Accurate Self-Configuration of Rectangular Multiport Interferometers

Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund

Accurate Self-Configuration of Rectangular Multiport Interferometers

Ryan Hamerly,[1, 2, *] Saumil Bandyopadhyay,[1] and Dirk Englund[1]

[1] *Research Laboratory of Electronics, MIT, 50 Vassar Street, Cambridge, MA 02139, USA*
[2] *NTT Research Inc., PHI Laboratories, 940 Stewart Drive, Sunnyvale, CA 94085, USA*
(Dated: July 22, 2022)

Multiport interferometers based on integrated beamsplitter meshes are widely used in photonic technologies. While the rectangular mesh is favored for its compactness and uniformity, its geometry resists conventional self-configuration approaches, which are essential to programming large meshes in the presence of fabrication error. Here, we present a configuration algorithm, related to the 2×2 block decomposition of a unitary matrix, that overcomes this limitation. Our proposed algorithm is robust to errors, requires no prior knowledge of the process variations, and relies only on external sources and detectors. We show that self-configuration using this technique reduces the effect of fabrication errors by the same quadratic factor observed in triangular meshes. This relaxes a significant limit to the size of multiport interferometers, removing a major roadblock to the scaling of optical quantum and machine-learning hardware.

I. INTRODUCTION

Large-scale programmable photonic circuits are the cornerstone of many emerging technologies, including quantum computing [1, 2], machine learning acceleration [3–5], and microwave photonics [6, 7]. One such circuit—the universal multiport interferometer, which functions as a linear optical input-output device with a programmable transfer matrix (Fig. 1)—is of special importance due to its generality and broad range of applications [8]. The most scalable designs involve meshes of Mach-Zehnder interferometers (MZIs): while the triangular RECK mesh (Fig. 1(a)) was initially employed and is straightforward to configure [9], more recently work has shifted to the CLEMENTS rectangle (Fig. 1(b)), which offers clear advantages of compactness, path-length uniformity, and reduced sensitivity to loss [10]. Much recent study has focused on scaling [11, 12] and optimizing [13–16] MZI meshes based on the CLEMENTS design.

A major challenge to scaling MZI meshes is the presence of component errors due to fabrication imperfections. Errors cause each MZI transfer matrix to deviate from its programmed value (Fig. 1(c)); since the overall circuit is a cascade of MZIs with $O(N)$ depth, the total error in the transfer matrix scales as $O(\sqrt{N})$, where N is the circuit size (assuming uncorrelated errors). At large mesh sizes, these errors (if left uncorrected) place unreasonable constraints on fabrication tolerances, ultimately limiting the scaling of multiport interferometers. Error-correction techniques are therefore critical for large-scale programmable photonics. Global optimization [13–15] and in-situ training [17] are promising in principle, but computationally inefficient and the optimization result is device-specific. Local per-MZI correction is also effective, but requires prior characterization of the device errors [18, 19]. A number of efficient "self-configuring" algorithms have been proposed [20–22], but these gen-erally only work for triangular meshes and require large numbers of internal power monitors [23], a significant addition in hardware complexity. Recently, we proposed an efficient error-correction algorithm that does not require internal detectors or accurate pre-characterization [24]. However, this algorithm only works for triangular (i.e. RECK) meshes, which excludes the more efficient CLEMENTS design.

