

56     ROTX   $\pi$ 
57     ROTY   $\pi/2$ 
58     SCTR  q
59     ROTX   $\pi/2$ 
60     ROTY   $\pi/4$ 
61
62
63 # Controlled- $\sigma_z$  between photons q1, q2
64 define CTRZ q1 q2:
65     GATE  q1  0   $3\pi/4$    $-\pi/2$ 
66     GATE  q2  0   $3\pi/4$    $-\pi/2$ 
67     SCTR  q1
68     ROTY   $-\pi/2$ 
69     SCTR  q2
70     ROTY   $+\pi/2$ 
71     SCTR  q1
72     MEAS  m
73     GATE  q1   $m*\pi$    $\pi/2$    $(-1)^{m*3\pi/2}$ 
74     GATE  q2   $\pi/2$    $3\pi/4$   0
75
76
77 # Run a 3-qubit QFT and measure the qubits
78 GATE  q1  5.668  2.094  0.615  # H
79 GATE  q1  3.757  2.094  5.668  #  $c\varphi(\pi/2)$ 
80 CTRZ  q2  q1
81 GATE  q1  2.101  1.718  4.182
82 CTRZ  q2  q1
83 GATE  q1  0.000  2.356  1.571
84 GATE  q3  1.571  0.785  4.712  #  $c\varphi(\pi/4)$ 
85 GATE  q1  4.712  2.356  0.000
86 CTRZ  q3  q1
87 GATE  q1  1.845  1.609  4.438
88 CTRZ  q3  q1
89 GATE  q1  5.918  2.283  1.041
90 GATE  q2  5.668  2.094  0.615  # H
91 GATE  q2  3.757  2.094  5.668  #  $c\varphi(\pi/2)$ 
92 CTRZ  q3  q2

```



```

93 GATE q2 2.101 1.718 4.182
94 CTRZ q3 q2
95 GATE q2 0.000 2.356 1.571
96 GATE q2 5.668 2.094 0.615 # H
97
98 # State readout
99 LOAD q1
100 MEAS b1
101 LOAD q2
102 MEAS b2
103 LOAD q3
104 MEAS b3

```

Program C.1: Assembly-like pseudocode for implementing the three-qubit quantum Fourier transform shown in Figure C.4a. For simplicity and readability, this code explicitly corrects for Pauli errors using the `CORR` subroutine and removes extraneous BZ terms in the $c\sigma_z$ gate construction using four additional `GATE` calls within `CTRZ`. The numerical values for the `GATE` angles in lines 78-96 were computed using a modified version of `OneQubitEulerDecomposer` in `Qiskit` [241].

C.7 Imperfection analysis

Here we describe the details of the imperfection analysis that we used for estimating the achievable circuit depth, shown in Figure 4 of the main text. The main sources of error for our protocol are the same as for the Duan-Kimble protocol [59], but with the added loss from the switches and propagation loss through the storage ring. We group these errors into three main classes:

- Pulse shape infidelity: mismatch between the cavity output pulses for the atom being in the $|g_0\rangle$ and $|g_1\rangle$ states. This loss can be minimized by choosing the photon's temporal width (τ) to be much larger than the cavity photon lifetime $1/\kappa$: $\kappa\tau \gg 1$.
- Spontaneous emission loss of the excited state of the atom, where the atom in the $|e\rangle$ state emits not into the desired cavity mode but into other modes or into free space. In our scheme, this causes photon leakage error when the atom is in the $|g_1\rangle$ state, since the photon causes the $|g_1\rangle$ state to temporarily transition to $|e\rangle$.
- Photon loss due to optical elements. This includes optical attenuation while propagating through the storage ring, insertion loss of the optical switches, and spurious loss from the cavity mirrors or the cavity medium.

We assume that the cavity mode at ω_c is resonant with the atom $|g_1\rangle \leftrightarrow |e\rangle$ transition frequency Ω_1 , since the detuning can be actively tuned to be zero, both in free-space by tuning the cavity length,

as well as in solid-state nanophotonic systems through temperature or strain. We also assume that rotations of the atomic state by the cavity laser and measurement of the state via the quantum jump technique can be done with fidelity $\mathcal{F} \approx 1$, since both processes have been demonstrated experimentally with very high fidelities [74] greatly exceeding that of the effects listed above.

To quantify the effects of these sources of error, we assume the input waveguide contains a single photon Fock state of the form $\int dt \phi_{\text{in}}(t) \hat{a}_{\text{in}}^\dagger(t) |\varnothing\rangle$, where $\phi_{\text{in}}(t)$ is the pulse shape, $|\varnothing\rangle$ represents the vacuum state of the waveguide modes, and $\hat{a}_{\text{in}}^\dagger(t)$ is a bosonic operator obeying the standard commutation relation $[\hat{a}_{\text{in}}(t), \hat{a}_{\text{in}}^\dagger(t')] = \delta(t - t')$ which creates a photon propagating toward the cavity in the waveguide at time t . For the cavity output, we assume a similar form, $\int dt \phi_{\text{out}}(t) \hat{a}_{\text{out}}^\dagger(t) |\varnothing\rangle$ [59, 224], where $\hat{a}_{\text{out}}^\dagger(t)$ is similarly defined and creates a photon propagating away from the cavity at time t . For our analysis, we choose a Gaussian pulse envelope centered at $t_0 = \Delta t/2$ for the input: $\phi_{\text{in}}(t) \propto \exp[-(t - t_0)^2/\tau^2]$, as studied in Ref. [59].

To solve for the output single-photon pulse, we use the analytical technique described by Shen and Fan [223, 224], which exactly solves the single-photon transport problem of a coupled atom-cavity-waveguide system, taking into account all relevant energy scales. The effective Hamiltonian of the overall system is given by [224]:

$$\begin{aligned} \mathcal{H}_{\text{eff}}/\hbar = & (\omega_c - i\kappa_i/2) \hat{a}^\dagger \hat{a} + (\Omega_e - i\gamma_s/2) |e\rangle\langle e| + \Omega_1 |g_1\rangle\langle g_1| + \Omega_0 |g_0\rangle\langle g_0| + (g \hat{a}^\dagger |g_1\rangle\langle e| + \text{H.c.}) \\ & + \int dx \delta(x) \left[\sqrt{\kappa v_g/2} \hat{a}^\dagger \hat{a}_{\text{in}}(x) + \sqrt{\kappa v_g/2} \hat{a}^\dagger \hat{a}_{\text{out}}(x) + \text{H.c.} \right] \\ & + \int dx \hat{a}_{\text{in}}^\dagger(x) (\omega_c - i v_g \partial_x) \hat{a}_{\text{in}}(x) + \int dx \hat{a}_{\text{out}}^\dagger(x) (\omega_c + i v_g \partial_x) \hat{a}_{\text{out}}(x), \end{aligned} \quad (\text{C.20})$$

where \hat{a}^\dagger is a bosonic operator that creates a photon in the cavity mode at ω_c obeying $[\hat{a}, \hat{a}^\dagger] = 1$, κ_i is the intrinsic dissipation rate of the cavity mode, $\Omega_{0,1,e}$ are the energies of the respective atomic states, g is the single-photon atom-cavity coupling rate (equal to half the vacuum Rabi splitting), v_g is the group velocity of the waveguide in the vicinity of the cavity resonant frequency ω_c , and γ_s is the spontaneous emission rate of the atomic $|e\rangle$ state². In the following analysis, we set $\kappa_i = 0$.

The spectrum of the output pulse, $\tilde{\phi}_{\text{out}}(\omega) = \mathcal{F}\{\phi_{\text{out}}(t)\}$ is related to the input pulse spectrum $\tilde{\phi}_{\text{in}}(\omega) = \mathcal{F}\{\phi_{\text{in}}(t)\}$ by the spectral response of the cavity-atom system $R(\omega, g, \kappa, \gamma_s, |A\rangle)$. Here, $\mathcal{F}\{\cdot\}$ denotes the Fourier transform, and ω denotes the input photon detuning from the cavity/atom resonance, $\omega = \omega_{\text{in}} - (\Omega_e - \Omega_1) = \omega_{\text{in}} - \omega_c$. The spectral response depends on the initial state of the atom $|A\rangle \in \{|g_0\rangle, |g_1\rangle\}$. This treatment captures the full quantum mechanical response of

²One should note that, while the use of the non-Hermitian $-i\gamma_s/2|e\rangle\langle e|$ term is known to produce correct scattering matrices for single-photon interactions, the direct substitution of $\Omega_e \rightarrow \Omega_e - i\gamma_s/2$ to describe spontaneous emission loss will yield incorrect results for temporally-overlapping multi-photon scattering interactions. [207] The more correct treatment here is to add additional couplings between the system Hamiltonian and a bath of modes describing the environment, but this is not necessary for our analysis, which is limited to single-photon interactions.

the system to a single-photon Fock state input for an arbitrary initialization of the atom, without making the semiclassical assumption of a weak coherent state for the input.

Pulse shape infidelity and delay correction — For an atom initialized as $|A\rangle = |g_0\rangle$, the response is identical to an empty cavity since the $|g_0\rangle \leftrightarrow |e\rangle$ transition frequency is far-detuned from the cavity mode frequency Ω_c [60]. In this case, the output pulse is slightly delayed from the input pulse by a time δt_0 , as it couples into the empty cavity mode before coupling out, leading to a fidelity below unity, as shown in Figure 4 of the main text. For an initialization $|A\rangle = |g_1\rangle$, the photon is directly reflected from the front mirror of the cavity, since the dressed cavity modes are well-separated from the input photon frequency by the vacuum Rabi splitting for strong coupling $g \gg \kappa, \gamma_s$, and the delay $\delta t_1 \approx 0$ is minimal. Here the pulse shape fidelity is defined as [60, 61]:

$$\mathcal{F}_{\text{shape}} \equiv \left| \int dt \bar{\phi}_{\text{in}}^*(t) \bar{\phi}_{\text{out}}(t) \right|, \quad (\text{C.21})$$

where $\bar{\phi}_{\text{in}}$ and $\bar{\phi}_{\text{out}}$ are the renormalized input and output pulses. The pulse shape infidelity is defined as $1 - \mathcal{F}_{\text{shape}}$. Importantly, this quantity only describes the infidelity due to shape mismatch of the input and output pulses, not amplitude mismatch; the infidelity due to spontaneous emission loss is computed separately. The average infidelity for an initialization in the $|+\rangle = (|g_0\rangle + |g_1\rangle)/\sqrt{2}$ state is calculated as the mean of the infidelities for the $|g_0\rangle$ and $|g_1\rangle$ states [59]. In our calculations, using a long pulse width $\tau = 100/\kappa$ and total interaction timescale $T = 500/\kappa$ and assuming no intrinsic losses in the cavity ($\kappa_i = 0$) aside from spontaneous emission results in a low infidelity below 10^{-3} per photon-cavity scattering event.

In Figure 4(b) of the main text, we plot the shape infidelity of various states as a function of the single-atom cavity cooperativity $C \equiv 4g^2/\kappa\gamma_s$, where γ_s measures the spontaneous emission rate and is fixed at $\gamma_s = \kappa/5$. The pulse shape infidelity of an interaction with the $|g_1\rangle$ state decreases to negligible values as C increases, while the infidelity of $|g_0\rangle$ reaches an asymptote at 8×10^{-4} due to the delay of the output pulse by a time δt_0 which is independent of C ; the infidelity of the $|+\rangle$ interaction asymptotes at 4×10^{-4} . Since the atom will usually be initialized to the $|+\rangle$ state during operation of the device, it is useful to minimize the infidelity of interacting with this state. This can be done by delaying the reference pulse by a time difference $t_{\text{delay}} = (\delta t_0 + \delta t_1)/2 \approx \delta t_0/2$ by adding an additional path length $ct_{\text{delay}}/2$ to the top waveguide in Figure 1 of the main text. This distributes the infidelity due to the output pulse delay equally between the $|g_0\rangle$ and $|g_1\rangle$ states, such that the output pulse of a $|g_1\rangle$ interaction is shifted forward by $\delta t_0/2$ and the output of a $|g_0\rangle$ interaction is delayed by $\delta t_0/2$. This results in an infidelity of approximately 2×10^{-4} which is independent of both cavity cooperativity (at $C \gg 1$) and atomic state initialization.

Spontaneous emission loss — Atomic spontaneous emission noise from the excited $|e\rangle$ state at a rate γ_s results in a partial loss of the photon, resulting in an output pulse with total photon number

$\int dt |\phi_{\text{out}}(t)|^2 < 1$. We calculate the probability P_s of spontaneous emission loss as:

$$P_s = 1 - \frac{\int dt |\phi_{\text{out}}(t)|^2}{\int dt |\phi_{\text{in}}(t)|^2}. \quad (\text{C.22})$$

Spontaneous emission noise only applies to the $|1\rangle \otimes |g_1\rangle$ component of the photon \otimes atom state. The atom will usually be initialized to the $|+\rangle$ state, and averaging over possible input photon states, we obtain an average leakage probability of $\bar{P}_s = P_s/4$, as shown in Figure 4(b), which is well-approximated by $\bar{P}_s = [4(1 + 2C)]^{-1}$.

Spurious photon loss and maximum circuit depth — Finally, we account for loss due to propagation through the optical paths and switches as an average loss per cycle L . To estimate the maximum circuit depth D attainable with an overall fidelity $\mathcal{F} > \mathcal{F}_{\text{target}}$, we compute a “bulk fidelity” accounting for shape mismatch and loss due to average spontaneous emission and propagation through the storage ring. For simplicity, we assume the circuit operates on only a single photonic qubit and that the photon is scattered off the atom with every pass through the storage ring. The achievable circuit depth operating with success probability $P_{\text{success}} = \mathcal{F}_{\text{target}}$ is thus the maximum D satisfying:

$$[\mathcal{F}_{\text{shape}} \times (1 - \bar{P}_s) \times (1 - L)]^D \geq \mathcal{F}_{\text{target}}, \quad (\text{C.23})$$

which is plotted as a function of cavity cooperativity and propagation loss in Figure 5.4(c).

Appendix D

Programmable photonic quantum emulator

In this Appendix, we give more detailed presentations of the results described the main text. In Sections D.1 and D.2 we derive a correspondence between the device physics and the terms in the target Hamiltonian, and in Section D.3 we provide additional details on the simulation methods used to tractably simulate interacting bosons on large lattices.

D.1 Deriving hopping coefficients between sites

The Hamiltonians of interest take the form:

$$\hat{H} = \sum_{\langle m,n \rangle} (\kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.}) + \mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m, \quad (\text{D.1})$$

where \hat{a}_m^\dagger creates a boson at node m , $\kappa_{m,n}$ denotes the hopping coefficients between sites m and n , with the Hermitian requirement that $\kappa_{m,n} = \kappa_{n,m}$, μ is the single-body energy per site, U is the Hubbard interaction strength, and $\alpha_{m,n}$ is a phase shift accumulated by moving from site m to n . The two-particle summation in the first term is taken over all connected sites $\langle m,n \rangle$, where the connectivity of the simulated system is determined with suitable choice of $\{\kappa_{m,n}\}$.

A system evolving under the Hamiltonian \hat{H} in Eq. D.1 will have a propagator (with $\hbar = 1$) of the form:

$$e^{-i\hat{H}t} = \exp \left[-it \left(- \sum_{\langle m,n \rangle} \kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \right) \right] \quad (\text{D.2})$$

If we add Hermitian constraints to the Hamiltonian, we have that $\kappa_{mn} = \kappa_{nm}$ and $\alpha_{mn} = -\alpha_{nm}$, so we can write:

$$e^{-i\hat{H}t} = \exp \left[-it \left(- \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) + \mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \right) \right], \quad (\text{D.3})$$

where the product over $\langle m,n \rangle$ now implicitly avoids double-counting, as we have explicitly included the Hermitian conjugate in the first term. We can series expand this using the Suzuki-Trotter expansion as $e^{t(X+Y)} = e^{tX} e^{tY} e^{-\frac{t^2}{2}[X,Y]} e^{\frac{t^3}{6}(2[Y,[X,Y]]+[X,[X,Y]])} \dots$ to separate the summation into a product of matrix exponentials:

$$\begin{aligned} e^{-i\hat{H}t} &\approx \exp \left(it \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \right) \times e^{-it(\mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \hat{a}_m^\dagger)} \\ &\quad \times \exp \left(-\frac{t^2}{2} \left[\sum_{\langle m,n \rangle} i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m), \quad \sum_m \mu \hat{a}_m^\dagger \hat{a}_m + \sum_m U \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \right] \right) \\ &\equiv \exp \left(it \sum_{\langle m,n \rangle} i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \right) \times e^{-it(\mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \hat{a}_m^\dagger)} \times \varepsilon_{\kappa,\mu} \times \varepsilon_{\kappa,U} \end{aligned} \quad (\text{D.4})$$

We evaluate the commutator error terms ε_μ and ε_U in Section D.1.1 and find that $\varepsilon_\mu = \mathbb{1}$ since the κ_{mn} and μ terms commute, and $\varepsilon_U = \exp \left(-2t^2 U \sum_{\langle m,n \rangle} \kappa_{mn} \cos \alpha_{mn} ((\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - \hat{a}_n^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) \right)$.

We now perform a second series expansion on the first term to separate the exponential of sum into a product of exponentials:

$$\begin{aligned} e^{-i\hat{H}t} &\approx \left(\prod_{\langle m,n \rangle} \exp i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \right)^t \times e^{-it(\sum_m \mu \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \hat{a}_m^\dagger)} \times \varepsilon_U \\ &\quad \times \exp \left(-\frac{t^2}{2} \sum_{\langle j,k \neq k' \rangle} [\kappa_{jk} (e^{i\alpha_{jk}} \hat{a}_j^\dagger \hat{a}_k + e^{-i\alpha_{jk}} \hat{a}_k^\dagger \hat{a}_j), \quad \kappa_{jk'} (e^{i\alpha_{jk'}} \hat{a}_j^\dagger \hat{a}_{k'} + e^{-i\alpha_{jk'}} \hat{a}_{k'}^\dagger \hat{a}_j)] \right) \\ &\equiv \left(\prod_{\langle m,n \rangle} \exp i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \right)^t \times e^{-it(\sum_m \mu \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \hat{a}_m^\dagger)} \times \varepsilon_U \times \varepsilon_\kappa. \end{aligned} \quad (\text{D.5})$$

We can expand the exponentials of the error terms as $e^A \approx \mathbb{1} + A$ to obtain error scaling on the final

result:

$$e^{-i\hat{H}t} = \left(\prod_{\langle m,n \rangle} \exp(i\kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.}) \right)^t \times e^{-it(\sum_m \mu \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)} \\ \times \left(\mathbb{1} + \mathcal{O}(\kappa U \cos \alpha) \sum_{\langle m,n \rangle} (\hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_n - \hat{a}_n \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m) + \mathcal{O}(\kappa^2) \sum_{\langle j,k \neq k' \rangle} [\hat{a}_j^\dagger \hat{a}_k, \hat{a}_j \hat{a}_{k'}^\dagger] \right)^{-t^2/2} \quad (\text{D.6})$$

where κ, α is shorthand for typical (or uniform) values of κ_{mn}, α_{mn} . If we have small coefficients $\kappa, U \ll 1$, then $\mathcal{O}(\kappa^2 + \kappa U)$ is negligible, and we ignore the commutator error term going forward. If κ, U is not small, we can reduce the emulated values of κ_{mn}, μ , and U by some constant factor C and run the emulation for a commensurately longer wall clock time Ct . Thus, to within $\mathcal{O}(\kappa^2 + \kappa U)$, we can write the propagator as:

$$e^{-i\hat{H}t} = \left[\prod_{\langle m,n \rangle} \exp(i\kappa_{mn} e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + \text{H.c.}) \right]^t \times e^{-it(\mu \sum_m \hat{a}_m^\dagger \hat{a}_m + U \sum_m \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)}. \quad (\text{D.7})$$

Now consider the tunable MZI connecting the storage ring to the register ring as shown in Figure 1 of the main text. Define bosonic operators \hat{a}_1^\dagger and \hat{a}_2^\dagger which create right-moving photons at the input waveguides to the MZI and operators \hat{b}_1^\dagger and \hat{b}_2^\dagger which create right-moving output photons. The transfer matrix of the MZI if the internal phase shifter is set to an angle θ and the external phase shifters are set to $\pm\phi$ is $T = R_z(-\phi)HR_z(\theta)HR_z(+\phi)$. Thus, we can relate the output modes to the input modes as:

$$\begin{bmatrix} \hat{b}_1^\dagger \\ \hat{b}_2^\dagger \end{bmatrix} = \begin{pmatrix} \cos \frac{\theta}{2} & ie^{i\phi} \sin \frac{\theta}{2} \\ ie^{-i\phi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \begin{bmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{bmatrix} = \exp \left[i\frac{\theta}{2} \left(e^{i\phi} \hat{a}_1^\dagger \hat{a}_2 + e^{-i\phi} \hat{a}_2^\dagger \hat{a}_1 \right) \right] \begin{bmatrix} \hat{a}_1^\dagger \\ \hat{a}_2^\dagger \end{bmatrix}. \quad (\text{D.8})$$

If we define $\kappa \equiv \frac{\theta}{2}$ and $\alpha \equiv \phi$, then we obtain the desired transfer matrix from which we can construct the first part of the propagator in Eq. D.7:

$$T_{12} = \exp \left[i\kappa \left(e^{i\alpha} \hat{a}_1^\dagger \hat{a}_2 + e^{-i\alpha} \hat{a}_2^\dagger \hat{a}_1 \right) \right]. \quad (\text{D.9})$$

If we apply a sequence of passes of the photon pulses through the MZI, then by appropriately choosing values of θ, ϕ to match κ_{mn}, α_{mn} , we obtain a total transfer matrix of:

$$T_{\langle m,n \rangle} = \prod_{\langle m,n \rangle} \exp \left(i\kappa_{mn} \left(e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m \right) \right), \quad (\text{D.10})$$

so after t repetitions of this sequence of passes, the total accumulated operation is:

$$T_{\langle m,n \rangle} = \left[\prod_{\langle m,n \rangle} \exp(i\kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)) \right]^t, \quad (\text{D.11})$$

which is exactly the desired form from the propagator in Eq. D.7.

D.1.1 Evaluating the commutator error terms

We can split the commutator in the last line of Eq. D.4 into two parts: $\exp(-it^2/2(\varepsilon_{\kappa,\mu} + \varepsilon_{\kappa,U}))$.

Starting with the κ_{mn}, μ commutator, we have:

$$\begin{aligned} \varepsilon_{\kappa,\mu} &= \left[\sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m), \quad \sum_m \mu \hat{a}_m^\dagger \hat{a}_m \right] \\ &= \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \sum_o \mu \hat{a}_o^\dagger \hat{a}_o - \sum_o \mu \hat{a}_o^\dagger \hat{a}_o \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \\ &= \sum_o \mu \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_o^\dagger \hat{a}_o - \hat{a}_o^\dagger \hat{a}_o (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)). \end{aligned} \quad (\text{D.12})$$

Imposing δ_{om} and δ_{on} , for each point o , we have $\varepsilon_{mu} = \varepsilon_{o=m} + \varepsilon_{o=n}$:

$$\begin{aligned} \varepsilon_{o=m} &= \mu \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_m^\dagger \hat{a}_m - \hat{a}_m^\dagger \hat{a}_m (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)) \\ \varepsilon_{o=n} &= \mu \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_n - \hat{a}_n^\dagger \hat{a}_n (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)). \end{aligned} \quad (\text{D.13})$$

Using that $\hat{a}\hat{a}^\dagger\hat{a} = \hat{a}^\dagger\hat{a}\hat{a} + \hat{a}$ and $\hat{a}^\dagger\hat{a}^\dagger\hat{a} = \hat{a}\hat{a}^\dagger\hat{a}^\dagger - \hat{a}^\dagger$, we expand these expressions to obtain:

$$\begin{aligned}
\varepsilon_{o=m} &= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_m^\dagger \hat{a}_m) - e^{i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_m^\dagger \hat{a}_n - e^{-i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m) \\
&= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m \hat{a}_m^\dagger \hat{a}_m - e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger \hat{a}_m \hat{a}_m^\dagger - e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \\
&= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m + e^{-i\alpha_{mn}} \hat{a}_n^\dagger (\hat{a}_m^\dagger \hat{a}_m + 1)) \hat{a}_m - e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger (\hat{a}_m^\dagger \hat{a}_m + 1) - e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \\
&= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m - e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger) \\
\varepsilon_{o=n} &= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_n^\dagger \hat{a}_n)) + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_n^\dagger \hat{a}_n) - e^{i\alpha_{mn}} (\hat{a}_n^\dagger \hat{a}_n) \hat{a}_m^\dagger \hat{a}_n - e^{-i\alpha_{mn}} (\hat{a}_n^\dagger \hat{a}_n) \hat{a}_n^\dagger \hat{a}_m) \\
&= \mu \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n - e^{-i\alpha_{mn}} \hat{a}_m \hat{a}_n^\dagger).
\end{aligned} \tag{D.14}$$

Therefore $\varepsilon_{\kappa,\mu} = \varepsilon_{o=m} + \varepsilon_{o=n} = 0$, so the κ_{mn} and μ terms commute. Now we evaluate the $\varepsilon_{\kappa,U}$ error term. Starting as before we have:

$$\begin{aligned}
\varepsilon_U &= \left[\sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m), \quad \sum_m U \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m \right] \\
&= \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \sum_o U \hat{a}_o^\dagger \hat{a}_o^\dagger \hat{a}_o \hat{a}_o - \sum_o U \hat{a}_o^\dagger \hat{a}_o^\dagger \hat{a}_o \hat{a}_o \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \\
&= \sum_o U \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_o^\dagger \hat{a}_o^\dagger \hat{a}_o \hat{a}_o - \hat{a}_o^\dagger \hat{a}_o^\dagger \hat{a}_o \hat{a}_o (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)).
\end{aligned} \tag{D.15}$$

Once again we impose δ_{om}, δ_{on} , so for each point o we have $\varepsilon_{\kappa,U} = \varepsilon_{o=m} + \varepsilon_{o=n}$:

$$\begin{aligned}
\varepsilon_{o=m} &= U \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m - \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)) \\
\varepsilon_{o=n} &= U \sum_{\langle m,n \rangle} \kappa_{mn} ((e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_n^\dagger \hat{a}_n \hat{a}_n - \hat{a}_n^\dagger \hat{a}_n^\dagger \hat{a}_n \hat{a}_n (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m)).
\end{aligned} \tag{D.16}$$

Using that $\hat{a}\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a} = \hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a}\hat{a} + 2\hat{a}^\dagger\hat{a}\hat{a}$ and $\hat{a}^\dagger\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a} = \hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a}\hat{a}^\dagger - 2\hat{a}^\dagger\hat{a}^\dagger\hat{a}$, we obtain:

$$\begin{aligned}
\varepsilon_{o=m} &= U \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)) + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)) \\
&\quad - e^{i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \hat{a}_n^\dagger \hat{a}_n - e^{-i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m \\
&= U \sum_{\langle m,n \rangle} \kappa_{mn} (e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m)) + e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \\
&\quad - e^{i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) - e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \\
&\quad - 2e^{i\alpha_{mn}} \hat{a}_n \hat{a}_m^\dagger \hat{a}_m^\dagger \hat{a}_m + 2e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_m \hat{a}_m) \\
&= U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{-i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - e^{+i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) \\
\varepsilon_{o=n} &= U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{+i\alpha_{mn}} (\hat{a}_n^\dagger \hat{a}_n) \hat{a}_m^\dagger \hat{a}_n - e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_n^\dagger \hat{a}_n)).
\end{aligned} \tag{D.17}$$

Therefore the error term $\varepsilon_{\kappa,U} = \varepsilon_{o=m} + \varepsilon_{o=n}$ is:

$$\begin{aligned}
\varepsilon_{\kappa,U} &= U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{-i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - e^{+i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) \\
&\quad + U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{+i\alpha_{mn}} (\hat{a}_n^\dagger \hat{a}_n) \hat{a}_m^\dagger \hat{a}_n - e^{-i\alpha_{mn}} \hat{a}_n^\dagger \hat{a}_m (\hat{a}_n^\dagger \hat{a}_n)).
\end{aligned} \tag{D.18}$$

In the second term, we will swap the m and n indices. Since the second summation is separate from the first and this is just a notational change, we do not switch $\alpha_{mn} = -\alpha_{nm}$, so we have:

$$\begin{aligned}
\varepsilon_{\kappa,U} &= U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{-i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - e^{+i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) \\
&\quad + U \sum_{\langle m,n \rangle} 2\kappa_{mn} (e^{+i\alpha_{mn}} (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - e^{-i\alpha_{mn}} \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)).
\end{aligned} \tag{D.19}$$

Combining these expressions, we obtain the final result:

$$\begin{aligned}
\varepsilon_{\kappa,U} &= U \sum_{\langle m,n \rangle} 2\kappa_{mn} ((e^{-i\alpha_{mn}} + e^{+i\alpha_{mn}}) (\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - (e^{+i\alpha_{mn}} + e^{-i\alpha_{mn}}) \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)) \\
&= 4U \sum_{\langle m,n \rangle} \kappa_{mn} \cos \alpha_{mn} ((\hat{a}_m^\dagger \hat{a}_m) \hat{a}_n^\dagger \hat{a}_m - \hat{a}_m^\dagger \hat{a}_n (\hat{a}_m^\dagger \hat{a}_m)).
\end{aligned} \tag{D.20}$$

Thus, employing the discrete time evolution for the emulation of the κ_{mn} terms with the MZI does impose an error term scaling as $\mathcal{O}(U\kappa \cos \alpha)$. However, for $\alpha_{mn} = \pi/2$, this error term vanishes and perfect emulation is possible. This is interesting and particularly fortunate as $\alpha = \pi/2$ is the hopping phase at which most topological effects are maximized in 2D.

D.2 Deriving nonlinear on-site interaction potential

For lossless and dispersionless media, the polarization density can be expanded in the form [33]

$$P(t) = \epsilon_0 \left[\chi^{(1)} E(t) + \chi^{(2)} E^2(t) + \chi^{(3)} E^3(t) + \dots \right], \quad (\text{D.21})$$

where ϵ_0 is the permittivity of free space, $\chi^{(n)}$ terms are optical susceptibilities, and where $P(t)$ and $E(t)$ are expressed as scalar quantities for simplicity. For media with inversion symmetry (e.g. glass), the $\chi^{(2)}$ term vanishes. [31]

We can write $E(t)$ in the frequency domain as a summation is over frequencies ω_n :