In this article, we present a self-configuring strategy that is naturally adapted to the CLEMENTS design and requires no additional hardware complexity beyond external sources and (coherent) detectors. This algorithm proceeds by configuring the diagonals of the mesh, starting at the corners, in a manner that progressively zeroes out the elements of a target matrix through a sequence of Givens rotations. Numerical experiments on imperfect MZI meshes show that our algorithm is stable and robust,
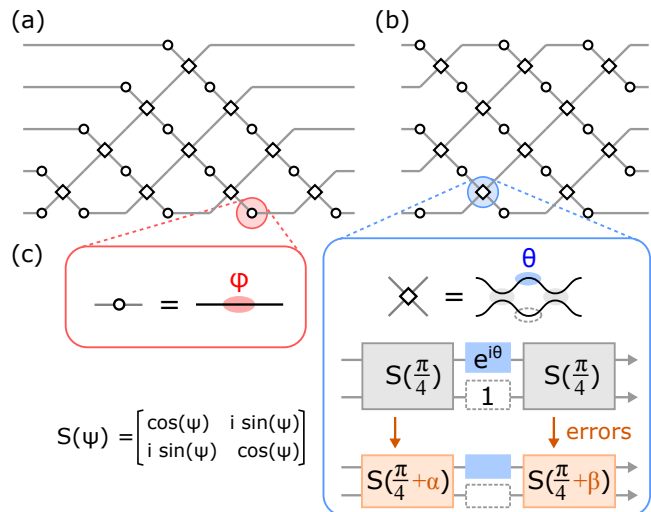

FIG. 1. (a) RECK and (b) CLEMENTS 5×5 MZI meshes. (c) Tunable building blocks: MZI and phase shifter. Effect of errors on MZI transfer function.
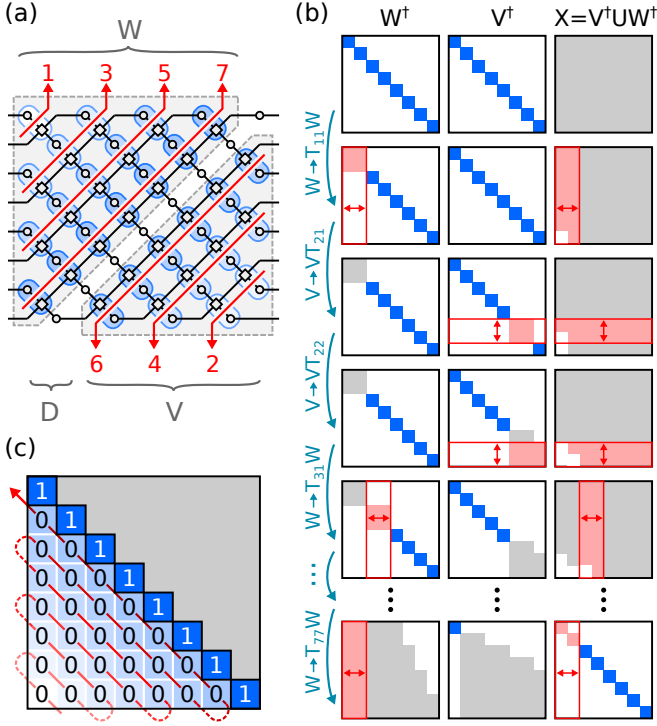
* rhamerly@mit.edu

FIG. 2. (a) CLEMENTS mesh and order of MZI configuration. (b) Building up the matrices V, W in a sequence of Givens rotations to diagonalize $X = V^\dagger U W^\dagger$. (c) Order of matrix elements zeroed by the procedure.

and reduces errors by at least a quadratic factor (while sufficiently small errors are corrected exactly), consistent with the behavior observed for triangular meshes [24]. This significantly relaxes the scaling constraints posed by imperfections in realistic MZI meshes. As an example, we consider the application to optical neural networks and show how this correction scheme provides a path to overcome the no-go results of Ref. [25].

II.   ALGORITHM

Our procedure is based on the diagonalization of a unitary using 2×2 Givens rotations [10], which we review here for clarity. Following Fig. 2(a), divide the mesh along the rising diagonal so that its transfer matrix becomes $U = VDW$, where D is a phase screen and V and W represent the upper and lower triangles, given by:

$$V = (T_{21}T_{22})(T_{41}\ldots T_{44})\ldots(T_{N-2,1}\ldots T_{N-2,N-2})$$
$$W = (T_{N-1,N-1}\ldots T_{N-1,1})\ldots(T_{33}\ldots T_{31})(T_{11}) \quad (1)$$

where T_{mn} is the 2×2 block unitary corresponding to the nth crossing (MZI / phase-shift pair) of the mth diagonal. Following the order in the figure, we "build up" the matrices (V, W) one block at a time while keeping track of $X = V^\dagger U W^\dagger$. Fig. 2(b) shows the first few steps

of this process. Updates to $W \to T_{mn}W$ right-multiply $X \to X T_{mn}^\dagger$, and the phases of T_{mn} are chosen to zero an element in the lower-left corner of X. Likewise, updates to $V \to V T_{mn}$ left-multiply $X \to T_{mn}^\dagger X$. Following the order in Fig. 2(c), this procedure zerores all elements in the lower-left triangle of X, enforcing diagonality. The remaining phases are read off from the diagonal elements.