$$E(t) = \sum_n \left[E(\omega_n) e^{-i\omega_n t} + E(-\omega_n) e^{+i\omega_n t} \right], \quad (\text{D.22})$$

where $E(\omega_n) = |E_n| e^{ik \cdot r} / 2 = E(-\omega_n)^*$. The polarization density can similarly be expressed as

$$P(t) = \sum_n \left[P(\omega_n) e^{-i\omega_n t} + P(-\omega_n) e^{+i\omega_n t} \right]. \quad (\text{D.23})$$

Since the Kerr nonlinearity arises from the third-order susceptibility $\chi^{(3)}$, we will focus on this term and refer to it as $\mathcal{P}_{\text{NL}}(\omega)$. We can write this nonlinear polarization vector as:

$$\mathcal{P}_{\text{NL},i}(\omega) = \epsilon_0 \sum_{ijkl} \sum_{(mno)} \chi_{ijkl}^{(3)}(\omega_n, \omega_m, \omega_o) \times E_j(\omega_m) E_k(\omega_n) E_l(\omega_o), \quad (\text{D.24})$$

where $\chi^{(3)}$ is a rank-4 tensor, i, j, k, l are Cartesian coordinates, and (mno) refers to the number of permutations of the distinguishable E fields that yield the same $\omega_m + \omega_n + \omega_o$ term, following the notation of Ref. [33]. The term that gives rise to the intensity-dependent refractive index is:

$$\mathcal{P}_{\text{NL},i}(\omega) = 3\epsilon_0 \sum_{ijkl} \chi_{ijkl}^{(3)}(\omega) \times E_j(\omega) E_k(\omega) E_l(-\omega). \quad (\text{D.25})$$

Assuming the medium is isotropic and homogeneous (e.g. glass), the susceptibility becomes polarization-independent and thus can be treated as a scalar. [234] Without loss of generality, assume the input E fields are linearly-polarized in the x direction, denoted by index $j, k, l = 1$. The electric field energy density for the third-order nonlinear polarization can be expressed as [33]:

$$\mathcal{U}_{\text{NL}} = \frac{3\epsilon_0}{4} \sum_{i111} \chi_{i111}^{(3)} \times E_i^*(\omega + \omega - \omega) E_1(\omega) E_1(\omega) E_1^*(\omega) \quad (\text{D.26})$$

$$= \frac{3\epsilon_0}{4} \sum_{i111} \chi_{i111}^{(3)} \times E_i^*(\omega + \omega - \omega) |E_1(\omega)|^2 E_1(\omega) \quad (\text{D.27})$$

From this expression, the nonlinear polarization \mathcal{P}_{NL} along a direction i can be obtained by

differentiating with respect to E_i^* :

$$\mathcal{P}_{\text{NL},i} = \frac{\partial \mathcal{U}_{\text{NL}}}{\partial E_i^*} \quad (\text{D.28})$$

Defining z to be the direction of field propagation in the fiber, the electric fields in the ring fiber can now be quantized as follows: [140, 65]

$$E = \sum_k i \left(\frac{\hbar \omega_k}{2\epsilon_0 n_0^2 V} \right)^{1/2} (\hat{a}_k e^{-i(\omega_k t - kz)} + \hat{a}_k^\dagger e^{+i(\omega_k t - kz)}) \quad (\text{D.29})$$

where \hat{a}_k^\dagger and \hat{a}_k are the bosonic creation and annihilation operators at the frequency ω_k . E has been normalized to the volume V of the fiber, and n_0 is the refractive index of the fiber. We can express the operators in real space through Fourier transform [223]:

$$\hat{a}_k = \int_0^L \hat{a}_{\omega_k}(z) e^{-ikz} dz \quad (\text{D.30})$$

$$\hat{a}_k^\dagger = \int_0^L \hat{a}_{\omega_k}^\dagger(z) e^{+ikz} dz, \quad (\text{D.31})$$

where $\hat{a}_{\omega_k}(z)$ and $\hat{a}_{\omega_k}^\dagger(z)$ annihilate and create a photon with frequency ω_k at position z in real space, respectively, and where L is the length of the fiber. Since we employ monochromatic light at frequency ω_0 , we only have modes within a narrow bandwidth around ω_0 (k_0). Thus, we can replace the k -space operators in Equation D.29 with the real-space operators $\hat{a}_{\omega_0}(z)$ and $\hat{a}_{\omega_0}^\dagger(z)$, which we will hereafter refer to as a and a^\dagger , respectively. Substituting the quantized electric fields into our expression for the energy density and applying the rotating-wave approximation, we have the following terms in our expression for the total energy of the third-order nonlinear susceptibility:

$$\begin{aligned} H &= \int_V \mathcal{U}_{\text{NL}} d^3 \mathbf{r} = \frac{3\epsilon_0}{4} \int_V \chi^{(3)} |E|^4 d^3 \mathbf{r} \\ &= \frac{3\epsilon_0 \hbar^2 \omega_0^2}{16\epsilon_0^2 n_0^4 V^2} \int_V \chi^{(3)} (\hat{a} \hat{a}^\dagger + \hat{a} \hat{a} e^{-2i(\omega_0 t - k_0 z)} + \hat{a}^\dagger \hat{a}^\dagger e^{+2i(\omega_0 t - k_0 z)} + \hat{a}^\dagger \hat{a})^2 d^3 \mathbf{r} \\ &\approx \frac{3\hbar^2 \omega_0^2}{16\epsilon_0 n_0^4 V^2} \int_V \chi^{(3)} (6\hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + 12\hat{a}^\dagger \hat{a} + 3) d^3 \mathbf{r} \\ &= \left(\frac{9\hbar^2 \omega_0^2}{8\epsilon_0 n_0^4 V^2} \int_V \chi^{(3)} d^3 \mathbf{r} \right) \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + C, \end{aligned} \quad (\text{D.32})$$

where C is some constant corresponding to an overall energy shift [265]. This gives the desired $\hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a}$ term in the Hamiltonian, where the coefficient $U = \frac{9\hbar^2 \omega_0^2}{8\epsilon_0 n_0^4 V^2} \int_V \chi^{(3)} d^3 \mathbf{r}$. Note that we only retain the two-photon interaction term $\hat{a}_m^\dagger \hat{a}_n^\dagger \hat{a}_p \hat{a}_q$ where $m = n = p = q$ in the Hamiltonian since only spatially localized photons in the same time bin interact to achieve the optical Kerr effect, and we are assuming monochromatic light with one frequency mode ω_0 .

D.3 Simulation details

Here we outline the details of the simulations we use to compare the evolution of the states under the emulated Hamiltonians against the exact target Hamiltonians. We use a custom simulation method built with QuTiP [107] for all simulations presented in this paper.

D.3.1 Tractable simulation with a Ponomarev state representation

Suppose the device contains L number of time bins (including the register) and the initial state is a N -photon Fock state. The Hilbert space \mathcal{H} which spans the possible evolutions of this state can thought of as a system of L entangleable $(N+1)$ -qudits (since the vacuum state $|\emptyset\rangle$ must additionally be supported). In this representation the Hilbert space has dimensionality $(N+1)^L$, and is spanned by the creation operators $\hat{a}_\ell^{\dagger n}$, which create $0 \leq n \leq N$ photons in time bin $1 \leq \ell \leq L$. However, this representation is highly degenerate, supporting many invalid states, such as $\bigotimes_{\ell=1}^L \hat{a}_\ell^{\dagger(N-1)} |\emptyset\rangle$, which has more than N total photons in the system. To simulate even the modest 3×3 lattice depicted in Figure 1 of the main text requires operators with dimensionality $59049 \times 59049 \approx 3.5 \times 10^9$.

A more tractable representation can be obtained by inverting the indexing and considering the possible placements of identical bosons into time bins. A number of identical bosons N can be placed into L distinct lattice sites in $\mathcal{N}_N^L = \binom{N+L-1}{L}$ different ways. From this, we can generate a unique labeling of the allowable states in the system using a procedure devised by Ponomarev [192, 191] and described in Ref. [201]. We use the fact that \mathcal{N}_N^L can be generated recursively as $\mathcal{N}^L = \sum_{n=0}^N \mathcal{N}_{N-n}^{L-1}$ to generate a counting scheme which evaluates all \mathcal{N}_n^ℓ up to L, N .

Suppose we have Fock state $|\psi\rangle$ with the occupancies of the L lattice sites described in an N -dimensional vector $(\ell_1, \ell_2, \dots, \ell_N)$, with $m_i \leq m_j$ for $i > j$. The integer label n_ψ of this state is

$$n_\psi = 1 + \sum_{i=1}^N \mathcal{N}_i^{L-\ell_i}. \quad (\text{D.33})$$

For example, as shown in Figure D.1, with $L = 8, N = 4$, the Fock state $|\psi\rangle = |0, 0, 2, 0, 1, 0, 0, 1\rangle$ maps to $(\ell_1, \ell_2, \ell_3, \ell_4) = (8, 5, 3, 3)$, and the integer label is $n_\psi = 1 + \mathcal{N}_1^3 + \mathcal{N}_2^3 + \mathcal{N}_3^5 + \mathcal{N}_4^8 = 112$. The inverse mapping is found with an iterative process: given n_ψ , we find the largest ℓ_N such that $\mathcal{N}_N^{\ell_N} < n_\psi$, then find the largest ℓ_{N-1} such that $\mathcal{N}_{N-1}^{\ell_{N-1}} < n_\psi - \mathcal{N}_N^{\ell_N}$, and so on.

To construct annihilation operators, we consider the direct sum of all smaller Hilbert spaces

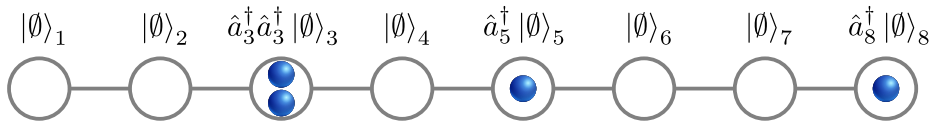


Figure D.1: An example Fock state $|\psi\rangle = |0, 0, 2, 0, 1, 0, 0, 1\rangle$, which can be re-indexed by boson number as $|\psi'\rangle = |8, 5, 3, 3\rangle$. This maps to the Ponomarev basis state $|n_\psi\rangle = |112\rangle$.

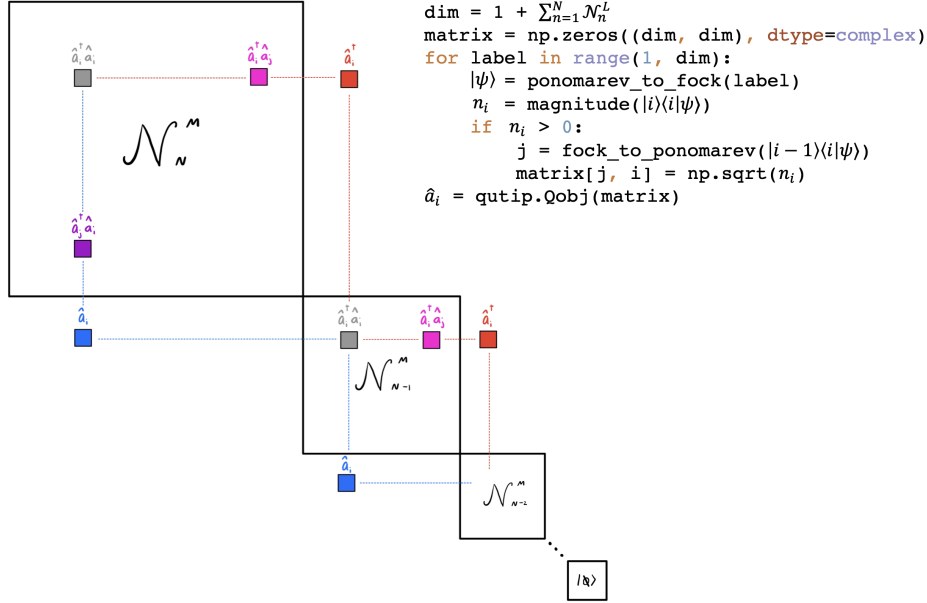


Figure D.2: Construction of an annihilation operator in the bosonic lattice representation. Python pseudocode showing the process for constructing a QuTiP object from the Ponomarev indexing functions is included in the upper right.

$\mathcal{H}_{n \leq N}^L = \left[\bigoplus_{n=0}^N \mathcal{H}_n^L \right]$. The annihilation operators \hat{a}_i are constructed from a $(\sum_n \mathcal{N}_n^L \times \sum_n \mathcal{N}_n^L)$ -dimensional matrix of zeros by iterating over all basis state labels $n_\psi = 1, 2, \dots, \mathcal{N}_N^L$. To construct \hat{a}_i , for each Fock state $|k_1 k_2 \dots k_i \dots k_L\rangle$ with $k_i > 0$, a transition element is added which maps $|k_1 k_2 \dots k_i \dots k_L\rangle \mapsto \sqrt{k_i} |k_1 k_2 \dots (k_i - 1) \dots k_L\rangle$. The corresponding creation operator \hat{a}_i^\dagger is simply the Hermitian conjugate of this operator, and photon expectation values in time bin i are still $\langle \hat{a}_i^\dagger \hat{a}_i \rangle$ in this representation. The matrices are converted to QuTiP operators and the rest of the simulation is carried out normally. This process is illustrated in Figure D.2.

This representation is especially effective for simulating systems with many more lattice sites than bosons $L \gg N$. For a 10×10 lattice containing two photons, the dimensionality of the state vectors is reduced from 5×10^{47} to 5050.

D.4 Details for bandstructure computation

The band structures in Fig. 2 of the main text are computed for the synthetic case by simulating one iteration of the propagator $\hat{G} = e^{-i\hat{H}(t=1)}$ in the device, taking the matrix logarithm $\hat{H} = \frac{\log \hat{G}}{-i}$, and then diagonalizing \hat{H} . In both the emulated and exact cases, the Hamiltonian is represented in real space in the computer simulation; for each eigenstate of \hat{H} with eigenvalue E , we compute k values using peak detection of its Fourier transform, as shown in Figure D.3. This can result in

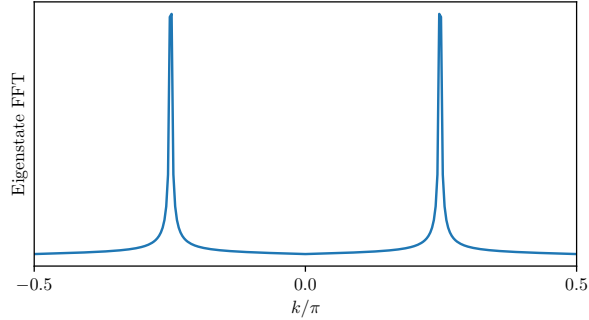


Figure D.3: Computing the k values for a lattice eigenstate using peak detection of the Fourier transform.

small numerical instabilities which are present in both the exact and emulated cases. Cases where the Fourier transform does not have clear peaks may result in outlying points with errant values of k ; highly outlying points in Figure 2(b) of the main text have been pruned, but do not comprise more than 1% of the data.

D.5 Dispersion analysis

In the linear regime of small U (where we can ignore the effects of nonlinear pulse broadening), if we assume the same initial pulse shape with pulse length δx within each bin of size Δx (with $\delta x \ll \Delta x$), and that the register and storage loops are made of the same fiber, then all pulses will deform equally over time, so pulse distinguishability is not an issue. The limiting factor will therefore be the dispersive length over which a pulse will broaden to the point where $\delta x \sim \Delta x$. Commercially available dispersion-shifted fibers allow for simultaneous low attenuation of ~ 0.2 dB/km and low dispersion of ~ 4 ps/nm/km at $\lambda = 1550$ nm. If we assume a bin size of $\Delta x = 20$ cm (time bin size of 1 ns), a pulse length of $\delta x = 2$ cm, and we saturate the uncertainty limit $\Delta\lambda = \lambda^2/4\pi\delta x$, then this dispersive length is approximately 26000 km of fiber. At these distances, attenuation losses would certainly dominate, so we can safely ignore dispersive errors.

Appendix E

Simulator for quantum networks and channels

In this Appendix, we describe the open-source distributed quantum network simulation framework **SQUANCH** (Simulator for Quantum Networks and Channels) used to perform many of the quantum gate simulations in Chapters 4 and 5. The framework includes many features of a general-purpose quantum computing simulator, but it is optimized specifically for simulating quantum networks. It includes functionality to allow users to easily design complex multi-party quantum networks, extensible classes for modeling noisy quantum channels, and a multiprocessed NumPy backend for performant simulations. We present an overview of the structure of the library, describing how the various API elements represent the underlying physics and providing simple usage examples for each module. Finally, we present several demonstrations of canonical quantum information protocols implemented using this framework.

In Section E.1 we give a brief primer on quantum information and introduce quantum agents, channels, and networks. In Section E.2, we present an overview of the **SQUANCH** framework. We first present the classes that represent quantum information, detailing the **QSystem**, **Qubit**, **QStream**, and **Gates** modules. Next, we describe the higher-abstraction components which are used to build the nodes and connections of a quantum network: **Agents**, **Channels**, and **Errors**. We provide pedagogical demonstrations throughout each of these discussions. Finally, in Section E.3, we show a series of more sophisticated demonstrations of canonical quantum information protocols implemented with the framework.

E.1 Preliminaries

In this section, we give a brief primer on quantum computation¹, reviewing some of the concepts necessary to understand the SQUANCH framework. In Section E.1.5, we also provide some basic formalism for describing quantum networks and channels. We encourage readers with familiarity in these topics to skip to Section E.2 for an overview of our platform. Throughout this section, a basic knowledge of linear algebra and quantum mechanics is assumed.

E.1.1 Qubits and quantum systems

In quantum computation, information is stored in the state of a *qubit*², the quantum generalization of a classical bit. Physically, a qubit can be any two-level quantum system, such as spin or polarization. The state³ of a qubit is a vector $|\psi\rangle$ over \mathbb{C}^2 in a linear superposition of states $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. The amplitudes $\alpha, \beta \in \mathbb{C}$ obey normalization, such that $|\alpha|^2 + |\beta|^2 = 1$. The state vectors $\{|0\rangle, |1\rangle\}$ (physically, the eigenstates of the logical observable) form the *computational basis* which spans the two-dimensional Hilbert space \mathcal{B} containing $|\psi\rangle$. If a *measurement* is performed on the qubit state, the state collapses to $|0\rangle$ or $|1\rangle$, “returning” a classical bit 0 or 1, with probability $|\alpha|^2$ or $|\beta|^2$, respectively.

A collection of qubits forms a multi-particle *quantum system*. The state of an N -qubit quantum system lives in a 2^N -dimensional Hilbert space \mathcal{H}_N and can be represented by a state vector:

$$|\Psi\rangle \in \mathcal{H}_N \equiv \bigotimes_{k=N}^1 \mathcal{B}_k, \quad (\text{E.1})$$

where $\bigotimes_{k=N}^1$ denotes a rightward tensor product ordered as $k = N, N-1, \dots, 1$ and \mathcal{B}_k is the state space for each constituent qubit. The system state space \mathcal{H}_N is spanned by the product of the qubit eigenstates $\bigotimes_{k=N}^1 |q_k\rangle$, for $q_k \in \{0, 1\}$. These basis vectors are (also) referred to as the computational basis for the N -qubit system; they can be written as $|q_1 q_2 \dots q_N\rangle$ and have the same lexicographic ordering as their binary representations (e.g. $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ for a two-qubit system).

In general, a multi-qubit state $|\Psi\rangle$ cannot be represented as the tensor product of the states of its qubits; that is:

$$|\Psi\rangle \neq |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle, \quad (\text{E.2})$$

so the state of each qubit is generally not independent of other qubits. (If $|\psi\rangle$ does take this form, it is

¹For an excellent in-depth review of this topic, we direct readers to the canonical textbook in quantum computation by Nielsen and Chuang [168].

²There are in fact multiple equivalent models of quantum computation. The prevailing model represents information with qubits, but other models can use discrete k -level quantum systems (qudits), or even continuous quantum variables, as detailed in [114].

³What we refer to as the “state” is more correctly the projection of the quantum state onto the basis of the observable in question.

called a separable state.) This non-local correlation between quantum states is called entanglement. The massive parallelism of quantum computation is due to superposition and entanglement, as they allow the size of the state space of a quantum system to scale exponentially with the number of qubits. By cleverly exploiting superposition and entanglement, quantum algorithms can achieve polynomial [86] or even exponential [229] speedup over classical algorithms for important problems.

E.1.2 Open quantum systems

The states for quantum systems described in Section E.1.1 are *pure* states, which encode all that can be known about a quantum system. However, in many cases, the full quantum state is not accessible to an observer (for example, if the system is a subset of a larger quantum system or if the state is coupled to the environment). In this case, the available quantum state is a statistical mixture of pure states $|\Psi_j\rangle$, each with probability p_j , which can be represented by a density matrix $\hat{\rho}$:

$$\hat{\rho} = \sum_j p_j |\Psi_j\rangle\langle\Psi_j|. \quad (\text{E.3})$$

If a quantum system with (pure) state $|\Psi\rangle$ is divided into two subsystems A and B , spanned by orthonormal bases $\{|\alpha_i\rangle\}$ and $\{|\beta_j\rangle\}$, respectively, then $|\Psi\rangle = \sum_{ij} |\alpha_i\rangle |\beta_j\rangle$ can equivalently be expressed by the density matrix:

$$\hat{\rho} = |\Psi\rangle\langle\Psi| = \sum_{ij} \sum_{i'j'} c_{ij} c_{i'j'}^* |\alpha_i\rangle |\beta_j\rangle \langle\alpha_{i'}| \langle\beta_{j'}|. \quad (\text{E.4})$$

If an observer only has access to subsystem A , then their (mixed) state $\hat{\rho}_A$ is given by the partial trace over all inaccessible degrees of freedom (that is, B):

$$\hat{\rho}_A = \text{tr}_B \hat{\rho} = \sum_j \langle\beta_j| \hat{\rho} |\beta_j\rangle. \quad (\text{E.5})$$

If a measurement of $\hat{\rho}$ can yield possible outcomes of $\{|j\rangle\}$, then the probability of each outcome is given by $p_j = \text{tr}[\hat{\rho}|j\rangle\langle j|]$ and the measurement collapses the state as:

$$\hat{\rho} \mapsto \hat{\rho}_j = \frac{|j\rangle\langle j| \hat{\rho} |j\rangle\langle j|^\dagger}{p_j}. \quad (\text{E.6})$$

E.1.3 Quantum gates

Just as classical algorithms can be represented as a sequence of Boolean logic gates⁴, quantum algorithms use quantum gates to manipulate the states of quantum systems. A quantum gate for

⁴More precisely, a Turing machine with arbitrary but finite memory can be constructed from a functionally complete set of logic gates.

an N -qubit system is a unitary operator $U \in \text{U}(2^N)$; when applied to a quantum state $\hat{\rho}$, the gate modifies the state as $\hat{\rho} \mapsto U\hat{\rho}U^\dagger$.

If a gate acts on a subset of a larger quantum system, their matrix representations are “padding” with the identity operator. For example, if $\hat{\rho}$ represents an N -qubit system, and a single-qubit gate U is applied to qubit k , then the state transforms as:

$$\hat{\rho} \mapsto U_k \hat{\rho} U_k^\dagger : U_k \equiv \mathbb{1}_N \otimes \cdots \otimes U \otimes \mathbb{1}_{k-1} \otimes \cdots \otimes \mathbb{1}_1, \quad (\text{E.7})$$

where $\mathbb{1}_j$ denotes identity applied to the j th qubit. Quantum gates are commonly described pictographically with quantum circuit diagrams, such as the one in Figure E.1, which depicts the transformation in Equation E.7. A list of circuit symbols for built-in gates included in **SQUANCH** is provided in Section E.5.3.

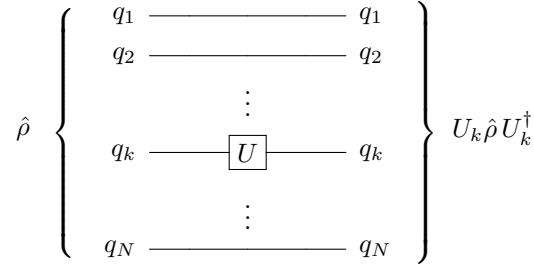


Figure E.1: Circuit diagram for applying the single-qubit gate U to qubit q_k . Wires implicitly denote tensor products with identity.

It is straightforward to define many (and in fact, using **SWAP**, all) multi-qubit gates using a similar strategy as in Equation E.7. For example, the controlled-X or **CNOT** gate, common for constructing entanglement in a quantum system, has a matrix representation for a two-qubit system:

$$\text{CNOT} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (\text{E.8})$$

Equivalently, we can define the gate CNOT_{jk} acting on control qubit j and target qubit k in an N -qubit system as:

$$\text{CNOT}_{jk} \equiv |0\rangle\langle 0|_j \otimes \mathbb{1}_k + |1\rangle\langle 1|_j \otimes \sigma_{x;k}, \quad (\text{E.9})$$

where $\sigma_{x;j}$ is Pauli-X applied to qubit j and padding with identity is implicit.⁵

⁵For the remainder of this paper, we adopt the convention that a k -qubit operator Ω acting on qubits $i_1 \cdots i_k$ is notated $\Omega_{i_1 \cdots i_k}$ and implicitly includes ordered tensor products with $\mathbb{1}_j$ for $j \notin \{i_1 \cdots i_k\}$.

E.1.4 Quantum agents

Algorithms for transmitting and manipulating quantum information are frequently explained in terms of a colorful cast of characters starring Alice and Bob. In the SQUANCH framework, these “quantum actors” are represented by **Agents**. An agent has a set of classical and quantum channels connecting it to other agents, classical memory, which can store arbitrary data, and quantum memory, which is an array of qubits. It also has runtime logic in the form of a classical program with access to operations that can manipulate quantum information, such as gates and measurement. For completeness, a formal definition of a quantum agent is given in Section E.5.2.

E.1.5 Quantum networks and channels

In this section, we provide some formalism to define quantum channels and quantum networks as used in our simulation framework.

Conceptually, a *quantum channel* is a communication channel which transmits quantum information (qubits) and an associated noise model⁶ \mathcal{N} which modifies the qubits, generally through coupling to the environment. Mathematically, \mathcal{N} is a linear completely-positive trace-preserving map which acts on a composite state consisting of the input quantum state $\hat{\rho}_{\text{in}}$ and the environment $\hat{\xi}_{\text{in}}$ (assumed by convention to be in the initial state $|0\rangle\langle 0|$). [87]

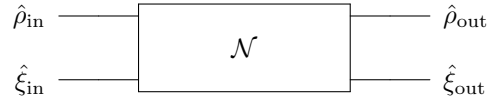


Figure E.2: Circuit diagram representing a quantum channel \mathcal{N} . Frequently, the coupling of the channel to the environment is assumed and not explicitly drawn, as in Figure E.11.

While qubits are in the channel, the state of the composite system undergoes unitary evolution as $\hat{\rho}_{\text{in}} \otimes \hat{\xi}_{\text{in}} \mapsto U(\hat{\rho}_{\text{in}} \otimes \hat{\xi}_{\text{in}})U^\dagger$, which can allow information to “leak” from the system $\hat{\rho}$ to the environment $\hat{\xi}$. After the qubit is retrieved from the channel, the output state $\hat{\rho}_{\text{out}}$ is given by tracing over inaccessible environmental degrees of freedom [87]:

$$\hat{\rho}_{\text{out}} = \mathcal{N}(\hat{\rho}_{\text{in}}) = \text{tr}_\xi \left[U(\hat{\rho}_{\text{in}} \otimes \hat{\xi}_{\text{in}})U^\dagger \right]. \quad (\text{E.10})$$

We direct readers to papers by Gyongyosi, et al., for in-depth discussions of the properties [87] and capacities [88] of quantum channels.

Finally, a *quantum network* can be defined⁷ as a directed graph $N = (A, C)$, where A is a set

⁶Most common definitions of a quantum channel define the channel as the noise model \mathcal{N} itself; as before, we make the distinction to mirror the codebase structure.

⁷Quantum networks generally lack the standard definition that quantum computers or quantum channels have. We provide a formal definition here for completeness, but it should not be interpreted as authoritative.

of agents and C is a set of quantum and classical channels⁸. Other practical network components, such as quantum repeaters [14], can be modeled (both mathematically and programmatically) as quantum agents.

E.2 Framework overview

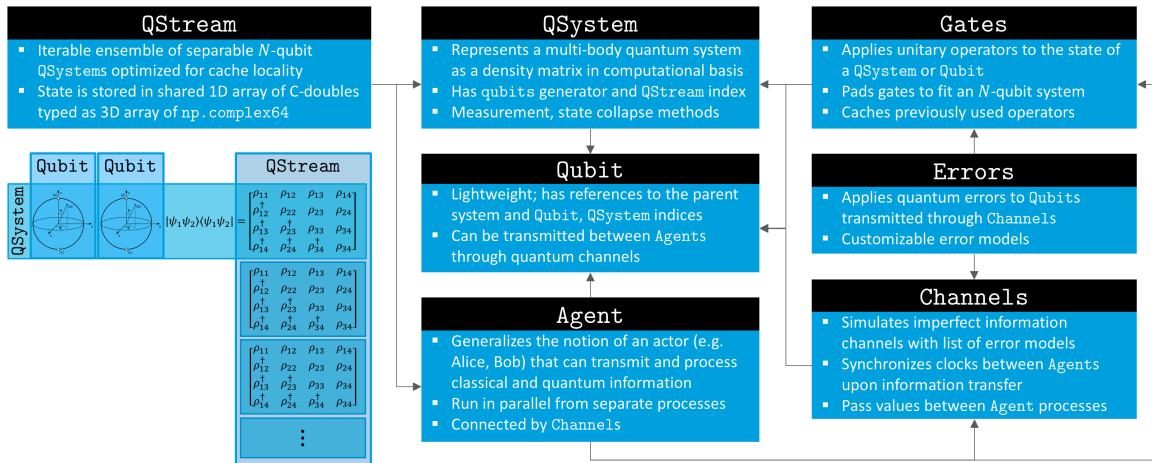


Figure E.3: Schematic overview of the modules available in the SQUANCH framework, with an illustration in the lower left showing how quantum streams and states are represented. The **QSystem** is the most fundamental class, representing a multi-particle quantum state as a density matrix. Ensembles of quantum systems are efficiently handled by **QStreams**, and each **QSystem** has references to its constituent **Qubits**. Functions in the **Gates** module can be used to manipulate the state of a quantum system. **Agents** are generalized quantum-mechanical “actors” which are initialized from a **QStream** instance and can alter the state of the quantum systems in their stream object, typically by interacting directly with **Qubits**. **Agents** run in parallel from separate processes and are connected by quantum and classical **Channels**, which apply customizable **Error** models to the transmitted information and synchronize agent clocks.

SQUANCH⁹ is a Python-based framework for simulating the dynamics of quantum networks and provides an intuitive, object-oriented API for representing and manipulating ensembles of separable quantum systems. A schematic overview of the SQUANCH framework is depicted in Figure E.3. In this section, we describe the structure of the framework, detailing the modules to represent and manipulate quantum information and to define and simulate quantum networks.

⁸Since our definition of a quantum agent includes a list of channels, explicitly including C in the definition is redundant but adds clarity.

⁹This manuscript refers to SQUANCH version 1.1.0, available at github.com/att-innovate/squanch or from the Python Package Index using `pip install squanch`. Documentation is available at att-innovate.github.io/squanch.

E.2.1 Working with quantum information

QSystems and Qubits

The most fundamental class in the framework is the `QSystem`, which represents the state of a multi-body, maximally entangleable quantum system. The state of each qubit $|\psi_k\rangle$ of a quantum system is represented in the abstract computational basis¹⁰ \mathcal{B}_k spanned by $\{|0\rangle_k, |1\rangle_k\}$, and the state of the system as a whole is represented by a density matrix, which is implemented in memory as a NumPy array of `np.complex64` values.

A `QSystem` contains a `QSystem.qubits` generator,¹¹ which enumerates the qubits of the quantum system. Each qubit is represented by a `Qubit` object, a lightweight wrapper class containing only a reference to the parent `QSystem` and the index of the qubit within the system¹².

`Qubit` and `QSystem` objects contain methods for simulating qubit measurement in the computational basis, as described in Sections E.1.1 and E.1.2. `Qubit.measure()` calls `QSystem.measure_qubit(index)`, which returns a bit value for the result of the measurement. Measurement of qubit k partially collapses the system state $\hat{\rho}$ by choosing a random outcome $|\psi_k\rangle = |i\rangle$ with probability $p_{i;k} = \text{tr}[\hat{\rho}(|i\rangle\langle i|)_k]$ for $i \in \{0, 1\}$, modifying the state according to Equation E.6.

`QSystems` also contain a method, `QSystem.apply()`, for applying an N -dimensional unitary operator to modify the system state. In practice, this method is rarely used, as qubits can be more intuitively manipulated using functions in the `gates` module, which provides a set of common single- and multi-qubit quantum logic gates.

Quantum gates

Quantum gates are implemented as standard Python functions which take one or more `Qubits` as arguments, return nothing, and call `Qubit.apply()` or `QSystem.apply()` to apply a unitary operator to the quantum state. A list of natively-included gates is provided in Section E.5.3. The corresponding N -qubit operator (as described in Section E.1.3) for each gate is lazily evaluated, cached, and applied directly to the the parent `QSystem.state`, modifying its density matrix in-place. A simple example of state preparation and measurement using built-in gate functions is demonstrated¹³ in Program E.1.

```

qsys = QSystem(2)    # create a two-qubit system
q1, q2 = qsys.qubits # enumerate the qubits
H(q1)                # Hadamard on qubit 1

```

¹⁰See Section E.1.1 for more detail.

¹¹Generators are used instead of lists for performance reasons, although they do introduce a small number of peculiarities in the code, which are discussed in the documentation.

¹²If the parent system is part of a `QStream`, the `Qubit` will also contain an index for the location of the system within the stream.

¹³For the remainder of this paper, we assume that all programs are run with the standard import statements of `import numpy as np, import matplotlib.pyplot as plt, and from squanch import *`.

```

CNOT(q1, q2)      # Control-NOT of q1 on q2
bit1 = q1.measure() # measure the qubits
bit2 = q2.measure()
assert bit1 == bit2

```

Program E.1: Preparing and measuring the entangled state $|+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The measurement results `bit1` and `bit2` will always be equal.

Additional gates can be added through compositions of existing gates, which can approximate any unitary operator with arbitrary accuracy [18], or by directly writing a function to implement the gate operation. As an example, consider the quantum Fourier transform [168], which operates on the 2^N lexicographically-enumerated N -qubit basis states $\{|j\rangle, |k\rangle \in \bigotimes_{k=N}^1 \mathcal{B}_k\}$ as:

$$\sum_j \alpha_j |j\rangle \mapsto \sum_k \left(\frac{1}{\sqrt{2^N}} \sum_{n=0}^{2^N-1} e^{\frac{2\pi i kn}{2^N}} \alpha_n \right) |k\rangle. \quad (\text{E.11})$$

This operation can be implemented iteratively with the quantum circuit depicted in Figure E.4.

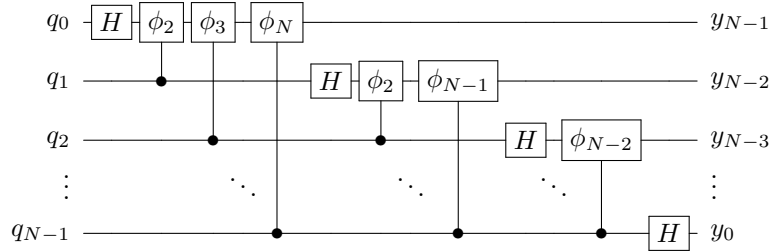


Figure E.4: Circuit implementation of a quantum Fourier transform. ϕ_m denotes the phase gate $\begin{pmatrix} 1 & 0 \\ 0 & \omega_m \end{pmatrix}$, where $\omega_m \equiv e^{\frac{2\pi i}{2^m}}$ is a primitive root of unity.

Programmatically implementing this relatively complicated quantum circuit is straightforward in SQUANCH, as demonstrated in Program E.2.

```

def QFT(qubits):
    '''Applies quantum Fourier transform to inputs'''
    N = len(qubits)
    for n in range(N, 0, -1):
        target = qubits[N - n]
        H(target)
        for m in range(1, n):
            CPHASE(qubits[N - n + m], target, 2*np.pi/2**(m+1))

```

Program E.2: Implementing the quantum Fourier transform using built-in gates. The `CPHASE` operator is the controlled-phase gate, which acts as the identity if the control qubit is in state $|0\rangle$ and maps $|0\rangle \rightarrow |0\rangle$ and $|1\rangle \rightarrow e^{i\phi} |1\rangle$ if the control qubit is in state $|1\rangle$.

QStreams – performant quantum ensembles

An ensemble of separable quantum systems, such as one million Bell pairs, is represented by a “quantum stream”, or `QStream`, which stores the collective state of its `QSystems` in a contiguous block of pre-allocated memory to optimize cache locality. A `QStream` is instantiated from two dimensions: the number of qubits in each quantum system, and the number of systems in the stream. The `QStream` is an iterable Python object, and most simulations will contain an iteration over the quantum systems in the stream. To minimize overhead, `QSystems` can be quickly instantiated from an existing section of the `QStream.state` array.

By default `QStream` state is stored in a shared memory as a C-type array of `doubles`, which is type-casted as a 3D array of `np.complex64` values. Storing the state in shared memory allows multiple agents to work in parallel. For performance reasons, there is no explicit concurrency safety when a system is modified by multiple agents, as sending and receiving `Qubits` are blocking operations which allow for naturally safe parallelism.

E.2.2 Simulating quantum networks

The top-level classes that provide the greatest abstraction are `Agents` and `Channels`, which implement the nodes and connections in a quantum network, respectively.

Agents – Alice and Bob in code

An `Agent` generalizes the notion of an actor (e.g. Alice, Bob) that can send, receive, store, and manipulate classical and quantum information. We provide a formal definition for a quantum agent in Section E.1.4; the structure of the `Agent` class mirrors this definition.

`Agents` are instantiated from a `QStream` object, and each `Agent` has a classical memory, which can store arbitrary Python objects, a quantum memory, which stores incoming qubits and can be modified to simulate error models. `Agents` have a processor in the form of a `run()` method that implements runtime logic. Additionally, `Agents` maintain internal clocks which are updated when sending or receiving information to each other; this allows users to roughly quantify the performance of various networking protocols in terms of simulated elapsed time. During simulations, `Agents` run in parallel from separate processes, synchronizing clocks and passing information between each other through `Channels` using `qsend/qrecv` and `csend/crecv`.

A typical workflow for creating a quantum network simulation using `Agents` is:

1. Define the runtime logic for each agent class. This usually involves an iteration over `self.qstream`, calls to send and receive information to/from other agents, and a call to output the results of the simulation to the main process using `self.output(<objects>)`.
 - If the agent is a “sender”, the runtime logic typically iterates for `qsystem` in `self.qstream`, followed by procedures to manipulate and transmit the qubits of `qsystem`.

- If the agent is a “receiver”, the runtime logic is usually `for _ in self.qstream: qubit = self.qrecv(<other_agent>)`, followed by processing and output procedures.
 - Of course, mixtures of both formats may be used.
2. Create and format a `QStream` object. State preparation may be done as part of the agent logic or in the main process.
 3. Create an output dictionary (e.g. `out = Agent.shared_output()`) and make instances for each agent class from the quantum stream (e.g. `alice = Alice(qstream, out)`).
 4. Define the structure of the network by connecting agent instances using `alice.qconnect(bob)` and/or `alice.cconnect(bob)`, optionally specifying the channel model(s) to use.
 5. Run the simulation using `Simulation(alice, bob, ...).run()` and post-process and visualize the output data. Output results for each agent will be returned to corresponding value in the previously created `out` dictionary (e.g. `alice_out = out["Alice"]`).

This general program structure is demonstrated in Program E.3.

```
class Alice(Agent):
    '''Alice sends states to Bob and receives results'''
    def run(self):
        measurements = []
        for i, qsys in enumerate(self.qstream):
            q = qsys.qubit(0)
            if (i % 2 == 0): X(q) # flip even qubits
            self.qsend(bob, q)
            result = self.crecv(bob)
            measurements.append(result)
        self.output({"meas": measurements,
                    "t": self.time})

class Bob(Agent):
    '''Bob measures Alice's states and sends outcomes'''
    def run(self):
        for _ in self.qstream:
            q = self.qrecv(alice)
            self.csend(alice, q.measure())
        self.output({"t": self.time})

# Set up QStream and Agent instances
```

```

qstream = QStream(1, 10)
out = Agent.shared_output()
alice = Alice(qstream, out)
bob = Bob(qstream, out)

# Connect agents with classical and quantum channels
alice.qconnect(bob)
alice.cconnect(bob)

# Run the simulation
Simulation(alice, bob).run()
print("Measurements: {}".format(out["Alice"]["meas"]))
# >> Measurements: [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]

```

Program E.3: A simple quantum network “ping” protocol implemented using agents. Alice prepares single-qubit states to send to Bob, who measures them and replies with the outcomes.

Channels – imperfect communication lines

Classical and quantum channels, which are derived from the `CChannel` and `QChannel` base classes, represent noisy information channels physically connecting distant nodes in a network. A formal definition of a quantum channel is provide in Section E.1.5; a practical example would be a fiber optic line with a specified length and attenuation coefficient. Functionally, `Channels` are multiprocessed queues that allow routines to communicate between processes. `Channels` track transmission times and speed-of-light delays (through an optional `length` parameter which can specify the physical size of the channel in kilometers) and apply customizable error models to the transmitted information. Quantum error models are implemented by extending the base `QError` class, which maintains a reference to the parent channel and has a single `apply(qubit)` method to apply the error model to a transmitted qubit. An implementation of a simple quantum error model is shown in Program E.4.

```

class SimpleError(QError):
    def apply(self, qubit):
        '''Flip the qubit with 50% probability'''
        if np.random.rand() < 0.5:
            X(qubit)
        return qubit

class SimpleQChannel(QChannel):
    def __init__(self, *args, **kwargs):
        QChannel.__init__(self, *args, **kwargs)

```

```

        self.errors = [SimpleError(self)]

# Set up QStream and Agent instances
qstream = QStream(1, 10)
out = Agent.shared_output()
alice = Alice(qstream, out)
bob = Bob(qstream, out)
alice.pulse_length = 10e-9 # 10ns pulse length
bob.pulse_length = 10e-9

# Connect agents with new model, specify 1km length
alice.qconnect(bob, SimpleQChannel, length=1.0)
alice.cconnect(bob, length=1.0)

# Run the simulation
Simulation(alice, bob).run()

alice_out, bob_out = out["Alice"], out["Bob"]
print("Alice's time: {}".format(alice_out["t"]))
print("Bob's time: {}".format(bob_out["t"]))
print("Measurements: {}".format(alice_out["meas"]))
# >> Alice's time: 8.761114076050701e-05
# >> Bob's time: 8.427558372248166e-05
# >> Measurements: [0, 1, 0, 0, 1, 1, 0, 1, 0, 1]

```

Program E.4: An extension of the quantum “ping” protocol from Program E.3 which adds simple quantum error models to the network. (Alice and Bob are defined as before.) Agent timing functionality is also demonstrated given a specified channel length of 1km.

E.3 Demonstrations

Finally, in this section we present several demonstrations of canonical experiments and protocols involving quantum information. For brevity, we omit code for plotting or displaying images and assume standard imports. The SQUANCH documentation website (see Section E.5.1) contains the full source code for each demonstration, along with more detailed step-by-step explanations.

E.3.1 Quantum teleportation

Quantum teleportation allows two parties that share an entangled pair to transfer an arbitrary quantum state using only classical communication [27]. This process has tremendous applicability

to quantum networks, transferring fragile quantum states between distant nodes. Conceptually, quantum teleportation is the inverse of superdense coding.

In general, all quantum teleportation experiments have the same underlying structure. Two distant parties, Alice and Bob, are connected via a classical information channel and share a maximally entangled state. Alice has an unknown state $|\psi\rangle$ which she wishes to send to Bob. She performs a joint projective measurement of her state and her half of the entangled state and communicates the outcomes to Bob, who operates on his half of the entangled state accordingly to reconstruct $|\psi\rangle$. In this demo, we'll implement the canonical two-party quantum teleportation protocol:

1. Alice generates an entangled two-particle state $|AB\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, keeping half of the state and sending the other half to Bob.
2. Alice entangles her qubit $|\psi\rangle$ with her ancilla $|A\rangle$ by applying controlled-not and Hadamard operators.
3. Alice jointly measures $|\psi\rangle$ and $|A\rangle$ and communicates the outcomes to Bob through a classical channel. Bob's qubit is now in one of four possible Bell states, one of which is $|\psi\rangle$, and he will use Alice's two bits to recover $|\psi\rangle$.
4. Bob applies a Pauli-X operator to his qubit if Alice's ancilla A collapsed to $|1\rangle$, and he applies a Pauli-Z operator to his qubit if her state $|\psi\rangle$ collapsed to $|1\rangle$. He has thus transformed $|B\rangle \mapsto |\psi\rangle$.

This protocol is illustrated in the circuit diagram shown in Figure E.5.

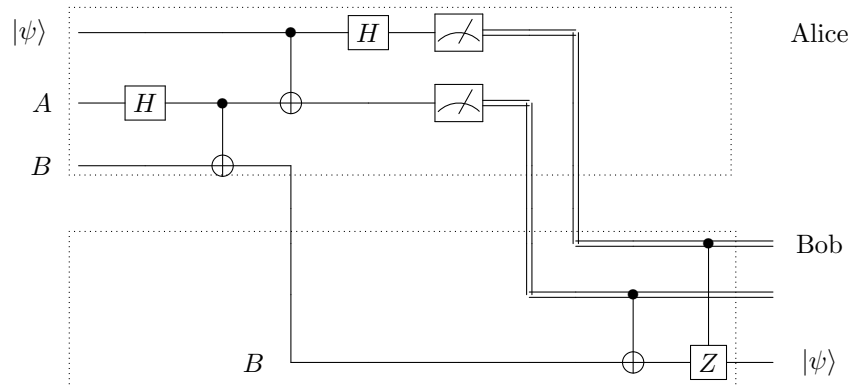


Figure E.5: Circuit diagram for the two-party quantum teleportation simulation described above.

For this demonstration, we'll prepare an ensemble of qubits in the state $|q_\theta\rangle = R_X(\theta)|0\rangle$ for various values of $\theta \in [0, 2\pi]$ and compare the expected and observed outcomes. (See Section E.5.3 for the definition of $R_X(\theta)$.) The results of the simulated experiment are shown in Figure E.6.