While this procedure correctly sets the phases (θ, ϕ) for an ideal CLEMENTS mesh, it does not work in the presence of errors because the relationship between (θ, ϕ) and T_{mn} also depends on the splitting angle imperfections (α, β) (Fig. 1(c)), which are unknown. Here, we describe the procedure for programming CLEMENTS in the presence of errors:

1. Initialize the MZIs to approximate the cross state ($\theta = 0$). An ideal cross state is not possible with errors, but an approximation will be fine.

2. Configure the crossings in the order given by Fig. 2(a). For each crossing in W [resp. V]:

   (a) Perform the Givens rotation $X \to X T_{mn}^\dagger$ [resp. $X \to T_{mn}^\dagger X$] that zeroes the next element of X_{ij} in the sequence Fig. 2(c) (note $(i,j) \neq (m,n)$).
   (b) Update $W \to T_{mn}W$ [resp. $V \to V T_{mn}$].
   (c) Send input $\vec{a}_{\text{in}} = \vec{w}_j^*$ (the jth column of $W^\dagger$) into the device, Fig. 3. The output $\vec{a}_{\text{out}}(\theta, \phi)$ depends on the phases being configured. Set (θ, ϕ) to zero the inner product $\langle \vec{v}_i | \vec{a}_{\text{out}}(\theta, \phi) \rangle$, where $\vec{v}_i$ is the ith column of V.
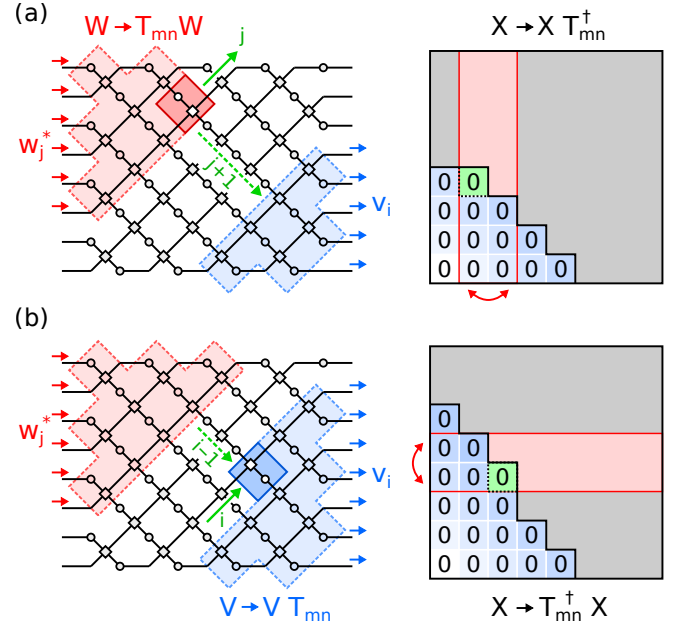

FIG. 3. Programming a physical CLEMENTS mesh in the presence of errors. (a) Procedure to set phases for a crossing in W. (b) Procedure for V.

3. Finally, set the diagonal: for each i, inject $\vec{a}_\text{in} = w_i^*$ and adjust ϕ to satisfy $\arg(\langle\vec{v}_i|\vec{a}_\text{out}(\phi)\rangle) = \arg(X_{ii})$.