```

class Alice(Agent):
    '''Alice teleports qubits to Bob'''
    def distribute_bell_pair(self, a, b):
        # Create a Bell pair and send half to Bob
        H(a)
        CNOT(a, b)
        self.qsend(bob, b)

    def teleport(self, q, a):
        # Perform the teleportation
        CNOT(q, a)
        H(q)
        # Tell Bob what to do over classical channel
        apply_x = a.measure()
        apply_z = q.measure()
        self.csend(bob, [apply_x, apply_z])

    def run(self):
        for qsystem in self.qstream:
            q, a, b = qsystem.qubits
            self.distribute_bell_pair(a, b)
            self.teleport(q, a)

class Bob(Agent):
    '''Bob receives Alice's qubit and measures it'''
    def run(self):
        measurement_results = []
        for _ in self.qstream:
            # Bob receives a qubit from Alice
            b = self.qrecv(alice)
            # Bob receives classical instructions
            apply_x, apply_z = self.crecv(alice)
            if apply_x: X(b)
            if apply_z: Z(b)
            # Measure the output state
            measurement_results.append(b.measure())
        # Put results in output object
        self.output(measurement_results)

angles = np.linspace(0, 2 * np.pi, 50) # R_X angles

```

```

num_trials = 250 # number of trials for each angle

# Prepare the initial states in the stream
qstream = QStream(3, len(angles) * num_trials)
for angle in angles:
    for _ in range(num_trials):
        q, _, _ = qstream.next().qubits
        RX(q, angle)

# Make agent instances and connect
out = Agent.shared_output()
alice = Alice(qstream, out = out)
bob = Bob(qstream, out = out)
alice.qconnect(bob)
alice.cconnect(bob)

# Run the simulation and plot the results (omitted)
Simulation(alice, bob).run()

```

Program E.5: Implementation of a two-party quantum teleportation experiment using SQUANCH.

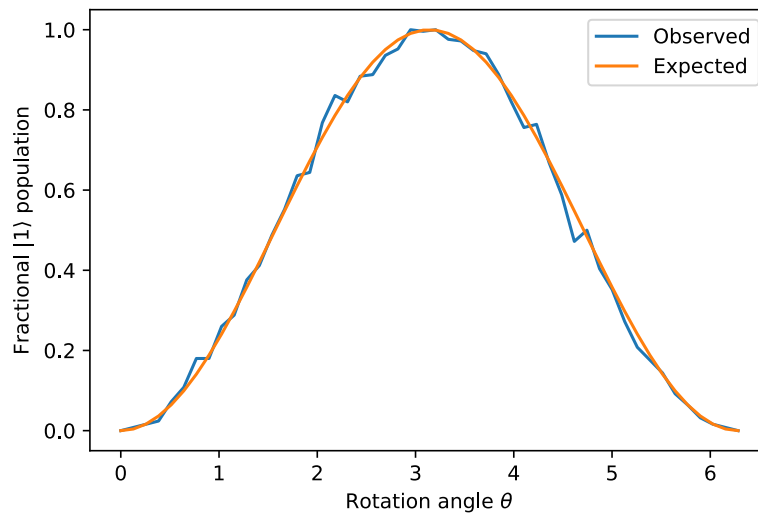


Figure E.6: Observed and expected fractional populations of measurements resulting in $|\psi\rangle \rightarrow |1\rangle$ for the two-party quantum teleportation simulation with an ensemble size of 250.

E.3.2 Superdense coding

Superdense coding is a process whereby two parties connected via a quantum channel and sharing an entangled pair can send two classical bits of information using only a single qubit [28]. Conceptually, superdense coding is the inverse of quantum teleportation.

In this demonstration, we'll implement the three-party superdense coding protocol depicted in the circuit diagram in Figure E.7. Charlie distributes entangled particles to Alice and Bob. Alice encodes her information in her particles and sends them to Bob, who decodes the information by matching Alice's qubits with his half of the shared state received from Charlie. More precisely:

1. Charlie generates entangled pairs in the state $|AB\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends one particle to Alice and the other to Bob.
2. Alice has a number of bits she wishes to send to Bob. For each pair, she encodes her two bits b_1 and b_2 in the relative sign and phase of her qubit by conditionally applying σ_z and σ_x , respectively. She then sends the modified qubit to Bob.
3. Bob disentangles the X and Z components of the qubit by applying CNOT and H to the qubits he received from Alice and from Charlie. He then measures each qubit to obtain b_1 and b_2 , respectively.

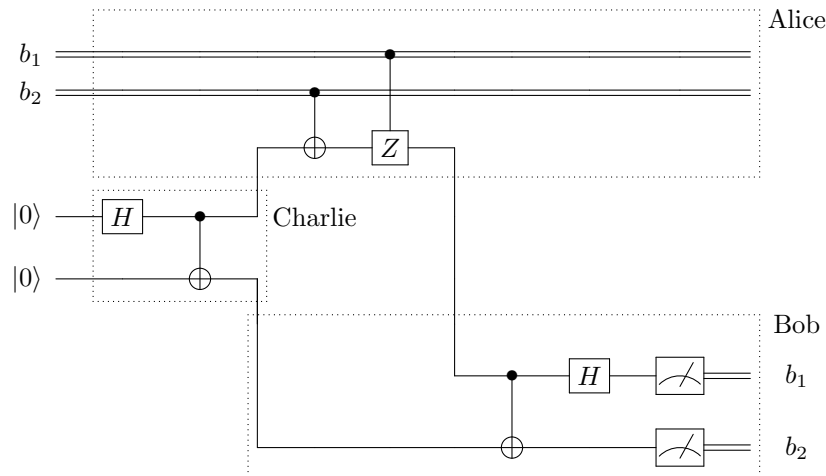


Figure E.7: Circuit diagram for the three-party quantum superdense coding experiment described above.

For this demonstration, Alice will send data to Bob in the form of a serialized bitstream representing an image. We'll use the built-in timing functionality to track the simulated time for each agent to complete their part of the protocol. Since superdense coding could be used as a networking protocol in the foreseeable future, even very rudimentary simulated timing data could be useful to quantify

the performance of the algorithm, especially if data validation and error correction through multiple transmission attempts is simulated. We assume a photon pulse interval of 1ns and a spatial separation between Alice and Bob of 1km, with Charlie at the midpoint. All agents are connected with the `FiberOpticQChannel` model, which simulates $0.16 \frac{\text{dB}}{\text{km}}$ attenuation errors by randomly changing transmitted Qubits to None.

```
import matplotlib.image as image

class Charlie(Agent):
    '''Charlie sends Bell pairs to Alice and Bob.'''
    def run(self):
        for qsys in self.qstream:
            a, b = qsys.qubits
            H(a)
            CNOT(a, b)
            self.qsend(alice, a)
            self.qsend(bob, b)
            self.output({"t": self.time})

class Alice(Agent):
    '''Alice sends Bob superdense-encoded bitstream'''
    def run(self):
        for _ in self.qstream:
            bit1 = self.data.pop(0)
            bit2 = self.data.pop(0)
            q = self.qrecv(charlie)
            # qubit could be lost to attenuation
            if q is not None:
                if bit2 == 1: X(q)
                if bit1 == 1: Z(q)
            self.qsend(bob, q)
            self.output({"t": self.time})

class Bob(Agent):
    '''Bob reconstructs Alice's data '''
    def run(self):
        bits = []
        for _ in self.qstream:
            a = self.qrecv(alice)
            c = self.qrecv(charlie)
```



```

        if a is not None and c is not None:
            CNOT(a, c)
            H(a)
            bits.extend([a.measure(), c.measure()])
        else: # if qubits are lost
            bits.extend([0,0])
    self.output({"t": self.time, "bits": bits})

# Load an image and serialize it to a bitstream
img = image.imread("../docs/source/img/foundryLogo.bmp")
bitstream = list(np.unpackbits(img))

# Initialize QStream and Agents
qstream = QStream(2, int(len(bitstream) / 2))
out = Agent.shared_output()
alice = Alice(qstream, out, data = bitstream)
bob = Bob(qstream, out)
charlie = Charlie(qstream, out)
# Set 1ns photon transmission rate
alice.pulse_length = 1e-9
bob.pulse_length = 1e-9
charlie.pulse_length = 1e-9
# Connect the agents with simulated fiber optic lines
alice.qconnect(bob, FiberOpticQChannel, length=1.0)
charlie.qconnect(alice, FiberOpticQChannel, length=0.5)
charlie.qconnect(bob, FiberOpticQChannel, length=0.5)

Simulation(alice, bob, charlie).run()
# Print simulated time at end
print("Alice:   {:.2e}s".format(out["Alice"]["t"]))
print("Bob:     {:.2e}s".format(out["Bob"]["t"]))
print("Charlie: {:.2e}s".format(out["Charlie"]["t"]))
# >> Alice:   4.16e-04s
# >> Bob:     4.20e-04s
# >> Charlie: 4.15e-04s

received_image = np.reshape(np.packbits(out["Bob"]["bits"]), img.shape)

```

Program E.6: Simulation of transmitting a serialized bitstream of an image via superdense coding.



Figure E.8: Results of Program E.6, showing Alice’s original image (left), and Bob’s received image (right). Bob applies no correction for attenuation errors, replacing dropped qubits with bit pairs of (0, 0).

E.3.3 Man-in-the-middle attack

In this demo, we show how quantum networks can be resistant to interception (“man-in-the-middle”) attacks by using a modified version of the superdense coding demonstration presented in Program E.6. As in the previous demo, Charlie will distribute Bell pairs to Alice and Bob, and Alice will attempt to send a classical message to Bob. However, a fourth party, Eve, will try to naively intercept the message Alice sends to Bob. Eve will measure each qubit from Alice, record the result, and re-transmit the qubit to Bob. This scenario is illustrated in the circuit diagram shown in Figure E.9.

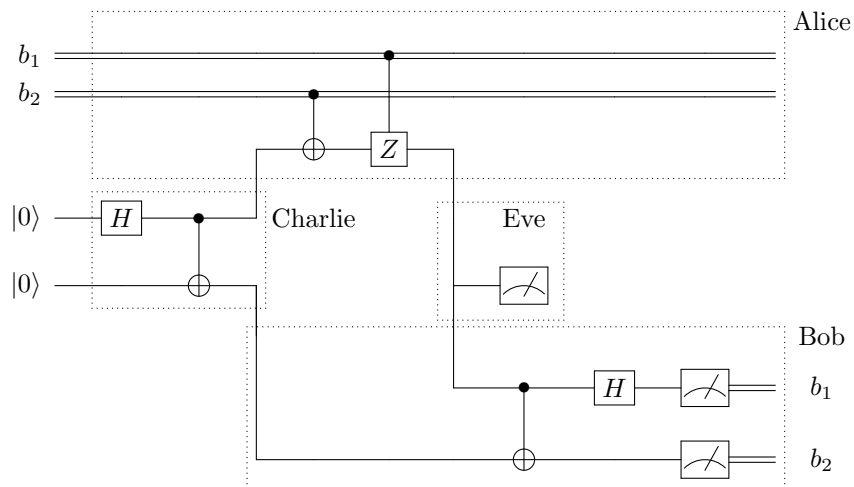


Figure E.9: Circuit diagram demonstrating a naive interception attack on the superdense networking protocol demonstrated in Figure E.7. Eve’s meter represents measurement of a qubit and transmission of a new qubit in the observed state.

This four-party quantum network scenario is simulated below in Program E.7.

```
import matplotlib.image as image

class Charlie(Agent):
    '''Charlie sends Bell pairs to Alice and Bob'''
```

```

def run(self):
    for qsys in self.qstream:
        a, b = qsys.qubits
        H(a)
        CNOT(a, b)
        self.qsend(alice, a)
        self.qsend(bob, b)

class Alice(Agent):
    '''Alice tries to send a bitstream to Bob'''
    def run(self):
        for _ in self.qstream:
            bit1 = self.data.pop(0)
            bit2 = self.data.pop(0)
            q = self.qrecv(charlie)
            if q is not None:
                if bit2 == 1: X(q)
                if bit1 == 1: Z(q)
            # Alice unknowingly sends the qubit to Eve
            self.qsend(eve, q)

class Bob(Agent):
    '''Bob receives Eve's intercepted data'''
    def run(self):
        bits = []
        for _ in self.qstream:
            a = self.qrecv(eve)
            c = self.qrecv(charlie)
            if a is not None and c is not None:
                CNOT(a, c)
                H(a)
                bits.extend([a.measure(), c.measure()])
            else:
                bits.extend([0,0])
        self.output(bits)

class Eve(Agent):
    '''Eve naively tries to intercept Alice's message'''
    def run(self):
        bits = []

```

```

    for _ in self.qstream:
        a = self.qrecv(alice)
        if a is not None:
            bits.append(a.measure())
        else:
            bits.append(0)
        self.qsend(bob, a)
    self.output(bits)

# Load an image and serialize it to a bitstream
img = image.imread("../docs/source/img/foundryLogo.bmp")
bitstream = list(np.unpackbits(img))

# Prepare QStream and Agent instances
qstream = QStream(2, int(len(bitstream) / 2))
out = Agent.shared_output()
alice = Alice(qstream, out, data = bitstream)
bob = Bob(qstream, out)
charlie = Charlie(qstream, out)
eve = Eve(qstream, out)
alice.qconnect(bob)
alice.qconnect(eve)
alice.qconnect(charlie)
bob.qconnect(charlie)
bob.qconnect(eve)
Simulation(alice, eve, bob, charlie).run()

```

Program E.7: A four-party interception attack scenario simulated using SQUANCH. This protocol is the same as Program E.6 except that Eve intercepts Alice’s transmitted qubits, measures them, and re-transmits them to Bob.



Figure E.10: The results of the simulated interception attack simulated by Program E.7. Since Eve only has access to half of the entangled $|AB\rangle$ state, she recovers only random noise, and Bob’s image is half-corrupted, alerting him to the presence of an eavesdropper.

E.3.4 Quantum error correction

When qubits are transmitted over quantum channels, they are subject to a complex set of errors which can cause them to decohere, depolarize, or be lost to the environment. For quantum information transfer to be feasible, the information must be encoded in a error-resistant format using any of a variety of quantum error correction models. In this demonstration, we show how to use SQUANCH’s channel and error modules to simulate quantum errors in a transmitted message, which we correct for using the Shor Code, shown as a circuit diagram in Figure E.11. This error correction model encodes a single logical qubit into the product of 9 physical qubits and is capable of correcting for arbitrary single-qubit errors.

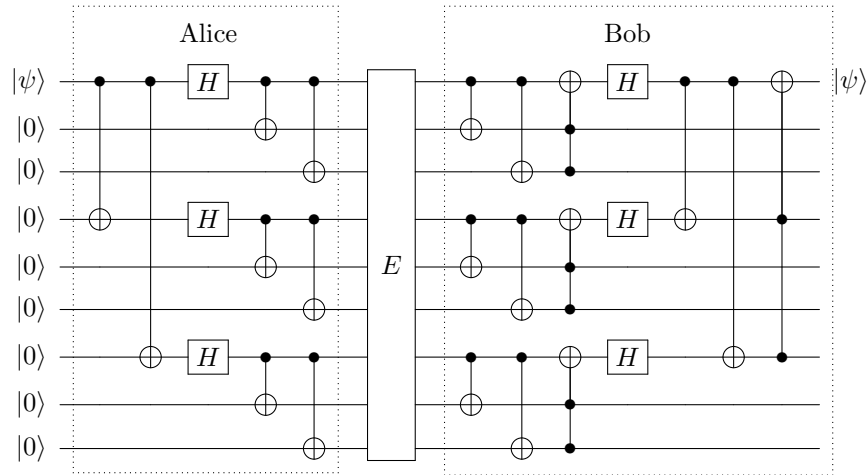


Figure E.11: Circuit diagram of encoding and decoding qubits using the Shor code. E represents a quantum channel with an error model which can corrupt a single physical qubit by applying a random unitary operator.

In this demo, we have two pairs of agents: Alice and Bob will communicate a message which is error-protected using the Shor code, and DumbAlice and DumbBob will transmit the message without error correction. Formally, for each state $|\psi\rangle$ to be transmitted through the channel, the following procedure is simulated:

1. Alice has some state $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$, which she wants to send to Bob through a noisy quantum channel. She encodes her single-qubit state in nine logical qubit as $|\psi\rangle \mapsto \alpha_0 \otimes_{j=1}^3 \frac{1}{\sqrt{2}} (|000\rangle + |111\rangle) + \alpha_1 \otimes_{k=1}^3 \frac{1}{\sqrt{2}} (|000\rangle - |111\rangle)$ using the circuit diagram in Figure E.7.
2. DumbAlice wants to send the same state, but she doesn’t error-protect the state and transmits the unencoded state $|\psi\rangle \otimes |00\dots 0\rangle$.
3. Alice and DumbAlice send their qubits through the quantum channel E to Bob and DumbBob,

respectively. The channel may apply an arbitrary unitary operation to a single physical qubit in each group of nine.

4. Bob receives Alice's qubits and decodes them using the Shor decoding circuit shown in Figure E.11.
5. DumbBob expects $|\psi\rangle \otimes |00\dots 0\rangle$ from DumbAlice and only measures the results of the the first qubit in each group of nine.

Transmitting an image is unsuitable for this scenario due to the larger size of the Hilbert space involved compared to the previous two demonstrations. (Each `QSystem.state` for $N = 9$ uses 2097264 bytes, compared to 240 bytes for $N = 2$.) Instead, Alice and DumbAlice will transmit the bitwise representation of a short message encoded as σ_z -eigenstates, and Bob and DumbBob will attempt to re-assemble the message.

Since this demonstration is fairly long, we'll split the code into two parts. First, we define the Shor encoding and decoding circuits as ordinary Python functions:

```
def shor_encode(qsys):
    # psi is state to send, q1...q8 are ancillas
    psi, q1, q2, q3, q4, q5, q6, q7, q8 = qsys.qubits
    CNOT(psi, q3)
    CNOT(psi, q6)
    H(psi)
    H(q3)
    H(q6)
    CNOT(psi, q1)
    CNOT(psi, q2)
    CNOT(q3, q4)
    CNOT(q3, q5)
    CNOT(q6, q7)
    CNOT(q6, q8)
    return psi, q1, q2, q3, q4, q5, q6, q7, q8

def shor_decode(psi, q1, q2, q3, q4, q5, q6, q7, q8):
    CNOT(psi, q1)
    CNOT(psi, q2)
    TOFFOLI(q2, q1, psi)
    CNOT(q3, q4)
    CNOT(q3, q5)
    TOFFOLI(q5, q4, q3)
    CNOT(q6, q7)
```

```

CNOT(q6, q8)
TOFFOLI(q7, q8, q6)
H(psi)
H(q3)
H(q6)
CNOT(psi, q3)
CNOT(psi, q6)
TOFFOLI(q6, q3, psi)
return psi # psi is now the original state

```

Program E.8: Implementation of the nine-qubit Shor code in SQUANCH.

Assuming that the functions defined in Program E.8 are imported, we can define the agent logic and quantum channel model in Program E.9.

```

import copy
from scipy.stats import unitary_group

# Agent runtime logic
class Alice(Agent):
    '''Alice sends an error-protected state to Bob'''
    def run(self):
        for qsys in self.qstream:
            # send the encoded qubits to Bob
            for qubit in shor_encode(qsys):
                self.qsend(bob, qubit)

class DumbAlice(Agent):
    '''DumbAlice sends unencoded state to DumbBob'''
    def run(self):
        for qsys in self.qstream:
            for qubit in qsys.qubits:
                self.qsend(dumb_bob, qubit)

class Bob(Agent):
    '''Bob receives and error-corrects Alice's state'''
    def run(self):
        measurements = []
        for _ in self.qstream:
            # Bob receives 9 logical qubits from Alice
            received = [self.qrecv(alice) for _ in range(9)]

```

```

        # Decode and measure the original state
        psi_true = shor_decode(*received)
        measurements.append(psi_true.measure())
    self.output(measurements)

class DumbBob(Agent):
    '''DumbBob gets DumbAlice's non-corrected state'''
    def run(self):
        measurements = []
        for _ in self.qstream:
            received = [self.qrecv(dumb_alice) for _ in range(9)]
            psi_true = received[0]
            measurements.append(psi_true.measure())
        self.output(measurements)

# Define the quantum error model
class ShorError(QError):
    '''Corrupt a random qubit with a random operator'''
    def __init__(self, qchannel):
        QError.__init__(self, qchannel)
        self.count = 0
        self.error_applied = False

    def apply(self, qubit):
        if self.count == 0: # corrupt 1/9 qubits max
            self.error_applied = False
        self.count = (self.count + 1) % 9
        if not self.error_applied and qubit is not None:
            if np.random.rand() < 0.5:
                # apply random U(2) operation to qubit
                random_unitary = unitary_group.rvs(2)
                qubit.apply(random_unitary)
                self.error_applied = True
        return qubit

# Define quantum channel with the error
class ShorQChannel(QChannel):
    def __init__(self, from_agent, to_agent):
        QChannel.__init__(self, from_agent, to_agent)
        self.errors = [ShorError(self)]

```



```

# Helper function code omitted, see Appendix A
def to_bits(string):
    '''Convert a string to a list of bits'''
def from_bits(bits):
    '''Convert a list of bits to a string'''

# Prepare a message and encode as spin eigenstates
msg = "Peter Shor once lived in Ruddock 238! But who was Airman?"
bits = to_bits(msg)
qstream = QStream(9, len(bits))
for bit, qsystem in zip(bits, qstream):
    if bit == 1:
        X(qsystem.qubit(0))

# Alice and Bob will use error correction
out = Agent.shared_output()
alice = Alice(qstream, out)
bob = Bob(qstream, out)
alice.qconnect(bob, ShorQChannel)

# Dumb agents won't use error correction
qstream2 = copy.deepcopy(qstream)
dumb_alice = DumbAlice(qstream2, out)
dumb_bob = DumbBob(qstream2, out)
dumb_alice.qconnect(dumb_bob, ShorQChannel)

# Run everything and record results
Simulation(dumb_alice, dumb_bob, alice, bob).run()

dumb_bob_msg = from_bits(out["DumbBob"])
bob_msg = from_bits(out["Bob"])
print("DumbAlice sent: {}".format(msg))
print("DumbBob received: {}".format(dumb_bob_msg))
print("Alice sent: {}".format(msg))
print("Bob received: {}".format(bob_msg))

```

Program E.9: Demonstration of quantum error correction using the Shor code.

The results of Program E.9 are shown in Figure E.12. (A screenshot is provided as Unicode characters are problematic to include in L^AT_EX documents.)

```

DumbAlice ██████████ 100% 456/456 [03:55<00:00, 1.94it/s]
DumbBob ██████████ 100% 456/456 [03:55<00:00, 1.94it/s]
Alice ██████████ 100% 456/456 [03:55<00:00, 1.94it/s]
Bob ██████████ 100% 456/456 [03:55<00:00, 1.94it/s]
DumbAlice sent: Peter Shor once lived in Ruddock 238! But who was Airman?
DumbBob received: øu<ar*W/>P0mn*3âyx5! ig3&A&ApÛî (r38e0V~Pvik\dab``á|mcj?
Alice sent: Peter Shor once lived in Ruddock 238! But who was Airman?
Bob received: Peter Shor once lived in Ruddock 238! But who was Airman?

```

Figure E.12: A screenshot of the output of Program E.9 running in a Jupyter notebook. The original message can be recovered if a quantum error correcting protocol is used. Progress bars are automatically updated from `Agent.qstream.__iter__` when running programs with `Simulation(...).run()` and are shown in notebook or terminal environments.

E.4 Conclusions

In this appendix we have introduced SQUANCH, an open-source Python framework for creating distributed and performant simulations of multi-party quantum networks. The framework includes modules for representing quantum information at the single-particle, multi-particle, and ensemble levels, and large number of quantum gates for manipulating quantum information, enumerated in full in Section E.5.3. The higher-abstraction modules in the framework provide software infrastructure for constructing the nodes of a quantum network by programming the actions of a quantum-mechanical “agent” and for connecting the network with noisy information channels. Virtually all components of the framework are configurable or extensible, allowing users to program simulations ranging from simple demonstrations to complex and detailed computational models. We hope that the development of this framework will stimulate exploration in the exciting field of quantum networking algorithms.

E.5 Appendix

E.5.1 Source code for simulator and demonstrations

The full source code for the SQUANCH framework is available on GitHub at github.com/att-innovate/squanch or from the Python Package Index at pypi.org/project/SQUANCH. Documentation is available on the documentation website at att-innovate.github.io/squanch. The full source code for each of the demonstrations in Section E.3 is available in the `/demos` directory of the GitHub repository.

E.5.2 Formal definition of a quantum agent

Our definition for a quantum agent is chosen to mirror the structure of the `squanch.Agent` class, but is essentially a networked quantum Turing machine. A quantum agent is represented by a 5-tuple

(C, M_C, M_Q, G, T) :

- $C = (I_C, I_Q, O_C, O_Q)$ is a configuration of channels connecting the agent to other agents. I_Q and O_Q are input and output quantum channels, through which the agent can receive and send qubits, and I_C and O_C are input and output classical channels¹⁴.
- M_C is the classical memory, where the k th bit is indexed as $M_C[k]$.
- M_Q is the quantum memory, which can store an arbitrary number of qubits, where the k th qubit is $M_Q[k]$. Qubits received through I_Q can be stored in M_Q until they are sent through O_Q , and the measurement result of a qubit can be stored in M_C .
- G is an arbitrary but finite¹⁵ set of quantum operations. In addition, we always include “operators” for measurement and for sending qubits to a connected agent.
- A classical Turing machine¹⁶ $T = (Q, \Sigma, \delta)$ where:
 - Q is a finite set of classical states
 - Σ is a finite tape alphabet with a blank symbol
 - $\delta : Q \times \Sigma \rightarrow \Sigma \times Q \times \{L, R\} \times \{L_Q, R_Q\} \times G$ is a modified transition function which also includes the traversal of quantum memory with $\{L_Q, R_Q\}$ and the ability to perform an operation $g \in G$ on the current qubit.

In the non-networked case that $C = (\emptyset, \emptyset, \emptyset, \emptyset)$, the Agent has full access to the state space of their quantum system, and this model is seemingly reducible to the definition of a quantum Turing machine given in [30], although a formal proof of this is beyond the scope of this paper.

E.5.3 List of gates

A complete list of gates available in **SQUANCH v1.1.0** is given below in Table E.1.

¹⁴Most definitions of quantum channels include the ability to transmit classical information; we make the explicit distinction between quantum and classical channels here for better consistency with the structure of the `squanch.Agent` class.

¹⁵Some gates, such as the **PHASE** gate, take parameters, which at first glance seems to make it problematic for the transition function δ to be able to access them from a finite set. However, assuming G contains a functionally complete set of quantum gates, compositions of $g \in G$ can be used to approximate any unitary operator to within arbitrary accuracy [18].

¹⁶This is not technically a Turing machine from a computability standpoint, since T has access to a random oracle by way of the measurement operation.





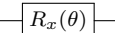
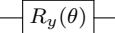
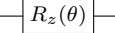







Operation	Name	Definition	Symbol
H(qubit)	Hadamard	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	
X(qubit)	Pauli-X	$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
Y(qubit)	Pauli-Y	$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	
Z(qubit)	Pauli-Z	$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	
RX(qubit, angle)	Rotation-X	$R_x(\theta) = \cos \frac{\theta}{2} \mathbb{1} - i \sin \frac{\theta}{2} \sigma_x$	
RY(qubit, angle)	Rotation-Y	$R_y(\theta) = \cos \frac{\theta}{2} \mathbb{1} - i \sin \frac{\theta}{2} \sigma_y$	
RZ(qubit, angle)	Rotation-Z	$R_z(\theta) = \cos \frac{\theta}{2} \mathbb{1} - i \sin \frac{\theta}{2} \sigma_z$	
PHASE(qubit, angle)	Phase shift	$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$	
CNOT(ctrl, targ)	Controlled-NOT	$\text{CNOT}_{jk} = 0\rangle\langle 0 _j \otimes \mathbb{1}_k + 1\rangle\langle 1 _j \otimes \sigma_{x;k}$	
CPHASE(ctrl, targ, angle)	Controlled-phase	$\text{CPHASE}_{jk}(\theta) = 0\rangle\langle 0 _j \otimes \mathbb{1}_k + 1\rangle\langle 1 _j \otimes \sigma_{x;k}$	
CU(ctrl, targ, unitary)	Controlled-unitary	$\text{C}_j(U)_k = (0\rangle\langle 0)_j \otimes \mathbb{1}_k + (1\rangle\langle 1)_j \otimes U_k$	
SWAP(qubit1, qubit2)	SWAP gate	$\text{SWAP}_{jk} = \text{CNOT}_{kj} \text{CNOT}_{jk} \text{CNOT}_{kj}$	
TOFOLLI(ctrl1, ctrl2, targ)	Toffoli gate	$\text{CCNOT}_{ijk} = 0\rangle\langle 0 _i 0\rangle\langle 0 _j \otimes \mathbb{1}_k + 0\rangle\langle 0 _i 1\rangle\langle 1 _j \otimes \mathbb{1}_k + 1\rangle\langle 1 _i 0\rangle\langle 0 _j \otimes \mathbb{1}_k + 1\rangle\langle 1 _i 1\rangle\langle 1 _j \otimes \sigma_{x;k}$	
qubit.measure()	Measurement	$\hat{\rho} \mapsto \hat{\rho}_i = \frac{ i\rangle\langle i \hat{\rho} i\rangle\langle i }{\text{tr}[i\rangle\langle i \hat{\rho}]}$, return bit i	

Table E.1: A list of the built-in quantum operations included in SQUANCH as of version 1.1.0. All gates take at least one `Qubit` argument and return nothing, modifying the state of the associated `QSystem` in-place.

Appendix F

Computing photon scattering in open quantum systems

This Appendix demonstrates a module I wrote for numerically computing photon scattering in arbitrary driven systems coupled to some configuration of output waveguides using QuTiP [107]. I have used this functionality for various calculations included in the main body of this thesis. This Appendix is adapted from a Jupyter notebook tutorial available on the QuTiP website. We first briefly review the generalized problem of photon scattering in quantum optical systems discussed in Ref. [70].

F.1 Problem definition

Consider an arbitrary system with a Hamiltonian $H_S(t)$ coupled to a bath of waveguide modes, as shown in Figure F.1.

The problem we address here is this: if we drive the system with some excitation field, such as a laser pulse, how do photons scatter from the system into the waveguide?

The system Hamiltonians we will consider take the form

$$H_S(t) = \begin{cases} H_{0S} + H_{1S}(t) & \text{if } 0 < t < T_P \\ H_{0S} & \text{otherwise,} \end{cases} \quad (\text{F.1})$$

where T_P is the pulse duration (if well-defined). The waveguide Hamiltonians can be described as

$$H_{0B} = \int_{-\infty}^{\infty} d\omega \omega b_{\omega}^{\dagger} b_{\omega}, \quad (\text{F.2})$$

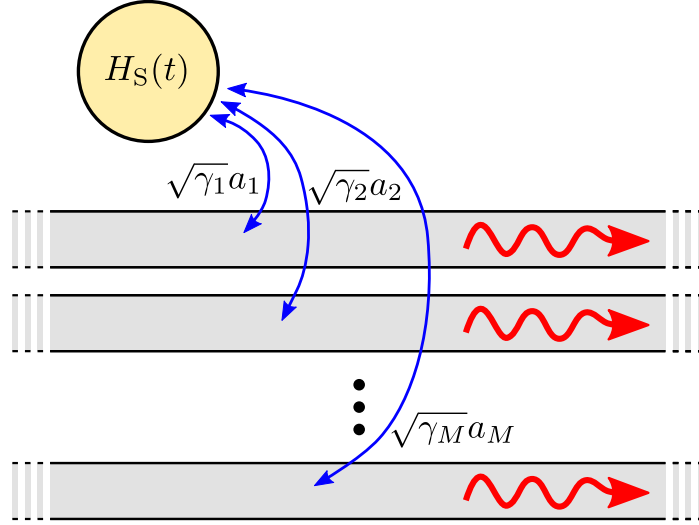


Figure F.1: The general problem is to compute the field scattered into unidirectional waveguide(s) from an energy-nonconserving system Hamiltonian with a unique ground state. This class of Hamiltonian is often used to represent coherent laser pulses scattering off quantum-optical systems such as a two-level system, Jaynes-Cummings system, or entangled photon pair source.

which can be rewritten in a temporal basis (roughly speaking, indexed by emission time) by Fourier transforming the operators b_ω :

$$b_\tau \equiv \int_{-\infty}^{\infty} \frac{d\omega}{\sqrt{2\pi}} e^{-i\omega\tau} b_\omega, \quad |\vec{\tau}^{(m)}\rangle \equiv b_{\tau_1}^\dagger \cdots b_{\tau_m}^\dagger |0_B\rangle \quad (\text{F.3})$$

The total Schrodinger-picture Hamiltonian can be written as a sum of system, bath, and coupling terms $H(t) = H_S(t) + V + H_{0B}$, and can be transformed into the interaction picture:

$$H_I(t) = H_S(t) + e^{iH_{0B}t} V e^{-iH_{0B}t}. \quad (\text{F.4})$$

To solve the dynamics of this system, we could integrate the Schrodinger equation:

$$i \frac{\partial}{\partial t} |\Psi_I(t)\rangle = H_I(t) |\Psi_I(t)\rangle. \quad (\text{F.5})$$

However, practically integrating this equation is not feasible, so we instead “coarse-grain” the temporal dynamics to Δt and take a continuous limit as $\Delta t \rightarrow 0$.

F.2 Coarse-grained dynamics and the scattering operator

If we define an “effective Hamiltonian” $H_{\text{eff}}(t) = H_S(t) - i\frac{\gamma}{2}a^\dagger a$, we can generate an effective propagator mapping the system from the k^{th} to the $k + 1^{\text{th}}$ time bin which is correct up to $\mathcal{O}(\Delta t)$:

$$U_{\text{eff}}[k + 1, k] \equiv \langle 0_k | U[k + 1, k] | 0_k \rangle \approx \exp \left[-i \int_{k\Delta t}^{(k+1)\Delta t} dt H_{\text{eff}}(t) \right]. \quad (\text{F.6})$$

From this, we can derive the scattering operator for the system into the system of waveguides (see the paper for more detail). For scattering of N photons into single waveguide, this operator $\langle \hat{\Omega}_-^\dagger \rangle_{\vec{\tau}^{(m)}}$ takes the form:

$$\langle \hat{\Omega}_-^\dagger \rangle_{\vec{\tau}^{(m)}} = \langle 0_S | U_{\text{eff}}(\tau_{\text{max}}, \tau_m) \prod_{q=N}^1 \sqrt{\gamma} a U_{\text{eff}}(\tau_q, \tau_{q-1}) | \psi_S(0) \rangle, \quad (\text{F.7})$$

with $\tau_0 = 0$, $\tau_{\text{max}} = \max(T_p, \tau_m)$. The multi-waveguide case will be discussed later in this Appendix.

F.3 The temporal basis

For a system coupled to W waveguides emitting N photons approximated by coarse-grained dynamics with T time bins, the temporal basis described in Fischer, et al. (Eq. 138 and 153, with slight notation changes) can be thought of as a system of T qubits for each of the W waveguides with a total of N creation operators applied to $|0\rangle$:

$$|\vec{\tau}_{(W)}^{(N)}\rangle = |\vec{\tau}_1^{(w_1)}, \vec{\tau}_2^{(w_2)}, \dots, \vec{\tau}_W^{(w_W)}\rangle = \prod_{i=1}^N b_{w_i, \tau_i}^\dagger b_{w_i, \tau_i}^\dagger \dots b_{i, \tau_{n_i}}^\dagger |0\rangle, \quad (\text{F.8})$$

where w_k denotes scattering into the k^{th} waveguide and n_i denotes the maximum number of photons scattered into some waveguide. Although this basis is exact, it has an intractable space complexity of $\mathcal{O}(2^{T \cdot M})$, making unsuitable for simulation work.

The temporal basis we use in the `qutip.scattering` module is more closely modeled after ladder operators and explicitly restricts the basis to N emissions. To generate the basis, we make M copies of the T time bins. Emission of a photon at the i^{th} time bin into the w^{th} waveguide is represented by an entry in the $(wT + i)^{\text{th}}$ index of a (WT) -dimensional vector, so the overall temporal basis is given by:

$$|\vec{\tau}_{(W)}^{(N)}\rangle = |\vec{\tau}_1^{(w_1)}, \vec{\tau}_2^{(w_2)}, \dots, \vec{\tau}_W^{(w_W)}\rangle = \bigotimes_{n=1}^N |\tau_{n, w_n}\rangle = \bigotimes_{n=1}^N \vec{\mathcal{T}}[w_n T + \tau_n], \quad (\text{F.9})$$

where τ_{n, w_n} denotes emission into the w_n^{th} waveguide of the n^{th} photon and $\vec{\mathcal{T}}[w_n T + \tau_n]$ denotes the basis vector corresponding to τ_{n, w_n} , namely the $(w_n T + \tau_n)$ -th index. The creation operators in

the original temporal basis are mapped to $(w_i T + \tau_n)$ applications of the “temporal ladder operator”:

$$b_{w_i, \tau_n}^\dagger = \frac{(a^\dagger)^{w_i T + \tau_n}}{\sqrt{(w_i T + \tau_n)!}} \quad (\text{F.10})$$

This gives this basis a space complexity of $\mathcal{O}((WT)^N)$, which is more manageable given that for most applications $T \gg W, N$.

F.4 Single waveguide: driven quantum two-level system

To demonstrate the `qutip.scattering` module, we’ll start with the simplest case of a two-level quantum system coupled to a single output waveguide. The system has initial state $|\psi_0\rangle = |e\rangle_{\text{sys}} \otimes |vac\rangle_{\text{wg}}$ with a bare Hamiltonian of $H_{0S} = \omega_0 \sigma^\dagger \sigma$. Adding an effective non-Hermitian term to govern the evolution of the system under spontaneous emission, $H_{\text{eff}} = H_{0S} - i\frac{\gamma}{2} \sigma^\dagger \sigma$. When the system is driven by a coherent pulse, it undergoes Rabi oscillations. Picking a square pulse to give a simple Hamiltonian, the overall effective Hamiltonian is $H_{\text{eff}}(t) = H_{0S} - H_{1S}(t) - i\frac{\gamma}{2} \sigma^\dagger \sigma$, where

$$H_{1S}(t) = \begin{cases} \Omega (ie^{-i\omega_0 t} \sigma^\dagger - ie^{i\omega_0 t} \sigma) & \text{if } 0 < t < T_P \\ 0 & \text{otherwise.} \end{cases} \quad (\text{F.11})$$

We define the Hamiltonian and choose pulse parameters below.

```
# Pulse parameters
w0      = 10 * 2 * np.pi          # arbitrary laser frequency
gamma   = 1.0                     # arbitrary coupling constant

# Operators
sm      = np.sqrt(gamma) * destroy(2) # TLS coupled collapse operator
psi0    = basis(2,0)               # starting state |psi(0)> = |0>

def Htls(gamma, pulseLength, pulseArea):
    RabiFreq = pulseArea / (2*pulseLength)

    # Bare Hamiltonian for a TLS
    HOS = w0 * create(2) * destroy(2)

    # Define H_1S(t)
    H1S1 = lambda t, args: RabiFreq * 1j*np.exp(-1j*w0*t) * (t < pulseLength)
    H1S2 = lambda t, args: RabiFreq * -1j*np.exp(1j*w0*t) * (t < pulseLength)
```



```
# Put the Hamiltonian in QuTiP list-callback form
return [HOS - 1j/2 * sm.dag() * sm,
        [sm.dag(), H1S1],
        [sm, H1S2]]
```

F.5 Computing photon scattering amplitudes

Let's begin by computing the scattering amplitude of a single-photon emission as a function of time. For this, we can use the `temporal_scattered_state()` function in the `scattering` module, which computes:

$$\begin{aligned}
|\phi_n\rangle &= \int_0^\infty d\tau_1 \int_{\tau_1}^\infty d\tau_2 \cdots \int_{\tau_{n-1}}^\infty d\tau_n \langle 0_S, \{\tau_1, \tau_2, \dots, \tau_n\} | \psi(t \rightarrow \infty) \rangle | \tau_1, \tau_2, \dots, \tau_n \rangle \\
&= \int_{\vec{\tau}_n} d\vec{\tau}_n \langle 0_S, \{\vec{\tau}_n\} | \psi(t \rightarrow \infty) \rangle | \vec{\tau}_n \rangle \\
&= \int_{\vec{\tau}_n} d\vec{\tau}_n \langle \hat{\Omega}_-^\dagger \rangle_{\vec{\tau}_n} | \vec{\tau}_n \rangle \\
&= \hat{\Omega}_-^\dagger | \psi_0 \rangle.
\end{aligned} \tag{F.12}$$

This function takes as arguments the Hamiltonian or the effective Hamiltonian, the initial system state, the number of emissions, a list of collapse operators (one for each waveguide - see the following section for more detail), and a list of times. The list of times must exceed the duration of the pulse for the function to yield sensible results (or, if the pulse does not have a well-defined end, the times list must contain most of the temporal region of interest).

By passing the keyword argument `construct_effective_hamiltonian`, you can tell the function whether the Hamiltonian you provided is H or H_{eff} ; by default, the value is `True`, so an effective Hamiltonian will be constructed from the provided list of collapse operators as $H_{\text{eff}} = H - \frac{i}{2} \sum_n c_{\text{ops}}[n]$. The function iteratively calls `photon_scattering_operator()` and returns the temporal scattered state as a `Qobj`, so to extract the amplitudes a_t , we will need to project it onto the temporal basis:

$$a_t = \langle t | \phi_n \rangle = \langle t | \hat{\Omega}_-^\dagger | \psi_0 \rangle = \langle t | \int_{\vec{\tau}_n} d\vec{\tau}_n \langle \hat{\Omega}_-^\dagger \rangle_{\vec{\tau}_n} | \vec{\tau}_n \rangle, \tag{F.13}$$

which we can do using the `temporal_basis_vector()` function. This function takes a nested list of temporal emission indices for each waveguide and the total number of time bins. For the single-waveguide case, the nested list of time indices simply reduces to `[[indexOf(t)]]`. Computing the

scattering amplitudes using the same parameters as the ones used in the analytical results of Figure 5(b) in Fischer, et al., we obtain visually identical results:

```

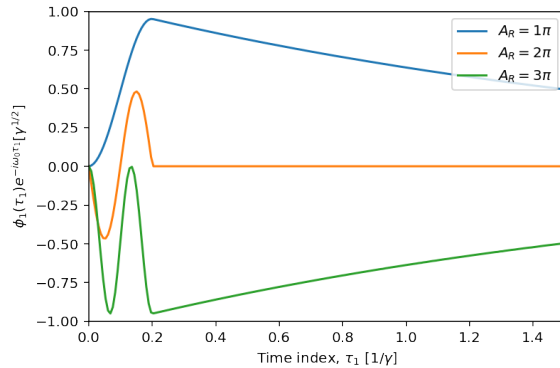
T = 200
tlist = np.linspace(0,1.5/gamma,T)
pulse_length = 0.2 / gamma
pulse_areas = [np.pi, 2*np.pi, 3*np.pi]

for pulse_area in pulse_areas:
    # Use construct_effective_hamiltonian=False since we are providing H_eff in this case
    scattered_state = temporal_scattered_state(HtIs(gamma, pulse_length, pulse_area),
        ↪ psi0, 1, [sm], tlist, construct_effective_hamiltonian = False)
    amplitudes = []
    for i in range(T):
        amplitudes.append((temporal_basis_vector([[i]], T).dag() *
            ↪ scattered_state).full().item())
    # Adjust amplitudes for time evolution
    amplitudes = np.real(np.array(amplitudes) * np.exp(1j * w0 * tlist))
    plt.plot(tlist, amplitudes, label = "$A_R = {} \pi$".format(round(pulse_area / np.pi)))

plt.ylim(-1,1)
plt.xlim(tlist[0],tlist[-1])
plt.xlabel('Time index, $\tau_1$ [1/\gamma$]')
plt.ylabel('$\phi_1(\tau_1) e^{-i \omega_0 \tau_1}$ [\gamma^{1/2}]$')
plt.legend(loc = 'upper right')
plt.show()

```

Output:



F.6 Total photon scattering probability

To calculate the total probability of emitting a certain number of photons, $P_n = \langle \phi_n | \phi_n \rangle$, we can expand in terms a complete set of temporal projection operators $\int_{\vec{\tau}_n} |\tau_n\rangle \langle \tau_n| d\tau_n$:

$$\begin{aligned}
 P_n &= \langle \phi_n | \phi_n \rangle \\
 &= \int_{\vec{\tau}_n} d\vec{\tau}_n \langle \phi_n | \vec{\tau}_n \rangle \langle \vec{\tau}_n | \phi_n \rangle \\
 &= \int_0^\infty d\tau_1 \int_{\tau_1}^\infty d\tau_2 \cdots \int_{\tau_{n-1}}^\infty d\tau_n \langle \phi_n | \tau_1, \tau_2, \dots, \tau_n \rangle \langle \tau_1, \tau_2, \dots, \tau_n | \phi_n \rangle
 \end{aligned}
 \tag{F.14}$$

More simply, however, you can use the `scattering_probability()` function, which recursively integrates the results of `temporal_scattered_state()` to return the total probability of N photons being scattered from the system over the specified list of times. Notably, the time list does not need to be linear - the integration routines will account for unevenly spaced time bins. This allows you to do things like provide logarithmically spaced times, which better captures regions closer to $t = 0$ where more interesting dynamics occur.

To make things faster, we'll remove the time dependence of H_{eff} with a rotating frame transformation. We'll also drop the $-\frac{i}{2}\sigma^\dagger\sigma$ term and the `construct_effective_hamiltonian = False` argument to allow `temporal_scattered_state()` to construct the effective Hamiltonian on its own.

Since `scattering_probability()` returns a pickleable result (a number), it is also very easily multiprocessed, so we'll take this opportunity to show how this can be done. (Note that this does make debugging untested code a more opaque process.) Computing the total scattering probabilities for $N = 0, 1, 2$ photons as a function of pulse area yields a similar result to Figure 5(a) in Fischer, et al:

```

def Htls_rft(gamma, pulseLength, pulseArea):
    RabiFreq = pulseArea / (2*pulseLength)
    return [[sm.dag() + sm, lambda t, args: RabiFreq * (t < pulseLength)]]

pulse_length = 0.2 / gamma
pulse_areas = np.linspace(0, 4*np.pi, 100)
tlist = np.geomspace(gamma, 7*gamma, 40) - gamma
emission_nums = [0, 1, 2]

def scattering_probability_multiprocess(pulse_area, n):
    # Helper function to allow pool.map parallelism
    return scattering_probability(Htls_rft(gamma, pulse_length, pulse_area), psi0, n,
    ↪ [sm], tlist)

```

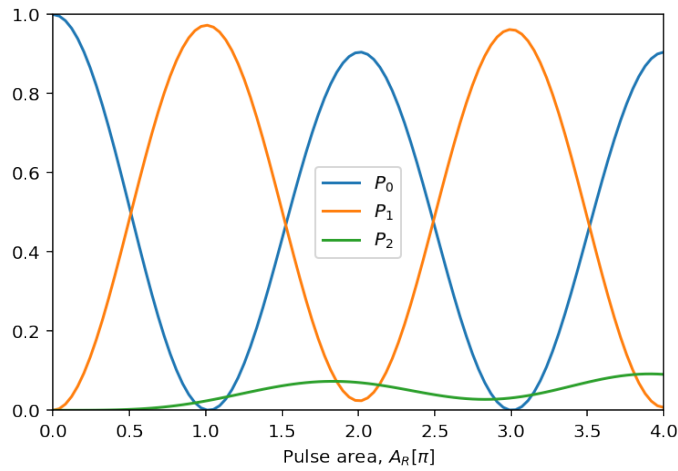
```

pool = Pool(worker_count)
for n in emission_nums:
    args = [(pulse_area, n) for pulse_area in pulse_areas]
    scatter_probs = pool.starmap(scattering_probability_multprocess, args)
    plt.plot(pulse_areas / np.pi, scatter_probs, label = "$P_{-{}-}$".format(n))
pool.close()

plt.ylim(0,1)
plt.xlim(pulse_areas[0]/np.pi, pulse_areas[-1]/np.pi)
plt.xlabel("Pulse area, $A_R [\pi]$")
plt.legend()
plt.show()

```

Output:



F.7 Computing second-order coherence in the scattered state

In experiments, the two-photon wavefunction is often characterized from the second-order coherence:

$$G^{(2)}(t_1, t_2) \approx \langle \phi_2 | b_0^\dagger(t_1) b_0^\dagger(t_2) b_0(t_2) b_0(t_1) | \phi_2 \rangle. \quad (\text{F.15})$$

Since the creation operators b_0^\dagger do not translate exactly into the basis used in `qutip.scattering`, this is not directly computable in this form, but we can still calculate $G^{(2)}$ with creative application of `temporal_basis_vector()`. The second-order coherence measures the correlations for photons to be

emitted at times t_1 and t_2 with corresponding time-bin indices i and j . To compute the coherence, we first compute the temporal scattered state, then project it onto `temporal_basis_vector([[i,j]], T)`, which gives the basis vector corresponding to photons emitted at time indices i and j (into the same - first - waveguide) out of T total time bins. This projection onto a (approximately) complete set of temporal basis vectors gives the second-order coherence, which is Figure 5(c) in Fischer, et al.:

```

T = 200
tlist = np.linspace(0,1.5/gamma,T)
pulse_area = 6*np.pi
pulse_length = 0.2 / gamma
correlations = np.zeros((T, T))