Of the main loop (Step 2), parts (a-b) are just a restatement of the CLEMENTS factorization [10]. However this is performed merely to keep track of matrices (V, W); we do not use the (θ, ϕ) provided by the algorithm. Instead, Step 2(c) uses physical measurements find the correct phases in the presence of hardware errors. This step, which amounts to zeroing matrix element $X_{ij} = v_i^* U w_j^*$ of the physical hardware, is visualized in Fig. 3:

1. When configuring an W (Fig. 3(a)), we input $\vec{a}_\text{in} = \vec{w}_j^*$ and program (θ, ϕ) to direct all the light to intermediate output j (solid arrow), zeroing the power that goes to $j+1$ (dashed arrow). We do not have access to these intermediate outputs, but $j+1$ connects to the input i of V; therefore $\langle\vec{v}_i|\vec{a}_\text{out}\rangle$ is a valid proxy for this field and zeroing it correctly configures the red block to match W.

2. When configuring V (Fig. 3(b)), we want $\vec{v}_i$ to be the output from light at intermediate port i (solid arrow); therefore, the output from light at port $i-1$ (dashed arrow) should be orthogonal to $\vec{v}_i$. We excite this field by inputting $\vec{w}_j^*$; any errors in the unconfigured mesh will hit downstream inputs $k < i-1$, but will not pollute input i. Therefore zeroing $\langle\vec{v}_i|\vec{a}_\text{out}\rangle$ correctly configures the blue block of the mesh to match V, provided W is properly configured.

In both cases, the nulling signal $\langle\vec{v}_i|\vec{a}_\text{out}\rangle$ arises from light transmitted down the dashed diagonals in Fig. 3; a (near) cross-state initialization is needed to maintain a non-negligible transmitted power. As we discuss later, this initialization is limited by random phase shifts in the MZIs, which can be fixed with a one-time calibration. It is also relatively straightforward to extend our proposed scheme to triangular meshes, as described in Appendix A.

III. RESULTS

To test our algorithm, we performed numerical experiments on CLEMENTS meshes of size up to 128×128. The algorithms were implemented in PYTHON and are available as part of the MESHES package [26]. We consider an error model based on imperfections in splitter angles α, β (Fig. 1(c)), as these are the dominant effect of fabrication error in MZI-based silicon photonic circuits. For simplicity, consider the case of uncorrelated Gaussian errors so that the error magnitude can be characterized by a single variable $\sigma = \langle\alpha\rangle_\text{rms} = \langle\beta\rangle_\text{rms}$ (the case of correlated errors is treated in Appendix B; the qualitative results are the same because most correlations cancel out in ensemble averaging [24]). Target matrices are sampled uniformly over the Haar measure [27, 28]. As a figure of merit, we consider the normalized matrix error

$\mathcal{E} = \langle\|\Delta U\|\rangle_\text{rms}/\sqrt{N}$. For unitary matrices, $\mathcal{E} \in [0, 2]$ corresponds to the average relative error of a given matrix element U_{ij}.

In the uncorrected case, each MZI introduces a mean error $\langle\|\Delta U\|\rangle_\text{rms} = \sqrt{2}\sigma$. These errors add in quadrature, leading to an overall normalized error $\mathcal{E}_0 = \sqrt{2N}\sigma$, which grows with mesh size. This is understandable given that a circuit depth that grows as $O(N)$, with each layer contributing $O(\sigma)$ error and the layers adding in quadrature. Fig. 4(a) plots the error as a function of σ for a 64×64 mesh, both without error correction (red) and with our algorithm (blue). The algorithm always improves the matrix fidelity, but there are two distinct regimes: for small σ, the error approaches machine precision, as errors can be corrected exactly. On the contrary, for large σ, the corrected error asymptotes to a finite value:

$$\mathcal{E}_c = \sqrt{2/3}\, N\sigma^2 = \frac{1}{\sqrt{6}}\mathcal{E}_0^2 \qquad (2)$$