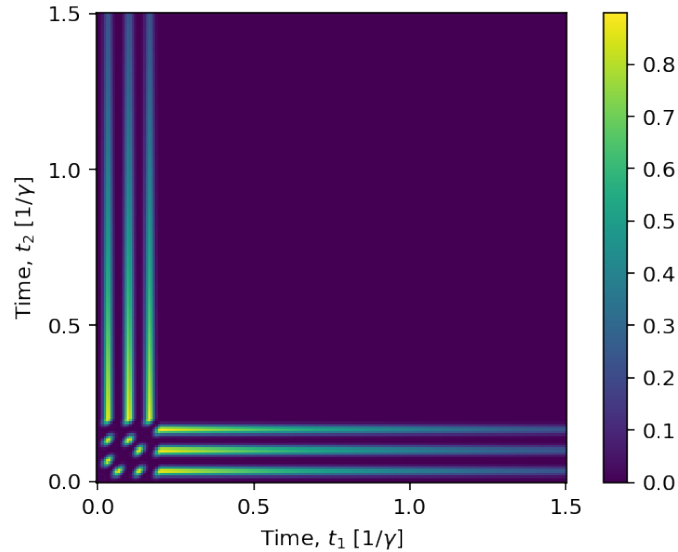
H = Htls_rft(gamma, pulse_length, pulse_area)
scattered_state = temporal_scattered_state(H, psi0, 2, [sm], tlist)

for i in range(T):
    for j in range(T):
        # temporal_scattered_state() computes only using ordered emission times, so to
        # get the full set of correlations, we need to use ordered temporal basis vector
        [a,b] = sorted([i,j])
        basis_vec = temporal_basis_vector([[a,b]], T)
        correlations[i,j] = np.abs((basis_vec.dag() * scattered_state).full().item())**2

fig, ax1 = plt.subplots(1,1)
cax = ax1.imshow(correlations, interpolation='nearest', origin='lower')
ax1.set_xticks(np.linspace(0,T-1,4))
ax1.set_xticklabels([0.0, 0.5, 1.0, 1.5])
ax1.set_xlabel("Time, $t_1$ [$1/\gamma$]")
ax1.set_yticks(np.linspace(0,T-1,4))
ax1.set_yticklabels([0.0, 0.5, 1.0, 1.5])
ax1.set_ylabel("Time, $t_2$ [$1/\gamma$]")
fig.colorbar(cax)
plt.show()

```

Output:



F.8 Pulse-wise second-order coherence

Experimentally accessing the temporal correlations given by $G^{(2)}$ or photocount distributions P_m can be quite challenging, so typically a quantity called the pulse-wise second-order coherence is used, defined as:

$$g^{(2)}[0] = \frac{\sum_m m(m-1)P_m}{(\sum_m mP_m)^2} \approx \frac{2P_2}{(P_1 + 2P_2)^2}. \quad (\text{F.16})$$

We can easily compute this with `scattering_probability`, obtaining similar results to Figure 5(d) in Fischer, et al.:

```

pulse_length = 0.2/gamma
pulse_areas = np.linspace(0.01,4*np.pi,150)
emission_nums = [1,2]
# you can use non-linearly space time bins with scattering_probability()
tlist = np.geomspace(gamma, 21*gamma, 40) - gamma

def scatter_prob(pulse_area, n):
    # Helper function to allow pool.map parallelism
    return scattering_probability(HtIs_rft(gamma, pulse_length, pulse_area), psi0, n,
    ↪ [sm], tlist)

pool = Pool(worker_count)

```

```

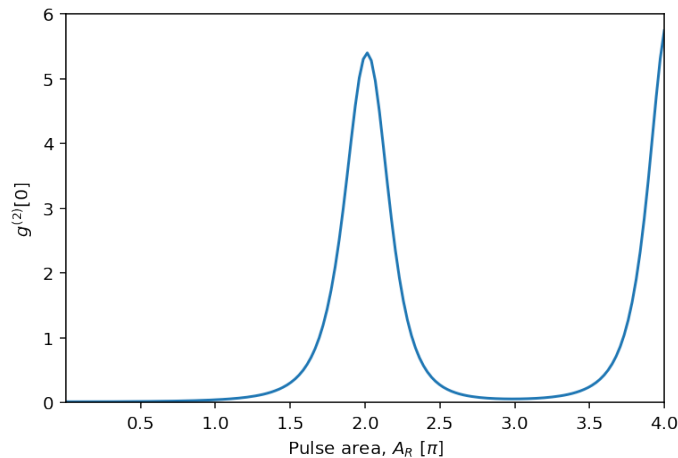
Pm = dict.fromkeys(emission_nums)
for n in emission_nums:
    args = [(pulse_area, n) for pulse_area in pulse_areas]
    Pm[n] = np.array(pool.starmap(scatter_prob, args))
pool.close()

# Calculate pulse-wise coherence
pulseWiseCoherence = np.sum([m * (m-1) * Pm[m] for m in Pm], axis=0) / \
    np.square(np.sum([m * Pm[m] for m in Pm], axis=0))

plt.plot(pulse_areas/np.pi, pulseWiseCoherence)
plt.ylim(0,6)
plt.xlim(pulse_areas[0]/np.pi, pulse_areas[-1]/np.pi)
plt.xlabel("Pulse area, $A_R$ $[\pi]$")
plt.ylabel("$g^{(2)}[0]$")
plt.show()

```

Output:



F.9 Multiple waveguides: spontaneous parametric downconversion

We'll now extend the problem to multiple waveguides by simulating the scattering dynamics of spontaneous parametric downconversion. The scattering amplitude discussed above extended to a

system with W waveguides is:

$$\left\langle \hat{\Omega}_-^\dagger \right\rangle_{\tilde{\tau}^{(N)}} \equiv \left\langle \hat{\Omega}_-^\dagger \right\rangle_{\tilde{\tau}_1^{(m_1)}, \tilde{\tau}_2^{(m_2)}, \dots, \tilde{\tau}_M^{(m_M)}} = \langle \mathbf{0}_S | U_{\text{eff}}(\tau_{\text{max}}, \tilde{\tau}_N) \prod_{q=N}^1 \sqrt{\gamma_{Q[q]}} a_{Q[q]} U_{\text{eff}}(\tilde{\tau}_q, \tilde{\tau}_{q-1}) | \psi_S(0) \rangle \quad (\text{F.17})$$

as a projection onto $|\tilde{\tau}_1^{(m_1)}, \tilde{\tau}_2^{(m_2)}, \dots, \tilde{\tau}_W^{(m_W)}\rangle$, where $N = m_1 + m_2 + \dots + m_W$ is the total number of photons scattered, $\tilde{\tau}^{(N)}$ is a chronologically sorted set of all time indices from the $\tilde{\tau}_i^{(m_i)}$'s, and $Q[q]$ is the index of the waveguide corresponding to the photon scattered at $\tilde{\tau}_q$. We present this equation without derivation; see Fischer, et al. for more details.

Consider a SPDC cavity with a Hamiltonian given by a sum of time-independent and -dependent parts $H = H_{0S} + H_{1S}$, with:

$$H_{0S} = \omega_1 a_1^\dagger a_1 + \omega_2 a_2^\dagger a_2, \quad (\text{F.18})$$

and

$$H_{1S} = g(t) \left(e^{i\omega_p t} a_1 a_2 + e^{-i\omega_p t} a_1^\dagger a_2^\dagger \right), \quad (\text{F.19})$$

where a_1 and a_2 annihilate photons at frequencies ω_1 and ω_2 , respectively, $\omega_p = \omega_1 + \omega_2$, and $g(t)$ is a function depending on the amplitude of the pump beam and the nonlinear susceptibility of the cavity. As a specific example, let's consider driving the system with a Gaussian pulse, such that $g(t) = g_0 \exp\left(-\frac{(t-t_0)^2}{2\tau^2}\right)$. Truncating the cavity excitation capacity to $n = 6$, we define the Hamiltonian for the system, again using a rotating frame transformation as before:

$$H_{\text{SPDC}} = \left(a_1^\dagger a_2^\dagger + a_1 a_2 \right) g(t) + H_{\text{eff}} \text{ terms}, \quad (\text{F.20})$$

where we allow the functions to construct the effective Hamiltonian by adding the $-\frac{i}{2} \sum_n \text{c.ops}[n]$ terms.

```

Ncav = 6 # cavity excitation capacity
a1 = tensor(destroy(Ncav), qeye(Ncav)) # left cavity annihilator
a2 = tensor(qeye(Ncav), destroy(Ncav)) # right cavity annihilator
cavity_vac = tensor(basis(Ncav, 0), basis(Ncav, 0)) # vacuum state
w1 = w2 = 1/gamma # cavity frequencies
wp = w1 + w2 # pump frequency
spdc_c_ops = [np.sqrt(gamma)*a1, np.sqrt(gamma)*a2] # cavity collapse operators

# Gaussian laser pulse
def g(t, t0, g0, tau):
    return g0 * np.exp(-1 * (t-t0)**2 / (2 * tau**2))
    
```

```
# SPDC Hamiltonian with rotating frame transformation applied
def Hspdc(t0, g0, tau):
    return [[a1.dag() * a2.dag() + a1 * a2, lambda t, args: g(t, t0, g0, tau)]]
```

F.10 Two-photon scattering amplitudes

Here we compute the amplitude for the two-photon part of the output state projected onto the temporal basis. We plot only the case where one photon is scattered into the first waveguide and the other into the second: this is of course symmetric under reversal, and the cases of two photons scattered into only one waveguide are forbidden and have amplitude 0, since the difference in the number of photons in the two cavities is conserved in the presence of the pump beam. Using similar parameters as Fig 6(a) in Fischer, et al., we obtain a similar result:

```
tau = 0.05 / gamma # width of gaussian pulse
t0 = 3.5 * tau    # center of gaussian pulse
g0 = gamma       # amplitude of gaussian pulse
T = 100          # number of time bins
W = 2            # number of waveguides

tlist = np.linspace(0, 3/gamma, T)
phi = temporal_scattered_state(Hspdc(t0, g0, tau), cavity_vac, 2, spdc_c_ops, tlist)
amplitudes = np.zeros((W, W, T, T))

for i, tau1 in enumerate(tlist):
    for j, tau2 in enumerate(tlist):
        [a,b] = sorted([i,j]) # sort the indices to comply with time-ordering
        for wg1 in [0,1]:
            for wg2 in [0,1]:
                indices = [[] for _ in range(W)]
                indices[wg1].append(a)
                indices[wg2].append(b)
                basisVec = temporal_basis_vector(indices, T)
                amplitudes[wg1,wg2,i,j] = np.abs((basisVec.dag() * phi).full().item())**2

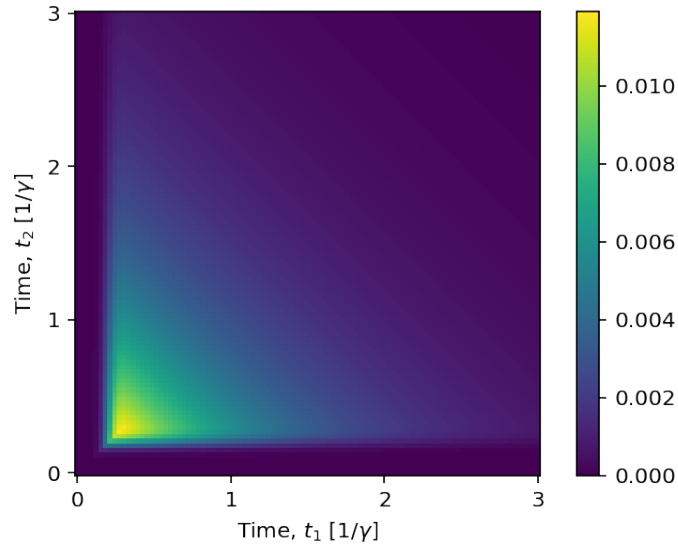
# Plot the correlation for emission times emitted into different waveguides; note
# that amplitudes[0][0] = amplitudes[1][1] = 0 and amplitudes[0][1] = amplitudes[1][0].
fig, ax1 = plt.subplots(1,1)
```

```

cax = ax1.imshow(amplitudes[0][1], interpolation='nearest', origin='lower')
ax1.set_xticks(np.linspace(0,T-1,4))
ax1.set_xticklabels([0, 1, 2, 3])
ax1.set_xlabel("Time,  $t_1$  [ $1/\gamma$ ]")
ax1.set_yticks(np.linspace(0,T-1,4))
ax1.set_yticklabels([0, 1, 2, 3])
ax1.set_ylabel("Time,  $t_2$  [ $1/\gamma$ ]")
fig.colorbar(cax)
plt.show()

```

Output:



F.11 Multi-waveguide photon emission probability

Finally, we can compute the variation in probability of single-and two-photon emission as a function of the pulse length. This simulation exhibits a slight variation from the expected behavior in Figure 5(c) of Fischer, et al., more apparent at larger times, due to the interaction timescale of interest increasing relative to the total timescale as a function of pulse length. However, the results do closely resemble the expected analytical results:

```

emission_nums = [0, 2]
pulse_lengths = np.linspace(0.05/gamma, 1.1 / gamma, 50)

```

```

tlist = np.geomspace(1/gamma, 21/gamma, 50) - 1/gamma

def scattering_probability_multiprocess(pulse_length, n):
    tau = pulse_length
    t0 = 3.5 * tau
    H = Hspdc(t0, gamma, tau)
    return scattering_probability(H, cavity_vac, n, spdc_c_ops, tlist)

pool = Pool(worker_count)
probs = {}
for n in emission_nums:
    args = [(pulse_length, n) for pulse_length in pulse_lengths]
    probs[n] = np.array(pool.starmap(scattering_probability_multiprocess, args))
pool.close()

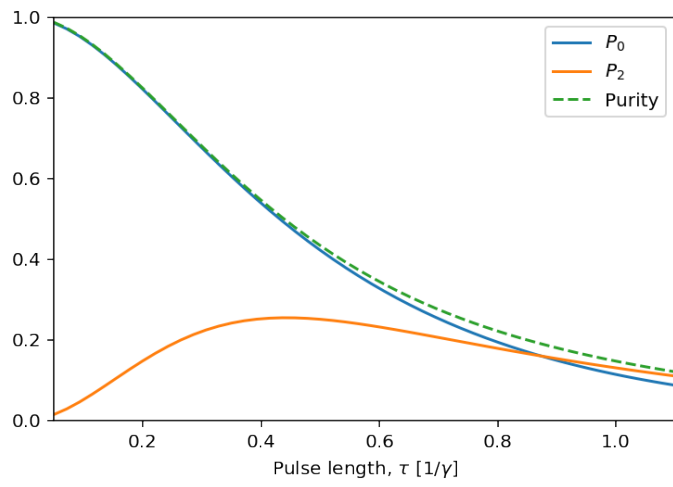
# Compute the purity of the output state
purity = [probs[2][p] / (1-probs[0][p]) for p in range(len(pulse_lengths))]

# Plot it
for n in probs:
    plt.plot(pulse_lengths / gamma, probs[n], label = "$P_{{}}$".format(n))
plt.plot(pulse_lengths / gamma, purity, '--', label = "Purity")
plt.ylim(0,1)
plt.xlim(pulse_lengths[0]/gamma, pulse_lengths[-1]/gamma)
plt.xlabel("Pulse length, $\tau$ $[1/\gamma]$")
plt.legend()
plt.show()

```

This notebook was executed on April 8, 2018, using QuTiP version 4.3.0 and NumPy version 1.13.3.

Output:



Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv: 1603.04467 [cs]*, 2015. Software available from tensorflow.org.
- [2] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A System for Large-Scale Machine Learning. In *Operating Systems Design and Implementation*, pages 265–283, Savannah, GA, 2016.
- [3] Yaser S. Abu-Mostafa and Demetri Psaltis. Optical Neural Computers. *Scientific American*, 256(3):88–95, March 1987.
- [4] Akshay Agrawal, Akshay Naresh Modi, Alexandre Passos, Allen Lavoie, Ashish Agarwal, Asim Shankar, Igor Ganichev, Josh Levenberg, Mingsheng Hong, Rajat Monga, and Shanqing Cai. TensorFlow Eager: A Multi-Stage, Python-Embedded DSL for Machine Learning. *arXiv:1903.01855 [cs]*, February 2019.
- [5] Igor Aharonovich, Dirk Englund, and Milos Toth. Solid-state single-photon emitters. *Nature Photon*, 10(10):631–641, October 2016.
- [6] M. N. Ahmed, J. Chong, and D. S. Ha. A 100 Gb/s transimpedance amplifier in 65 nm CMOS technology for optical communications. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1885–1888, June 2014.

- [7] A. V. Akimov, A. Mukherjee, C. L. Yu, D. E. Chang, A. S. Zibrov, P. R. Hemmer, H. Park, and M. D. Lukin. Generation of single optical plasmons in metallic nanowires coupled to quantum dots. *Nature*, 450(7168):402–406, 2007.
- [8] I. Alonso Calafell, T. Strömberg, D. R.M. Arvidsson-Shukur, L. A. Rozema, V. Saggio, C. Grainger, N. C. Harris, M. Prabhu, J. Carolan, M. Hochberg, T. Baehr-Jones, D. Englund, C. H.W. Barnes, and P. Walther. Trace-free counterfactual communication with a nanophotonic processor. *npj Quantum Information*, 5(61), 2019.
- [9] Andrea Annoni, Emanuele Guglielmi, Marco Carminati, Giorgio Ferrari, Marco Sampietro, David Ab Miller, Andrea Melloni, and Francesco Morichetti. Unscrambling light - Automatically undoing strong mixing between modes. *Light: Science and Applications*, 6(12), 2017.
- [10] Juan Miguel Arrazola, Thomas R. Bromley, Josh Izaac, Casey R. Myers, Kamil Brádler, and Nathan Killoran. Machine learning method for state preparation and gate synthesis on photonic quantum computers. *arXiv preprint*, 7 2018.
- [11] Juan Miguel Arrazola, Thomas R Bromley, Josh Izaac, Casey R Myers, Kamil Brádler, and Nathan Killoran. Machine learning method for state preparation and gate synthesis on photonic quantum computers. *Quantum Science and Technology*, 4(2):024004, 1 2019.
- [12] Warit Asavanant, Yu Shiozawa, Shota Yokoyama, Baramée Charoensombutamon, Hiroki Emura, Rafael N. Alexander, Shuntaro Takeda, Jun-ichi Yoshikawa, Nicolas C. Menicucci, Hidehiro Yonezawa, and Akira Furusawa. Generation of time-domain-multiplexed two-dimensional cluster state. *Science*, 366(6463):373–376, October 2019. Publisher: American Association for the Advancement of Science Section: Report.
- [13] Assa Auerbach. *Interacting electrons and quantum magnetism*. New York, Springer-Verlag, USA, 1st edition, 1994.
- [14] Koji Azuma, Kiyoshi Tamaki, and Hoi Kwong Lo. All-photonic quantum repeaters. *Nature Communications*, 6, 2015.
- [15] Thomas M. Babinec, Birgit J.M. Hausmann, Mughees Khan, Yinan Zhang, Jeronimo R. Maze, Philip R. Hemmer, and Marko Lončar. A diamond nanowire single-photon source. *Nature Nanotechnology*, 5(3):195–199, 2010.
- [16] Tom Baehr-Jones, Ran Ding, Ali Ayazi, Thierry Pinguet, Matt Streshinsky, Nick Harris, Jing Li, Li He, Mike Gould, Yi Zhang, Andy Eu-Jin Lim, Tsung-Yang Liow, Selin Hwee-Gee Teo, Guo-Qiang Lo, and Michael Hochberg. A 25 Gb/s Silicon Photonics Platform. *arXiv e-prints*, page arXiv:1203.0767, Mar 2012.

- [17] Qiaoliang Bao, Han Zhang, Zhenhua Ni, Yu Wang, Lakshminarayana Polavarapu, Zexiang Shen, Qing-Hua Xu, Dingyuan Tang, and Kian Ping Loh. Monolayer graphene as a saturable absorber in a mode-locked laser. *Nano Research*, 4(3):297–307, March 2011.
- [18] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. Divincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995.
- [19] Ben Bartlett. A distributed simulation framework for quantum networks and channels. *arXiv:1808.07047*, 8 2018.
- [20] Ben Bartlett, Avik Dutt, and Shanhui Fan. Deterministic photonic quantum computation in a synthetic time dimension. *Optica*, 8(12):1515, 2021.
- [21] Ben Bartlett and Shanhui Fan. Universal programmable photonic architecture for quantum information processing. *Physical Review A*, 101(4):042319, 2020.
- [22] Ben Bartlett, Momchil Minkov, Tyler Hughes, and Ian A. D. Williamson. Neuroptica: Flexible simulation package for optical neural networks. <https://github.com/fancompute/neuroptica>, 2019.
- [23] Andreas Bärtschi and Stephan Eidenbenz. Deterministic Preparation of Dicke States. *arXiv preprint*, 2019.
- [24] Yuval Baum and Gil Refael. Setting Boundaries with Memory: Generation of Topological Boundary States in Floquet-Induced Synthetic Crystals. *Phys. Rev. Lett.*, 120(10):106402, March 2018.
- [25] Kristin M. Beck, Mahdi Hosseini, Yiheng Duan, and Vladan Vuletic. Large conditional single-photon cross-phase modulation. *PNAS*, 113(35):9740–9744, August 2016. Publisher: National Academy of Sciences Section: Physical Sciences.
- [26] Bryn A. Bell, Kai Wang, Alexander S. Solntsev, Dragomir N. Neshev, Andrey A. Sukhorukov, and Benjamin J. Eggleton. Spectral photonic lattices with complex long-range coupling. *Optica*, 4(11):1433–1436, November 2017.
- [27] Charles H Bennett, Gilles Brassard, Richard Jozsa, Asher Peres, William K Wootters, C Crépeau, and Richard Jozsa. Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels. *Physical Review Letters*, 70(12):1895–1899, 1993.
- [28] Charles H Bennett and Stephen J Wiesner. Communication via One- and Two-Particle Operators on Einstein-Podolsky-Rosen States. *Physical Review Letters*, 69(20):2881–2884, 1992.

- [29] A. Bermudez, X. Xu, R. Nigmatullin, J. O’Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller. Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation. *Physical Review X*, 7(4):041061, 2017.
- [30] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
- [31] Nicolaas Bloembergen. *Nonlinear Optics*. World Scientific, Singapore, 4th edition, 1996.
- [32] O. Boada, A. Celi, J. I. Latorre, and M. Lewenstein. Quantum Simulation of an Extra Dimension. *Phys. Rev. Lett.*, 108(13):133001, March 2012.
- [33] Robert W. Boyd. *Nonlinear Optics, Third Edition*. Academic Press, Inc., USA, 3rd edition, 2008.
- [34] Robert W Boyd and Svetlana G Lukishova. *Quantum Photonics: Pioneering Advances and Emerging Applications*. Springer, 2019.
- [35] David O. Bracher, Xingyu Zhang, and Evelyn L. Hu. Selective Purcell enhancement of two closely linked zero-phonon transitions of a silicon carbide color center. *Proceedings of the National Academy of Sciences of the United States of America*, 114(16):4060–4065, 2017.
- [36] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 2019.
- [37] Roel Burgwal, William R. Clements, Devin H. Smith, James C. Gates, W. Steven Kolthammer, Jelmer J. Renema, and Ian A. Walmsley. Using an imperfect photonic network to implement random unitaries. *Optics Express*, 25(23):28236, 11 2017.
- [38] A. H. Burrell, D. J. Szwer, S. C. Webster, and D. M. Lucas. Scalable simultaneous multiqubit readout with 99.99% single-shot fidelity. *Physical Review A*, 81(4):040302, 2010.
- [39] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 2017.
- [40] José Capmany and Dalma Novak. Microwave photonics combines two worlds. *Nature Photonics*, 1(6):319, 2007.
- [41] Jaime Cardenas, Mark A. Foster, Nicolás Sherwood-Droz, Carl B. Poitras, Hugo L. R. Lira, Beibei Zhang, Alexander L. Gaeta, Jacob B. Khurgin, Paul Morton, and Michal Lipson. Wide-bandwidth continuously tunable optical delay line using silicon microring resonators. *Optics Express*, 18(25):26525, December 2010.

- [42] Jacques Carolan, Christopher Harrold, Chris Sparrow, Enrique Martín-López, Nicholas J. Russell, Joshua W. Silverstone, Peter J. Shadbolt, Nobuyuki Matsuda, Manabu Oguma, Miki-taka Itoh, Graham D. Marshall, Mark G. Thompson, Jonathan C.F. Matthews, Toshikazu Hashimoto, Jeremy L. O'Brien, and Anthony Laing. Universal linear optics. *Science*, 349(6249):711–716, 2015.
- [43] A. Celi, P. Massignan, J. Ruseckas, N. Goldman, I.*B. Spielman, G. Juzeliunas, and M. Lewenstein. Synthetic Gauge Fields in Synthetic Dimensions. *Phys. Rev. Lett.*, 112(4):043001, January 2014.
- [44] Hamidreza Chalabi, Sabyasachi Barik, Sunil Mittal, Thomas E. Murphy, Mohammad Hafezi, and Edo Waks. Synthetic Gauge Field for Two-Dimensional Time-Multiplexed Quantum Random Walks. *Physical Review Letters*, 123(15):150503, 2019.
- [45] D.*E. Chang, J.*S. Douglas, A. Gonzalez-Tudela, C.-L. Hung, and H.*J. Kimble. Colloquium: Quantum matter built from nanoscopic lattices of atoms and photons. *Rev. Mod. Phys.*, 90(3):031002, August 2018.
- [46] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific Reports*, 8(1):12324, August 2018.
- [47] Tzu-Han Chang, Brian M. Fields, May E. Kim, and Chen-Lung Hung. Microring resonators on a suspended membrane circuit for atom–light interactions. *Optica*, 6(9):1203–1210, September 2019. Publisher: Optical Society of America.
- [48] Sung Won Chung, Hooman Abediasl, and Hossein Hashemi. A 1024-element scalable optical phased array in 0.18 μm SOI CMOS. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 60:262–263, 2017.
- [49] F. Ciccarello, D. E. Browne, L. C. Kwek, H. Schomerus, M. Zarccone, and S. Bose. Quasideterministic realization of a universal quantum gate in a single scattering process. *Physical Review A - Atomic, Molecular, and Optical Physics*, 85(5):1–5, 2012.
- [50] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. An Optimal Design for Universal Multiport Interferometers. *Optica*, 3(2):1460–1465, 2016.
- [51] F. D. Coarer, M. Sciamanna, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and D. Rontani. All-Optical Reservoir Computing on a Photonic Chip Using Silicon-Based Ring Resonators. *IEEE Journal of Selected Topics in Quantum Electronics*, 24(6):1–8, November 2018.

- [52] Shane Colburn, Yi Chu, Eli Shilzerman, and Arka Majumdar. Optical frontend for a convolutional neural network. *Applied Optics*, 58(12):3179–3186, April 2019.
- [53] Yves Colombe, Tilo Steinmetz, Guilhem Dubois, Felix Linke, David Hunger, and Jakob Reichel. Strong atom-field coupling for Bose-Einstein condensates in an optical cavity on a chip. *Nature*, 450(7167):272–276, November 2007. Number: 7167 Publisher: Nature Publishing Group.
- [54] G. Coudourier-Maruri, F. Ciccarello, Y. Omar, M. Zarccone, R. de Coss, and S. Bose. Implementing quantum gates through scattering between a static and a flying qubit. *Physical Review A - Atomic, Molecular, and Optical Physics*, 82(5):1–7, 2010.
- [55] Jacob P. Covey, Alp Sipahigil, Szilard Szoke, Neil Sinclair, Manuel Endres, and Oskar Painter. Telecom-Band Quantum Optics with Ytterbium Atoms and Silicon Nanophotonics. *Phys. Rev. Applied*, 11(3):034044, March 2019. Publisher: American Physical Society.
- [56] Rumén Dangovski, Li Jing, and Marin Soljacic. Rotational Unit of Memory. In *International Conference on Learning Representations*, 10 2017.
- [57] Persi Diaconis and Peter J. Forrester. Hurwitz and the origins of random matrix theory in mathematics. *Random Matrices: Theory and Applications*, 06(01):1730001, 1 2017.
- [58] Po Dong, Wei Qian, Shirong Liao, Hong Liang, Cheng-Chih Kung, Ning-Ning Feng, Roshanak Shafiiha, Joan Fong, Dazeng Feng, Ashok V. Krishnamoorthy, and Mehdi Asghari. Low loss shallow-ridge silicon waveguides. *Optics Express*, 18(14):14474, 2010.
- [59] L. M. Duan and H. J. Kimble. Scalable photonic quantum computation through cavity-assisted interactions. *Physical Review Letters*, 92(12):127902, 2004.
- [60] L. M. Duan, A. Kuzmich, and H. J. Kimble. Cavity QED and quantum-information processing with “hot” trapped atoms. *Physical Review A*, 67(3):032305, 2003.
- [61] L.-M. Duan, B. Wang, and H. J. Kimble. Robust quantum gates on neutral atoms with cavity-assisted photon scattering. *Phys. Rev. A*, 72(3):032333, September 2005.
- [62] François Duport, Anteo Smerieri, Akram Akrouf, Marc Haelterman, and Serge Massar. Fully analogue photonic reservoir computer. *Scientific Reports*, 6:22381, March 2016.
- [63] Avik Dutt, Qian Lin, Luqi Yuan, Momchil Minkov, Meng Xiao, and Shanhui Fan. A single photonic cavity with two independent physical synthetic dimensions. *Science*, 367(6473):59–64, 2020.

- [64] Avik Dutt, Qian Lin, Luqi Yuan, Momchil Minkov, Meng Xiao, and Shanhui Fan. A single photonic cavity with two independent physical synthetic dimensions. *Science*, 367(6473):59–64, January 2020.
- [65] Avik Dutt, Momchil Minkov, Qian Lin, Luqi Yuan, David A. B. Miller, and Shanhui Fan. Experimental band structure spectroscopy along a synthetic dimension. *Nature Communications*, 10(1):3122, 2019.
- [66] Omjyoti Dutta, Mariusz Gajda, Philipp Hauke, Maciej Lewenstein, Dirk-Soren Luhmann, Boris A Malomed, Tomasz Sowinski, and Jakub Zakrzewski. Non-standard Hubbard models in optical lattices: a review. *Rep. Prog. Phys.*, 78(6):066001, June 2015.
- [67] Bryan Eastin and Steven T. Flammia. Q-circuit Tutorial. *arXiv preprint*, 6 2004.
- [68] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Physical Review Letters*, 102(11):1–4, 2009.
- [69] Sophia E. Economou, Netanel Lindner, and Terry Rudolph. Optically Generated 2-Dimensional Photonic Cluster State from Coupled Quantum Dots. *Phys. Rev. Lett.*, 105(9):093601, August 2010.
- [70] Kevin A. Fischer, Rahul Trivedi, Vinay Ramasesh, Irfan Siddiqi, and Jelena Vučković. Scattering into one-dimensional waveguides from a coherently-driven quantum-optical system. *Quantum*, 2(69), 2018.
- [71] Fulvio Flamini, Nicolò Spagnolo, Niko Viggianiello, Andrea Crespi, Roberto Osellame, and Fabio Sciarrino. Benchmarking integrated linear-optical architectures for quantum information processing. *Scientific Reports*, 7(1):15133, 12 2017.
- [72] Ilya Fushman, Dirk Englund, Andrei Faraon, Nick Stoltz, Pierre Petroff, and Jelena Vuckovic. Controlled Phase Shifts with a Single Quantum Dot. *Science*, 320(5877):769–772, May 2008. Publisher: American Association for the Advancement of Science Section: Report.
- [73] I. García López, A. Awny, P. Rito, M. Ko, A. C. Ulusoy, and D. Kissinger. 100 gb/s differential linear bias with less than 10 pa/ $\sqrt{\text{hz}}$ in 130-nm size: c bicmos. *IEEE Journal of Solid-State Circuits*, 53(2):458–469, Feb 2018.
- [74] Roger Gehr, Jurgen Volz, Guilhem Dubois, Tilo Steinmetz, Yves Colombe, Benjamin L. Lev, Romain Long, Jerome Esteve, and Jakob Reichel. Cavity-Based Single Atom Preparation and High-Fidelity Hyperfine State Readout. *Phys. Rev. Lett.*, 104(20):203602, May 2010.
- [75] Alan Genz. Methods for Generating Random Orthogonal Matrices. *Monte Carlo and Quasi-Monte Carlo Methods*, pages 199–213, 1998.

- [76] Paolo Ghelfi, Francesco Laghezza, Filippo Scotti, Giovanni Serafino, Amerigo Capria, Sergio Pinna, Daniel Onori, Claudio Porzi, Mirco Scaffardi, Antonio Malacarne, Valeria Vercesi, Emma Lazzeri, Fabrizio Berizzi, and Antonella Bogoni. A fully photonics-based coherent radar system. *Nature*, 507(7492):341, March 2014.
- [77] Thierry Giamarchi, Christian Rüegg, and Oleg Tchernyshyov. Bose-Einstein condensation in magnetic insulators. *Nature Physics*, 4(3):198–204, 2008.
- [78] Wallace Givens. Computation of plane unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1), 1958.
- [79] Joseph W. Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, 2005.
- [80] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997.
- [81] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. *arXiv preprint*, 2009.
- [82] Daniel Gottesman and Isaac L Chuang. Quantum Teleportation is a Universal Computational Primitive. *arXiv preprint*, 1999.
- [83] Markus Gräfe, René Heilmann, Maxime Lebugle, Diego Guzman-Silva, Armando Perez-Leija, and Alexander Szameit. Integrated photonic quantum walks. *Journal of Optics*, 18(10):23, 2016.
- [84] M Grassl, Th Beth, and T Pellizzari. Codes for the quantum erasure channel. *Physical Review A*, 56(1):33–38, 1997.
- [85] Daniel Greenberger, Michael Horne, and Anton Zeilinger. Going beyond Bell’s Theorem. *arXiv preprint*, 2007.
- [86] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pages 212–219, 1996.
- [87] Laszlo Gyongyosi and Sandor Imre. Properties of the Quantum Channel. *arXiv preprint*, aug 2012.
- [88] Laszlo Gyongyosi, Sandor Imre, and Hung Viet Nguyen. A Survey on Quantum Channel Capacities. *arXiv preprint*, jan 2018.
- [89] Bastian Hacker, Stephan Welte, Gerhard Rempe, and Stephan Ritter. A photon-photon quantum gate based on a single atom in an optical resonator. *Nature*, 536(7615):193–196, August 2016. Number: 7615 Publisher: Nature Publishing Group.

- [90] Lisa Hales and Sean Hallgren. Improved quantum Fourier transform algorithm and applications. *Annual Symposium on Foundations of Computer Science - Proceedings*, pages 515–525, 2000.
- [91] Sangyoon Han, Tae Joon Seok, Niels Quack, Byung-Wook Yoo, and Ming C. Wu. Large-scale silicon photonic switches with movable directional couplers. *Optica*, 2(4):370, 2015.
- [92] Nicholas C. Harris, Jacques Carolan, Darius Bunandar, Mihika Prabhu, Michael Hochberg, Tom Baehr-Jones, Michael L. Fanto, A. Matthew Smith, Christopher C. Tison, Paul M. Alsing, and Dirk Englund. Linear programmable nanophotonic processors. *Optica*, 5(12):1623, 2018.
- [93] Nicholas C. Harris, Gregory R. Steinbrecher, Mihika Prabhu, Yoav Lahini, Jacob Mower, Darius Bunandar, Changchen Chen, Franco N.C. Wong, Tom Baehr-Jones, Michael Hochberg, Seth Lloyd, and Dirk Englund. Quantum transport simulations in a programmable nanophotonic processor. *Nature Photonics*, 11(7):447–452, 2017.
- [94] Nicholas Christopher Harris. *Programmable Nanophotonics for Quantum Information Processing and Artificial Intelligence*. Thesis, Massachusetts Institute of Technology, 2017.
- [95] T. P. Harty, D. T.C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas. High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. *Physical Review Letters*, 113(22):220501, 2014.
- [96] Paul Hilaire, Leonid Vidro, Hagai S. Eisenberg, and Sophia E. Economou. Near-deterministic hybrid generation of arbitrary photonic graph states using a single quantum emitter and linear optics, May 2022.
- [97] Mark Hillery, Vladimír Bužek, and André Berthiaume. Quantum secret sharing. *Physical Review A - Atomic, Molecular, and Optical Physics*, 59(3):1829–1834, 1999.
- [98] Yaowen Hu, Christian Reimer, Amirhassan Shams-Ansari, Mian Zhang, and Marko Loncar. Realization of high-dimensional frequency crystals in electro-optic microcombs. *Optica*, 7(9):1189–1194, September 2020.
- [99] Dario Hugel and Belen Paredes. Chiral ladders and the edges of quantum Hall insulators. *Physical Review A*, 89(2):023619, February 2014.
- [100] Tyler W Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan. Training of photonic neural networks through in situ backpropagation. *Optica*, 5(7):864–871, 2018.
- [101] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7):864–871, July 2018.

- [102] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7):864, 7 2018.
- [103] Adolf Hurwitz. über die Erzeugung der Invarianten durch Integration. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, pages 71–72, 1897.
- [104] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue. A coherent Ising machine for 2000-node optimization problems. *Science*, 354(6312):603–606, November 2016.
- [105] Xiantao Jiang, Simon Gross, Michael J. Withford, Han Zhang, Dong-Il Yeom, Fabian Rotermund, and Alexander Fuerbach. Low-dimensional nanomaterial saturable absorbers for ultrashort-pulsed waveguide lasers. *Optical Materials Express*, 8(10):3055, October 2018.
- [106] Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable Efficient Unitary Neural Networks (EUNN) and their application to RNNs. In *Proceedings of Machine Learning Research*, pages 1733–1741, 7 2017.
- [107] J. R. Johansson, P. D. Nation, and Franco Nori. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.
- [108] Chaitali Joshi, Alessandro Farsi, Stephane Clemmen, Sven Ramelow, and Alexander L. Gaeta. Frequency multiplexing for quasi-deterministic heralded single-photon sources. *Nature Communications*, 9(1):847, February 2018.
- [109] Chaitali Joshi, Alessandro Farsi, Avik Dutt, Bok Young Kim, Xingchen Ji, Yun Zhao, Andrew M. Bishop, Michal Lipson, and Alexander L. Gaeta. Frequency-Domain Quantum Interference with Correlated Photons from an Integrated Microresonator. *Physical Review Letters*, 124(14):143601, 2020.
- [110] Richard Jozsa. An introduction to measurement based quantum computation. *arXiv:0508124*, 2005.
- [111] F. Kaneda and P. G. Kwiat. High-efficiency single-photon generation via large-scale active time multiplexing. *Science Advances*, 5(10):1–7, 2019.
- [112] Phillip Kaye and Michele Mosca. Quantum Networks for Generating Arbitrary Quantum States. *arXiv preprint*, 2004.
- [113] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *arXiv preprint*, 6 2018.

- [114] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry Fields: A Software Platform for Photonic Quantum Computing. *arXiv preprint*, apr 2018.
- [115] Je Hyung Kim, Shahriar Aghaieimeibodi, Christopher J.K. Richardson, Richard P. Leavitt, Dirk Englund, and Edo Waks. Hybrid Integration of Solid-State Quantum Emitters on a Silicon Photonic Chip. *Nano Letters*, 17(12):7394–7400, 2017.
- [116] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014.
- [117] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint*, 12 2014.
- [118] Diederik P Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015.
- [119] E. Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005.
- [120] E Knill and R Laflamme. Concatenated Quantum Codes. *arXiv preprint*, 1996.
- [121] E Knill, R Laflamme, A Ashikhmin, H Barnum, L Viola, and W H Zurek. Introduction to Quantum Error Correction. *arXiv preprint*, 2002.
- [122] E. Knill, R. Laflamme, and G. Milburn. Thresholds for Linear Optics Quantum Computation. *arXiv preprint*, 2000.
- [123] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 1 2001.
- [124] Daniel Koch, Avery Torrance, David Kinghorn, Saahil Patel, Laura Wessing, and Paul M Alsing. Simulating Quantum Algorithms Using Fidelity and Coherence Time as Principle Models for Error. *arXiv preprint*, 2019.
- [125] Pieter Kok, W. J. Munro, Kae Nemoto, T. C. Ralph, Jonathan P. Dowling, and G. J. Milburn. Linear optical quantum computing with photonic qubits. *Reviews of Modern Physics*, 79(1):135–174, 1 2007.
- [126] C. Koos, L. Jacome, C. Poulton, J. Leuthold, and W. Freude. Nonlinear silicon-on-insulator waveguides for all-optical signal processing. *Optics Express*, 15(10):5976–5990, May 2007.
- [127] Guillaume Labroille, Bertrand Denolle, Pu Jian, Jean François Morizur, Philippe Genevaux, and Nicolas Treps. Efficient and mode selective spatial mode multiplexer based on multi-plane light conversion. In *2014 IEEE Photonics Conference, IPC 2014*, 2014.

- [128] Guillaume Labroille, Pu Jian, Nicolas Barré, Bertrand Denolle, and Jean-François Morizur. Mode Selective 10-Mode Multiplexer based on Multi-Plane Light Conversion. *Optical Fiber Communication Conference*, 2016.
- [129] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.
- [130] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. Photonic information processing beyond Turing: An optoelectronic implementation of reservoir computing. *Optics Express*, 20(3):3241, January 2012.
- [131] Mikkel V. Larsen, Xueshi Guo, Casper R. Breum, Jonas S. Neergaard-Nielsen, and Ulrik L. Andersen. Deterministic generation of a two-dimensional cluster state. *Science*, 366(6463):369–372, October 2019. Publisher: American Association for the Advancement of Science Section: Report.
- [132] A. Laucht, S. Pütz, T. Günthner, N. Hauke, R. Saive, S. Frédérick, M. Bichler, M. C. Amann, A. W. Holleitner, M. Kaniber, and J. J. Finley. A waveguide-coupled on-chip single-photon source. *Physical Review X*, 2(1):1–7, 2012.
- [133] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [134] Christian Leefmans, Avik Dutt, James Williams, Luqi Yuan, Midya Parto, Franco Nori, Shan-hui Fan, and Alireza Marandi. Topological dissipation in a time-multiplexed photonic resonator network. *Nature Physics*, 015006(1846273):2013–2014, 2022.
- [135] Christian Leefmans, Avik Dutt, James Williams, Luqi Yuan, Midya Parto, Franco Nori, Shan-hui Fan, and Alireza Marandi. Topological dissipation in a time-multiplexed photonic resonator network. *Nat. Phys.*, February 2022.
- [136] A. L. Lentine and D. A. B. Miller. Evolution of the SEED technology: Bistable logic gates to optoelectronic smart pixels. *IEEE Journal of Quantum Electronics*, 29(2):655–669, February 1993.
- [137] Xinwei Li, Motoaki Bamba, Qi Zhang, Saeed Fallahi, Geoff C. Gardner, Weilu Gao, Minhan Lou, Katsumasa Yoshioka, Michael J. Manfra, and Junichiro Kono. Vacuum Bloch-Siegert shift in Landau polaritons with ultra-high cooperativity. *Nature Photon*, 12(6):324–329, June 2018.
- [138] Ying Li, Peter C. Humphreys, Gabriel J. Mendoza, and Simon C. Benjamin. Resource costs for fault-tolerant linear optical quantum computing. *Physical Review X*, 5(4):041007, 2015.

- [139] Li Liu, Jianji Dong, Dingshan Gao, Aoling Zheng, and Xinliang Zhang. On-chip passive three-port circuit of all-optical ordered-route transmission. *Scientific Reports*, 5:1–9, 2015.
- [140] Rodney Loudon. *The Quantum Theory of Light*. Oxford University Press, Oxford, 3rd edition, 2000.
- [141] Xiyuan Lu, Ashutosh Rao, Gregory Moille, Daron A. Westly, and Kartik Srinivasan. Universal frequency engineering tool for microcavity nonlinear optics: multiple selective mode splitting of whispering-gallery resonances. *Photon. Res., PRJ*, 8(11):1676–1686, November 2020.
- [142] Xi-Wang Luo, Xingxiang Zhou, Chuan-Feng Li, Jin-Shi Xu, Guang-Can Guo, and Zheng-Wei Zhou. Quantum simulation of 2D topological physics in a 1D array of optical cavities. *Nature Communications*, 6:7704, July 2015.
- [143] Eran Lustig, Steffen Weimann, Yonatan Plotnik, Yaakov Lumer, Miguel A. Bandres, Alexander Szameit, and Mordechai Segev. Photonic topological insulator in synthetic dimensions. *Nature*, 567(7748):356–360, March 2019.
- [144] B. Machielse, S. Bogdanovic, S. Meesala, S. Gauthier, M.*J. Burek, G. Joe, M. Chalupnik, Y.*I. Sohn, J. Holzgrafe, R.*E. Evans, C. Chia, H. Atikian, M.*K. Bhaskar, D.*D. Sukachev, L. Shao, S. Maity, M.*D. Lukin, and M. Loncar. Quantum Interference of Electromechanically Stabilized Emitters in Nanophotonic Devices. *Phys. Rev. X*, 9(3):031022, August 2019. Publisher: American Physical Society.
- [145] Arka Majumdar and Armand Rundquist. Cavity-enabled self-electro-optic bistability in silicon photonics. *Optics Letters*, 39(13):3864, July 2014.
- [146] M. Mancini, G. Pagano, G. Cappellini, L. Livi, M. Rider, J. Catani, C. Sias, P. Zoller, M. Inguscio, M. Dalmonte, and L. Fallani. Observation of chiral edge states with neutral fermions in synthetic Hall ribbons. *Science*, 349(6255):1510–1513, September 2015.
- [147] Alireza Marandi, Zhe Wang, Kenta Takata, Robert L. Byer, and Yoshihisa Yamamoto. Network of time-multiplexed optical parametric oscillators as a coherent Ising machine. *Nature Photonics*, 8(12):937–942, December 2014.
- [148] David Marpaung, Chris Roeloffzen, René Heideman, Arne Leinse, Salvador Sales, and José Capmany. Integrated microwave photonics. *Laser & Photonics Reviews*, 7(4):506–538, July 2013.
- [149] Ivar Martin, Gil Refael, and Bertrand Halperin. Topological Frequency Conversion in Strongly Driven Quantum Systems. *Physical Review X*, 7(4):041008, October 2017.
- [150] Serge Massar and Stéphane Clemmen. Resource efficient single photon source based on active frequency multiplexing. *Optics Letters*, 46(12):2832, 2021.

- [151] Michael Mathieu and Yann LeCun. Fast Approximation of Rotations and Hessians matrices. *arXiv preprint*, 4 2014.
- [152] J. McKeever, A. Boca, A. D. Boozer, J. R. Buck, and H. J. Kimble. Experimental realization of a one-atom laser in the regime of strong coupling. *Nature*, 425(6955):268–271, September 2003.
- [153] J. McKeever, A. Boca, A. D. Boozer, R. Miller, J. R. Buck, A. Kuzmich, and H. J. Kimble. Deterministic Generation of Single Photons from One Atom Trapped in a Cavity. *Science*, 303(5666):1992–1994, 2004.
- [154] Peter L. McMahon, Alireza Marandi, Yoshitaka Haribara, Ryan Hamerly, Carsten Langrock, Shuhei Tamate, Takahiro Inagaki, Hiroki Takesue, Shoko Utsunomiya, Kazuyuki Aihara, Robert L. Byer, M. M. Fejer, Hideo Mabuchi, and Yoshihisa Yamamoto. A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science*, 354(6312):614–617, November 2016.
- [155] David A. B. Miller. Establishing Optimal Wave Communication Channels Automatically. *Journal of Lightwave Technology*, Vol. 31, Issue 24, pp. 3987-3994, 31(24):3987–3994, 12 2013.
- [156] David A. B. Miller. Self-aligning universal beam coupler. *Optics Express*, 21(5):6360, 3 2013.
- [157] David A B Miller. Self-configuring universal linear optical component. *Photonics Research*, 1(1):1, 2013.
- [158] David A. B. Miller. Perfect optics with imperfect components. *Optica*, 2(8):747, 2015.
- [159] David A. B. Miller. Setting up meshes of interferometers – reversed local light interference method. *Optics Express*, 25(23):29233, 11 2017.
- [160] David A.B. Miller. Sorting out light. *Science*, 347(6229):1423–1424, 2015.
- [161] R. Miller, T. E. Northup, K. M. Birnbaum, A. Boca, A. D. Boozer, and H. J. Kimble. Trapped atoms in cavity QED: coupling quantized light and matter. *J. Phys. B: At. Mol. Opt. Phys.*, 38(9):S551–S565, April 2005. Publisher: IOP Publishing.
- [162] C. Monroe. Quantum information processing with atoms and cavities. *Nature*, 416:238–246, 2002.
- [163] Jean-François Morizur, Lachlan Nicholls, Pu Jian, Seiji Armstrong, Nicolas Treps, Boris Hage, Magnus Hsu, Warwick Bowen, Jiri Janousek, and Hans-A. Bachor. Programmable unitary spatial mode manipulation. *Journal of the Optical Society of America A*, 27(11):2524, 11 2010.

- [164] Keith R. Motes, Alexei Gilchrist, Jonathan P. Dowling, and Peter P. Rohde. Scalable boson sampling with time-bin encoding using a loop-based architecture. *Physical Review Letters*, 113(12):1–5, 2014.
- [165] M. Mottonen and J. J. Vartiainen. Decompositions of general quantum gates. *arXiv preprint*, 2005.
- [166] A. H. Myerson, D. J. Szwer, S. C. Webster, D. T.C. Allcock, M. J. Curtis, G. Imreh, J. A. Sherman, D. N. Stacey, A. M. Steane, and D. M. Lucas. High-fidelity readout of trapped-ion qubits. *Physical Review Letters*, 100(20):200502, 2008.
- [167] Michael A. Nielsen. Cluster-state quantum computation. *Reports on Mathematical Physics*, 57(1):147–161, 2006.
- [168] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniv edition, 2010.
- [169] K. Nozaki, S. Matsuo, A. Shinya, and M. Notomi. Amplifier-Free Bias-Free Receiver Based on Low-Capacitance Nanophotodetector. *IEEE Journal of Selected Topics in Quantum Electronics*, 24(2):1–11, March 2018.
- [170] Kengo Nozaki, Shinji Matsuo, Takuro Fujii, Koji Takeda, Akihiko Shinya, Eiichi Kuramochi, and Masaya Notomi. Femtofarad optoelectronic integration demonstrating energy-saving signal conversion and nonlinear functions. *Nature Photonics*, 13(7):454–459, April 2019.
- [171] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv e-prints*, page arXiv:1811.03378, Nov 2018.
- [172] Jeremy L. O’Brien. Optical Quantum Computing. *Science*, 318(5856):1567–1570, December 2007. Publisher: American Association for the Advancement of Science Section: Review.
- [173] Tomoki Ozawa and Iacopo Carusotto. Synthetic Dimensions with Magnetic Fields and Local Interactions in Photonic Lattices. *Phys. Rev. Lett.*, 118(1):013601, January 2017.
- [174] Tomoki Ozawa and Hannah M. Price. Topological quantum matter in synthetic dimensions. *Nature Reviews Physics*, 1:349–357, April 2019.
- [175] Tomoki Ozawa, Hannah M. Price, Nathan Goldman, Oded Zilberberg, and Iacopo Carusotto. Synthetic dimensions in integrated photonics: From optical isolation to four-dimensional quantum Hall physics. *Phys. Rev. A*, 93(4):043827, April 2016.
- [176] Sunil Pai. Neurophox: A simulation framework for unitary neural networks and photonic devices. <https://github.com/solgaardlab/neurophox>, 2018.

- [177] Sunil Pai, Ben Bartlett, Olav Solgaard, and David A. B. Miller. Matrix Optimization on Universal Unitary Photonic Devices. *Physical Review Applied*, 11(6):064044, 6 2019.
- [178] Sunil Pai, Ian A. D. Williamson, Tyler W. Hughes, Momchil Minkov, Olav Solgaard, Shanhui Fan, and David A. B. Miller. Parallel fault-tolerant programming of an arbitrary feedforward photonic network. *arXiv preprint*, 2019.
- [179] Vivek K. Pallipuram, Mohammad Bhuiyan, and Melissa C. Smith. A comparative study of GPU programming models and architectures using neural networks. *The Journal of Supercomputing*, 61(3):673–718, September 2012.
- [180] Nam Hun Park, Hwanseong Jeong, Sun Young Choi, Mi Hye Kim, Fabian Rotermund, and Dong-Il Yeom. Monolayer graphene saturable absorbers with strongly enhanced evanescent-field interaction for ultrafast fiber laser mode-locking. *Optics Express*, 23(15):19806, July 2015.
- [181] Bo Peng, Shuo Yan, Dali Cheng, Danying Yu, Zhanwei Liu, Vladislav V. Yakovlev, Luqi Yuan, and Xianfeng Chen. Novel optical neural network architecture with the temporal synthetic dimension. *arxiv preprint*, pages 1–18, 2021.
- [182] Bo Peng, Shuo Yan, Dali Cheng, Danying Yu, Zhanwei Liu, Vladislav V. Yakovlev, Luqi Yuan, and Xianfeng Chen. Novel optical neural network architecture with the temporal synthetic dimension. *arXiv:2101.08439 [physics]*, January 2021.
- [183] H. Peng, M. A. Nahmias, T. F. de Lima, A. N. Tait, and B. J. Shastri. Neuromorphic Photonic Integrated Circuits. *IEEE Journal of Selected Topics in Quantum Electronics*, 24(6):1–15, November 2018.
- [184] Daniel Perez, Ivana Gasulla, Jose Capmany, and Richard A. Soref. Hexagonal waveguide mesh design for universal multiport interferometers. In *2016 IEEE Photonics Conference, IPC 2016*, pages 285–286, 2017.
- [185] Daniel Pérez, Ivana Gasulla, and José Capmany. Programmable multifunctional integrated nanophotonics. *Nanophotonics*, 7(8):1351–1371, 2018.
- [186] Daniel Pérez, Ivana Gasulla, Lee Crudgington, David J. Thomson, Ali Z. Khokhar, Ke Li, Wei Cao, Goran Z. Mashanovich, and José Capmany. Multipurpose silicon photonics signal processor core. *Nature Communications*, 2017.
- [187] Daniel Pérez, Ivana Gasulla, Lee Crudgington, David J. Thomson, Ali Z. Khokhar, Ke Li, Wei Cao, Goran Z. Mashanovich, and José Capmany. Silicon RF-Photonics Processor Reconfigurable Core. In *European Conference on Optical Communication, ECOC*, 2018.

- [188] Hannes Pichler, Soonwon Choi, Peter Zoller, and Mikhail D. Lukin. Universal photonic quantum computation via time-delayed feedback. *PNAS*, 114(43):11362–11367, October 2017. Publisher: National Academy of Sciences Section: Physical Sciences.
- [189] Alexander Y. Piggott, Jan Petykiewicz, Logan Su, and Jelena Vučković. Fabrication-constrained nanophotonic inverse design. *Scientific Reports*, 2017.
- [190] T. B. Pittman and J. D. Franson. Cyclical quantum memory for photonic qubits. *Physical Review A - Atomic, Molecular, and Optical Physics*, 66(6):4, 2002.
- [191] A. V. Ponomarev, S. Denisov, and P. Hanggi. Ac-driven atomic quantum motor. *Physical Review Letters*, 102(23):1–4, 2009.
- [192] A. V. Ponomarev, S. Denisov, and P. Hanggi. Thermal equilibration between two quantum systems. *Physical Review Letters*, 106(1):1–4, 2011.
- [193] Monireh Moayedi Pour Fard, Ian A. D. Williamson, Matthew Edwards, Ke Liu, Sunil Pai, Ben Bartlett, Momchil Minkov, Tyler W. Hughes, Shanhui Fan, and Thien-An Nguyen. Experimental realization of arbitrary activation functions for optical neural networks. *Optics Express*, 28(8):12138, 2020.
- [194] John Preskill. Fault-tolerant quantum computation. *arXiv:9712048*, 1997.
- [195] Hannah Price, Yidong Chong, Alexander Khanikaev, Henning Schomerus, Lukas J. Maczewsky, Mark Kremer, Matthias Heinrich, Alexander Szameit, Oded Zilberberg, Yihao Yang, Baile Zhang, Andrea Alu, Ronny Thomale, Iacopo Carusotto, Philippe St-Jean, Alberto Amo, Avik Dutt, Luqi Yuan, Shanhui Fan, Xuefan Yin, Chao Peng, Tomoki Ozawa, and Andrea Blanco-Redondo. Roadmap on topological photonics. *J. Phys. Photonics*, 2022.
- [196] Demetri Psaltis, David Brady, Xiang-Guang Gu, and Steven Lin. Holography in artificial neural networks. *Nature*, 343(6256):325–330, January 1990.
- [197] Matthew Pysher, Yoshichika Miwa, Reihaneh Shahrokhshahi, Russell Bloomer, and Olivier Pfister. Parallel Generation of Quadripartite Cluster Entanglement in the Optical Frequency Comb. *Phys. Rev. Lett.*, 107(3):030505, July 2011.
- [198] Marina Radulaski, Ranojoy Bose, Tho Tran, Thomas Van Vaerenbergh, David Kielpinski, and Raymond G. Beausoleil. Thermally Tunable Hybrid Photonic Architecture for Nonlinear Optical Circuits. *ACS Photonics*, 5(11):4323–4329, November 2018.
- [199] Marina Radulaski, Jingyuan Linda Zhang, Yan-Kai Tzeng, Konstantinos G. Lagoudakis, Hitoshi Ishiwata, Constantin Dory, Kevin A. Fischer, Yousif A. Kelaita, Shuo Sun, Peter C. Maurer, Kassem Alassaad, Gabriel Ferro, Zhi-Xun Shen, Nicholas A. Melosh, Steven Chu,

- and Jelena Vučković. Nanodiamond Integration with Photonic Devices. *Laser & Photonics Reviews*, 13(8):1800316, 2019.
- [200] Robert Raussendorf and Hans J. Briegel. A One-Way Quantum Computer. *Phys. Rev. Lett.*, 86(22):5188–5191, May 2001.
- [201] David Raventós, Tobias Graß, Maciej Lewenstein, and Bruno Juliá-Díaz. Cold bosons in optical lattices: A tutorial for exact diagonalization. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 50(11), 2017.
- [202] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73(1):58–61, 1994.
- [203] Alois Regensburger, Christoph Bersch, Mohammad-Ali Miri, Georgy Onishchukov, Demetrios N. Christodoulides, and Ulf Peschel. Parity-time synthetic photonic lattices. *Nature*, 488(7410):167–171, August 2012.
- [204] Christian Reimer, Yaowen Hu, Amirhassan Shams-Ansari, Mian Zhang, and Marko Loncar. High-dimensional frequency crystals and quantum walks in electro-optic microcombs. *arXiv:1909.01303*, September 2019. arXiv: 1909.01303.
- [205] Andreas Reiserer and Gerhard Rempe. Cavity-based quantum networks with single atoms and optical photons. *Reviews of Modern Physics*, 87(4):1379–1418, 2015.
- [206] C. Repellin, J. Léonard, and N. Goldman. Fractional Chern insulators of few bosons in a box: Hall plateaus from center-of-mass drifts and density profiles. *Physical Review A*, 102(6):1–12, 2020.
- [207] Eden Rephaeli and Shanhui Fan. Dissipation in few-photon waveguide transport [Invited]. *Photonics Research*, 1(3):110, 2013.
- [208] A. Ribeiro, A. Ruocco, L. Vanacker, and W. Bogaerts. Demonstration of a 4 x 4-port self-configuring universal linear optical circuit. *Optica*, 3(12):1348–1357, 2016.
- [209] Peter P. Rohde. Simple scheme for universal linear-optics quantum computing with constant experimental complexity using fiber loops. *Physical Review A - Atomic, Molecular, and Optical Physics*, 91(1):1–8, 2015.
- [210] Jonathan Roslund, Renne Medeiros de Araujo, Shifeng Jiang, Claude Fabre, and Nicolas Treps. Wavelength-multiplexed quantum networks with ultrafast frequency combs. *Nature Photonics*, 8(2):109–112, February 2014.

- [211] Jonathan E. Roth, Onur Fidaner, Rebecca K. Schaevitz, Yu-Hsuan Kuo, Theodore I. Kamins, James S. Harris, and David A. B. Miller. Optical modulator on silicon employing germanium quantum wells. *Optics Express*, 15(9):5851–5859, April 2007.
- [212] Nicholas J. Russell, Levon Chakhmakhchyan, Jeremy L. O’Brien, and Anthony Laing. Direct dialling of Haar random unitary matrices. *New Journal of Physics*, 19(3), 2017.
- [213] Alex Ryan, Riley Pinkerton, Lasya Marla, Eric Liu, Ruchi Pandya, Sean Dolan, Michael Goulet, Andrew Kennedy, Abhijit Puranam, Matt Kempster, Henry Swanson, Anne Dorsey, Dan Spaeth, Eric Theis, Zach Ontiveros, Adam Reeves, Jake Narkizian, Kevin King, Ryan Mentley, Kai van Ess, Noah Zempsky, Nick Cannavo, Alex Barata, Brandon Kirklen, Isaac Snyder, Trever Ewing, Harrison Wienerman, Robert Widmann, Kyle Jennings, Marq Salay, Hunter Wren, Josh Beal, Ruben Mayer, Jordan Dillard, Jack Ryan, Joseph Daniels, Padraig Feain, Justin Donnelley, Thomas Fuller, Josh Green, and Justin Greenberg. Providing social support for Ben during grad school. *Journal of Amazing Friends*, 1(1):1–99, 2022.
- [214] Polnop Samutpraphoot, Tamara Dordevic, Paloma L. Ocola, Hannes Bernien, Crystal Senko, Vladan Vuletic, and Mikhail D. Lukin. Strong Coupling of Two Individually Controlled Atoms via a Nanophotonic Cavity. *Phys. Rev. Lett.*, 124(6):063602, February 2020.
- [215] Maria Schuld, Alex Bocharov, Krysta Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *arXiv preprint*, 2018.
- [216] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, 122(4):040504, 2 2019.
- [217] Shankar Kumar Selvaraja, Peter De Heyn, Gustaf Winroth, Patrick Ong, Guy Lepage, Celine Cailler, Arnaud Rigny, Konstantin K. Bourdelle, Wim Bogaerts, Dries Van Thourhout, Joris Van Campenhout, and Philippe Absil. Highly uniform and low-loss passive silicon photonics devices using a 300mm CMOS platform. In *Optical Fiber Communication Conference (2014)*, Paper Th2A.33, page Th2A.33. Optical Society of America, March 2014.
- [218] K. T. Settaluri, C. Lalau-Keraly, E. Yablonovitch, and V. Stojanović. First Principles Optimization of Opto-Electronic Communication Links. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(5):1270–1283, May 2017.
- [219] Jeffrey M. Shainline, Sonia M. Buckley, Richard P. Mirin, and Sae Woo Nam. Superconducting Optoelectronic Circuits for Neuromorphic Computing. *Physical Review Applied*, 7(3):034013, March 2017.
- [220] Bhavin J. Shastri, Alexander N. Tait, Thomas Ferreira de Lima, Mitchell A. Nahmias, Hsuan-Tung Peng, and Paul R. Prucnal. Principles of Neuromorphic Photonics. *arXiv:1801.00016 [physics]*, pages 1–37, 2018.

- [221] J T Shen and Shanhui Fan. Coherent photon transport from spontaneous emission in one-dimensional waveguides. *Optics Letters*, 30(15):2001–2003, 2005.
- [222] Jung Tsung Shen and Shanhui Fan. Strongly correlated two-photon transport in a one-dimensional waveguide coupled to a two-level system. *Physical Review Letters*, 98(15):1–4, 2007.
- [223] Jung Tsung Shen and Shanhui Fan. Theory of single-photon transport in a single-mode waveguide. I. Coupling to a cavity containing a two-level atom. *Physical Review A*, 79(2):059904, 2009.
- [224] Jung Tsung Shen and Shanhui Fan. Theory of single-photon transport in a single-mode waveguide. II. Coupling to a whispering-gallery resonator containing a two-level atom. *Physical Review A*, 79(2):023838, 2009.
- [225] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Dirk Englund, and Marin Soljačić. Deep learning with coherent nanophotonic circuits. In *Summer Topicals Meeting Series, SUM 2017*, pages 189–190, 2017.
- [226] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11:441–447, June 2017.
- [227] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić. Supplementary information: Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, June 2017.
- [228] Tuo Shi, Tzung-I Su, Ning Zhang, Ching yin Hong, and Dong Pan. Silicon Photonics Platform for 400G Data Center Applications. In *Optical Fiber Communication Conference*, page M3F.4. Optical Society of America, 2018.
- [229] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4), 1995.
- [230] Marcus Silva. *Erasure Thresholds for Efficient Linear Optics Quantum Computation*. PhD thesis, University of Waterloo, 2004.
- [231] Immo Söllner, Sahand Mahmoodian, Sofie Lindskov Hansen, Leonardo Midolo, Alisa Javadi, Gabija Kiršanskė, Tommaso Pregnolato, Haitham El-Ella, Eun Hye Lee, Jin Dong Song, Søren Stobbe, and Peter Lodahl. Deterministic photon-emitter coupling in chiral photonic circuits. *Nature Nanotechnology*, 10(9):775–778, 2015.

- [232] Christoph Spengler, Marcus Huber, and Beatrix C. Hiesmayr. Composite parameterization and Haar measure for all unitary and special unitary groups. *Journal of Mathematical Physics*, 2012.
- [233] Justin B. Spring, Benjamin J. Metcalf, Peter C. Humphreys, W. Steven Kolthammer, Xian Min Jin, Marco Barbieri, Animesh Datta, Nicholas Thomas-Peter, Nathan K. Langford, Dmytro Kundys, James C. Gates, Brian J. Smith, Peter G.R. Smith, and Ian A. Walmsley. Boson sampling on a photonic chip. *Science*, 339(6121):798–801, 2013.
- [234] Dmitry V Strelakov, Christoph Marquardt, Andrey B Matsko, Harald G L Schwefel, and Gerd Leuchs. Nonlinear and quantum optics with whispering gallery resonators. *Journal of Optics*, 18(12):123002, nov 2016.
- [235] B. K. Stuhl, H.-I. Lu, L. M. Ayccock, D. Genkina, and I. B. Spielman. Visualizing edge states with an atomic Bose gas in the quantum Hall regime. *Science*, 349(6255):1514–1518, September 2015.
- [236] Aziza Suleymanzade, Alexander Anferov, Mark Stone, Ravi K. Naik, Andrew Oriani, Jonathan Simon, and David Schuster. A tunable high-Q millimeter wave cavity for hybrid circuit and cavity QED experiments. *Appl. Phys. Lett.*, 116(10):104001, March 2020. Publisher: American Institute of Physics.
- [237] Alexander N. Tait, Thomas Ferreira de Lima, Ellen Zhou, Allie X. Wu, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports*, 7(1):7430, August 2017.
- [238] Alexander N. Tait, Thomas Ferreira de Lima, Mitchell A. Nahmias, Heidi B. Miller, Hsuan-Tung Peng, Bhavin J. Shastri, and Paul R. Prucnal. Silicon Photonic Modulator Neuron. *Physical Review Applied*, 11(6):064043, June 2019.
- [239] S. Takeda and A. Furusawa. Toward large-scale fault-tolerant universal photonic quantum computing. *APL Photonics*, 4(6):1–13, 2019.
- [240] Rui Tang, Takuo Tanemura, Samir Ghosh, Keijiro Suzuki, Ken Tanizawa, Kazuhiro Ikeda, Hitoshi Kawashima, and Yoshiaki Nakano. Reconfigurable all-optical on-chip MIMO three-mode demultiplexing based on multi-plane light conversion. *Optics Letters*, 43(8):1798, 4 2018.
- [241] Qiskit Development Team. Qiskit: An open-source framework for quantum computing, 2021.
- [242] Daniel Tiarks, Steffen Schmidt-Eberle, Thomas Stolz, Gerhard Rempe, and Stephan Durr. A photon-photon quantum gate based on Rydberg interactions. *Nature Physics*, 15(2):124–126, February 2019. Number: 2 Publisher: Nature Publishing Group.

- [243] T. G. Tiecke, J. D. Thompson, N. P. de Leon, L. R. Liu, V. Vuletic, and M. D. Lukin. Nanophotonic quantum phase switch with a single atom. *Nature*, 508(7495):241–244, April 2014. tex.ids: tiecke_nanophotonic_2014-1 number: 7495 publisher: Nature Publishing Group.
- [244] Konstantin Tiurev, Pol Llopart Mirambell, Mikkel Bloch Lauritzen, Martin Hayhurst Appel, Alexey Tiranov, Peter Lodahl, and Anders Søndberg Sørensen. Fidelity of time-bin-entangled multiphoton states from a quantum emitter. *Phys. Rev. A*, 104(5):052604, November 2021.
- [245] Chiheb Trabelsi, Olexa Bilaniuk, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. *CoRR*, abs/1705.09792, 2017.
- [246] Edmondo Trentin. Networks with trainable amplitude of activation functions. *Neural Networks*, 14(4):471–493, May 2001.
- [247] Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi, and H. J. Kimble. Measurement of Conditional Phase Shifts for Quantum Logic. *Phys. Rev. Lett.*, 75(25):4710–4713, December 1995. Publisher: American Physical Society.
- [248] Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. Efficient decomposition of quantum gates. *Physical Review Letters*, 92(17):1–4, 2004.
- [249] Vivek Venkataraman, Kasturi Saha, and Alexander L. Gaeta. Phase modulation at the few-photon level for weak-nonlinearity-based quantum computing. *Nature Photonics*, 7(2):138–141, February 2013.
- [250] Jurgen Volz, Michael Scheucher, Christian Junge, and Arno Rauschenbeutel. Nonlinear * phase shift for single fibre-guided photons interacting with a single resonator-enhanced atom. *Nature Photon*, 8(12):965–970, December 2014.
- [251] Cheng Wang, Mian Zhang, Xi Chen, Maxime Bertrand, Amirhassan Shams-Ansari, Sethumadhavan Chandrasekhar, Peter Winzer, and Marko Lončar. Integrated lithium niobate electro-optic modulators operating at CMOS-compatible voltages. *Nature*, page 1, September 2018.
- [252] Guan Yu Wang, Qian Liu, Hai Rui Wei, Tao Li, Qing Ai, and Fu Guo Deng. Universal quantum gates for photon-atom hybrid systems assisted by bad cavities. *Scientific Reports*, 6:1–9, 2016.
- [253] Jianwei Wang, Fabio Sciarrino, Anthony Laing, and Mark G. Thompson. Integrated photonic quantum technologies. *Nature Photonics*, 14(5):273–284, 2020.
- [254] Kai Wang, Bryn A. Bell, Alexander S. Solntsev, Dragomir N. Neshev, Benjamin J. Eggleton, and Andrey A. Sukhorukov. Multidimensional synthetic chiral-tube lattices via nonlinear frequency conversion. *Light: Science & Applications*, 9(1):132, July 2020.

- [255] Eric W Weisstein. Gray Code, 1999.
- [256] C. M. Wilkes, X. Qiang, J. Wang, R. Santagati, S. Paesani, X. Zhou, D. A. B. Miller, G. D. Marshall, M. G. Thompson, and J. L. O'Brien. 60dB high-extinction auto-configured Mach-Zehnder interferometer. *Optics Letters*, 41(22):5318, 11 2016.
- [257] Ian A. D. Williamson, Tyler W. Hughes, Momchil Minkov, Ben Bartlett, Sunil Pai, and Shan-hui Fan. Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1):1–12, July 2019.
- [258] Ian A. D. Williamson, Tyler W. Hughes, Momchil Minkov, Ben Bartlett, Sunil Pai, and Shan-hui Fan. Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks. *arXiv preprint*, 3 2019.
- [259] Ian A. D. Williamson, Tyler W. Hughes, Momchil Minkov, Ben Bartlett, Sunil Pai, and Shan-hui Fan. Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1):1–12, 1 2020.
- [260] Martin Wimmer, Hannah M. Price, Iacopo Carusotto, and Ulf Peschel. Experimental measurement of the Berry curvature from anomalous transport. *Nature Physics*, 13(6):545–550, June 2017.
- [261] Zhaoju Yang, Eran Lustig, Gal Harari, Yonatan Plotnik, Yaakov Lumer, Miguel A. Bandres, and Mordechai Segev. Mode-Locked Topological Insulator Laser Utilizing Synthetic Dimensions. *Phys. Rev. X*, 10(1):011059, March 2020. Publisher: American Physical Society.
- [262] Jun-ichi Yoshikawa, Shota Yokoyama, Toshiyuki Kaji, Chanond Sornphiphatphong, Yu Shiozawa, Kenzo Makino, and Akira Furusawa. Invited Article: Generation of one-million-mode continuous-variable cluster state by unlimited time-domain multiplexing. *APL Photonics*, 1(6):060801, September 2016.
- [263] G. Yu, X. Zou, L. Zhang, Q. Zou, M. zheng, and J. Zhong. A low-noise high-gain transimpedance amplifier with high dynamic range in 0.13 μ m CMOS. In *2012 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*, pages 37–40, November 2012.
- [264] Luqi Yuan, Avik Dutt, Mingpu Qin, Shan-hui Fan, and Xianfeng Chen. Creating locally interacting Hamiltonians in the synthetic frequency dimension for photons. *Photon. Res.*, 8(9):B8–B14, September 2020.
- [265] Luqi Yuan, Avik Dutt, Mingpu Qin, Shan-hui Fan, and Xianfeng Chen. Creating locally interacting hamiltonians in the synthetic frequency dimension for photons. *Photon. Res.*, 8(9):B8–B14, Sep 2020.

- [266] Luqi Yuan, Qian Lin, Meng Xiao, and Shanhui Fan. Synthetic dimension in photonics. *Optica*, 5(11):1396–1405, November 2018.
- [267] Luqi Yuan, Qian Lin, Anwei Zhang, Meng Xiao, Xianfeng Chen, and Shanhui Fan. Photonic Gauge Potential in One Cavity with Synthetic Frequency and Orbital Angular Momentum Dimensions. *Physical Review Letters*, 122(8):083903, March 2019.
- [268] Luqi Yuan, Yu Shi, and Shanhui Fan. Photonic gauge potential in a system with a synthetic frequency dimension. *Opt. Lett.*, 41(4):741–744, February 2016.
- [269] Vivien Zapf, Marcelo Jaime, and C. D. Batista. Bose-Einstein condensation in quantum magnets. *Reviews of Modern Physics*, 86(2):563–614, 2014.
- [270] Yuan Zhan and Shuo Sun. Deterministic Generation of Loss-Tolerant Photonic Cluster States with a Single Quantum Emitter. *Physical Review Letters*, 125(22):223601, 2020.
- [271] Jingyuan Linda Zhang, Shuo Sun, Michael J. Burek, Constantin Dory, Yan Kai Tzeng, Kevin A. Fischer, Yousif Kelaita, Konstantinos G. Lagoudakis, Marina Radulaski, Zhi Xun Shen, Nicholas A. Melosh, Steven Chu, Marko Lončar, and Jelena Vučković. Strongly Cavity-Enhanced Spontaneous Emission from Silicon-Vacancy Centers in Diamond. *Nano Letters*, 18(2):1360–1365, 2018.
- [272] Yan Zhang, Qingyang Du, Chuangtang Wang, Takian Fakhrol, Shuyuan Liu, Longjiang Deng, Duanni Huang, Paolo Pintus, John Bowers, Caroline A. Ross, Juejun Hu, and Lei Bi. Monolithic integration of broadband optical isolators for polarization-diverse silicon photonics. *Optica*, 6(4):473, 2019.
- [273] Mengdi Zhao and Kejie Fang. InGaP quantum nanophotonic integrated circuits with 1.5% nonlinearity-to-loss ratio. *Optica*, 9(2):258–263, February 2022.
- [274] Huaixiu Zheng, Daniel J. Gauthier, and Harold U. Baranger. Waveguide-QED-based photonic quantum computation. *Physical Review Letters*, 111(9):090502, 2013.
- [275] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 2020.
- [276] Leimeng Zhuang, Chris G. H. Roeloffzen, Marcel Hoekman, Klaus-J. Boller, and Arthur J. Lowery. Programmable photonic signal processor chip for radiofrequency applications. *Optica*, 2(10):854, 2015.

- [277] K. Zyczkowski and M. Kus. Random unitary matrices. *Journal of Physics A: General Physics*, 1994.