This form can be derived rigorously from the distribution of MZI splitting angles for Haar-sampled unitary matrices [24, 29], where errors arise solely from MZIs whose target splitting ratios cannot be realized in the imperfect hardware. Since $\mathcal{E}_c \propto \mathcal{E}_0^2$, we can say that self-calibration
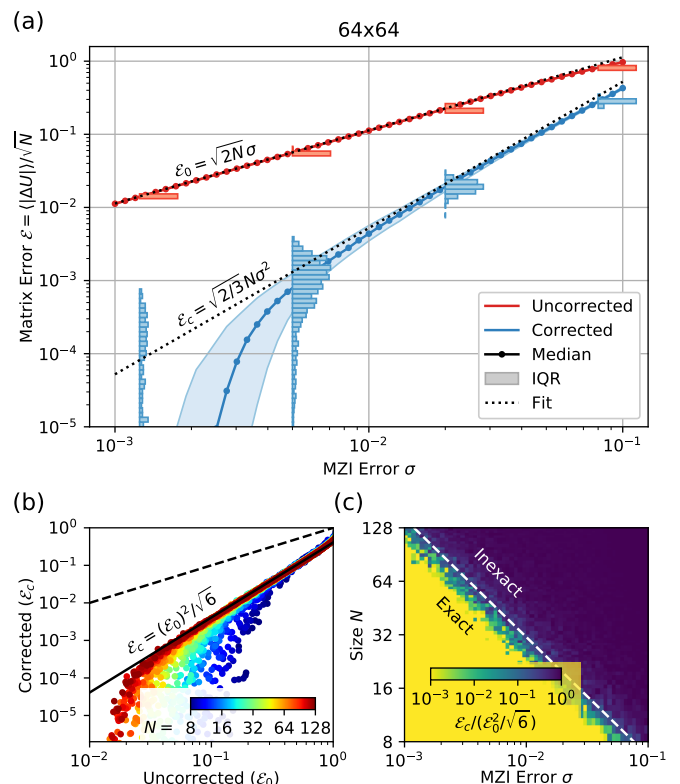

FIG. 4. (a) Corrected and uncorrected matrix error as a function of σ, 64×64 CLEMENTS mesh. (b) Scaling with mesh size, showing the quadratic suppression of errors due to correction. (c) Boundary between the regimes of exact and inexact error correction.
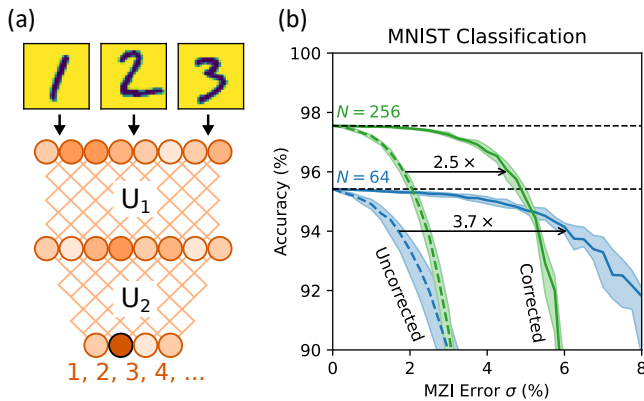
FIG. 5. (a) 2-layer DNN for MNIST classification where synaptic connections are represented by CLEMENTS meshes. (b) Simulated classification accuracy as a function of network size N and hardware MZI error σ.

FIG. 6. Comparison of hardware error types: $\mathcal{E}$ plotted against mesh size. Here $\sigma_{\mathrm{bs}} = 2\%$, $\sigma_{\mathrm{ph}} = 0.2$, $\sigma_\ell = 5 \times 10^{-4}$, and $\sigma_{\mathrm{dac}} = 2\pi/\sqrt{12} \times 2^{-B}$ with $B = 12$ bits of precision. See Appendix D for derivation.

leads to a *quadratic* suppression of errors: the smaller the initial error, the greater the benefit of error correction.

Fig. 4(b-c) tests the scalability of the algorithm by varying the mesh size N. Fig. 4(b) shows that, as expected, as the mesh size grows, the corrected error asymptotes to the quadratic factor Eq. (2). By plotting $\mathcal{E}_c/(\mathcal{E}_0^2/\sqrt{6})$, Fig. 4(c) shows the boundary between the exactly correctable small-error regime ($\mathcal{E}_c = 0$) and the inexact large-error regime where Eq. (2) holds. The regimes meet where the coverage cov(N) of $U(N)$ dips below unity [29]; we have previously shown that cov$(N) \sim e^{-N^3\sigma^2/3}$ [24]; therefore the boundary lies at roughly $N^3\sigma^2 = 3$ (dashed curve in Fig. 4(c)).

To illustrate the benefits of this error reduction, we consider as an example deep neural network (DNN) inference on optical hardware. DNNs process data in a sequence of layers, each consisting of (linear) synaptic connections and (nonlinear) neuron activation (Fig. 5(a)). One exciting possibility is to use photonics to accelerate this process: encode the input in optical amplitudes, use a programmable MZI mesh to implement the synaptic weights, and perform the activations with an all-optical (or electro-optic) nonlinearity [3]. A major challenge is that useful learning tasks require large mesh sizes ($N > 100$), which are particularly susceptible to fabrication error; a recent study showed that accurate DNN inference might require unrealistic process tolerances in the hardware [25]. This challenge has spurred investigations into alternative proposals, which have their own limitations [4, 5, 30].

Fig. 5(b) illustrates the advantage of error correction in an MZI-mesh DNN accelerator. Here, input images are preprocessed by a Fourier transform and fed into a two-layer DNN, with electro-optic neuron activations designed to approximate a complex modReLU [31, 32]. Models with inner layer sizes $N = 64$ and $N = 256$ were trained on the MNIST digit dataset [33] using the NEU-
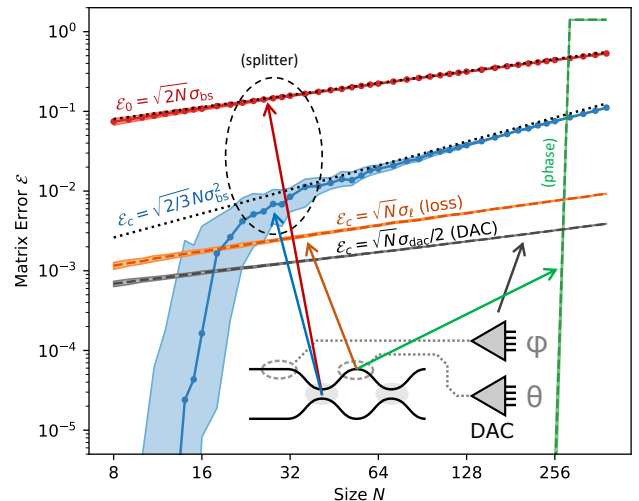
ROPHOX package [34]. Details and code are provided in Appendix C and the Supplementary Material [35], respectively. These pre-trained models were then simulated numerically on imperfect CLEMENTS meshes to calculate the classification accuracy. Applying our correction algorithm increases the error tolerance of these DNNs by over $2\times$. Note that directional couplers in silicon typically exhibit $\sigma \approx 2\%$ [36]; in this regime, the uncorrected DNNs show significant degradation, while error correction restores them to their canonical accuracy. Error correction may even allow the use of broad-band multi-mode interference couplers, which typically exhibit larger hardware errors.

IV. DISCUSSION

Beyond splitting-ratio errors, realistic meshes must contend with a range of additional imperfections, including undesired phase shifts, unbalanced losses, and limited programming precision. We analyze these error sources in Appendix D, with the main results shown in Fig. 6. Phase-shift and loss errors introduce a random transmission coefficient $e^{i\psi - \ell/2}$ to each waveguide segment, where $\langle\psi\rangle_{\mathrm{rms}} = \sigma_{\mathrm{ph}}$, $\langle\ell\rangle_{\mathrm{rms}} = \sigma_\ell$. If known, phase errors can be corrected exactly with a one-time calibration. However, even without calibration, self-configuration still succeeds provided the errors are small enough, and we observe a sharp error-correction threshold of $\sqrt{N}\sigma_{\mathrm{ph}} \lesssim O(1)$. This condition is related to the need to initialize the MZIs close enough to the near-cross state that a significant fraction of the light can propagate down the nulling diagonal (dashed green line in Fig. 3) to give a non-negligible

signal $\langle \vec{v}_i | \vec{a}_{\text{out}} \rangle$.

Unlike phase errors, unbalanced losses and programming (DAC) errors are not correctable, as evidenced by the observed behavior $\mathcal{E}_c = \sqrt{N}\sigma_\ell$ and $\sqrt{N}\sigma_{\text{dac}}/2$, which follows the same $\sqrt{N}$ scaling observed for uncorrected errors. Loss imbalance is limited by statistical fluctuations in surface roughness; an analysis based on Ref. [37] gives $\sigma_\ell \approx \pi^{-1}\alpha_{\text{wg}}\sqrt{L_c L \log(4L/L_c)}$, where α_{wg} is the waveguide loss, L is the waveguide length, and the scatterer size $L_c = \lambda/n_0$ depends on wavelength and cladding refractive index. For a $L = 200$ μm and $\alpha_{\text{wg}} = 2$ dB/cm, one calculates $\sigma_\ell = 5 \times 10^{-4}$, which is close to observed wafer-scale loss variations [38]. DAC error is set by the average truncation to B bits of a signal in the range $[0, 2\pi]$: $\sigma_{\text{dac}} = 2\pi/\sqrt{12} \times 2^{-B}$. While neither loss or DAC error are correctable, Fig. 6 clearly shows that these effects are about an order of magnitude smaller than the typical splitter errors, even after correction. These errors will only become relevant if the (correctable) unitary imperfections can be substantially reduced, e.g. with post-fabrication trimming [39] or "perfect" MZI-doubled structures [40].

Our algorithm runs in $O(N^2)$ steps and requires only $O(N^3)$ FLOPs of computation. This figure is at least an order of magnitude faster than in-situ or direct numerical optimization schemes (Appendix E). While local error correction [18] and parallelized progressive methods [16] are faster still, these schemes respectively require calibration and internal detectors, a major challenge to their deployment on large-scale photonic meshes.

We have proposed a self-configuration technique for rectangular MZI meshes. Our technique requires only external sources and (coherent) detectors and does not rely on an accurate characterization of device errors. This method is based on the diagonalization of a unitary matrix by Givens rotations, with a specific set of measurements performed to ensure that the Givens rotations are properly implemented in the hardware. For sufficiently small hardware errors, our approach leads to perfect realization of the target matrix. For large errors, it achieves the same quadratic reduction $\mathcal{E} \rightarrow \mathcal{E}^2/\sqrt{6}$ observed for local correction algorithms [18] and self-configuration on triangular meshes [24]. As a target application, we considered optically accelerated DNNs and showed that the proposed technique increases their robustness to hardware error, particularly in the critical region around $\sigma \approx 2\%$ characteristic of directional couplers in silicon.

One open question is increasing the robustness of error correction to non-unitary errors (unbalanced losses), as many emerging photonic devices often have undesired state-dependent loss [41–43]. In addition, extensions of the algorithm to more recently developed mesh geometries [44–48] may prove useful, as some geometries are less sensitive to hardware error and may be easier to scale up to large dimensions.

S.B. is supported by an NSF Graduate Research Fellowship under grant no. 1745302 and the Air Force

Appendix A: Triangular Meshes

The key to self-configuring the CLEMENTS mesh [10] was realizing that it could be divided into two triangles with a phase screen in the center: $U = VDW$. By configuring one diagonal at a time, alternating between V and W, we could zero all the elements of the target matrix $X = V^\dagger U W^\dagger$, thus realizing the desired unitary. This procedure is simplified when configuring RECK [9], which can be expressed in terms of a single triangle:

$$U = D\Big(\underbrace{\prod_{mn} T_{mn}}_{W}\Big) \quad \text{or} \quad \Big(\underbrace{\prod_{mn} T_{mn}}_{V}\Big)D \qquad \text{(A1)}$$
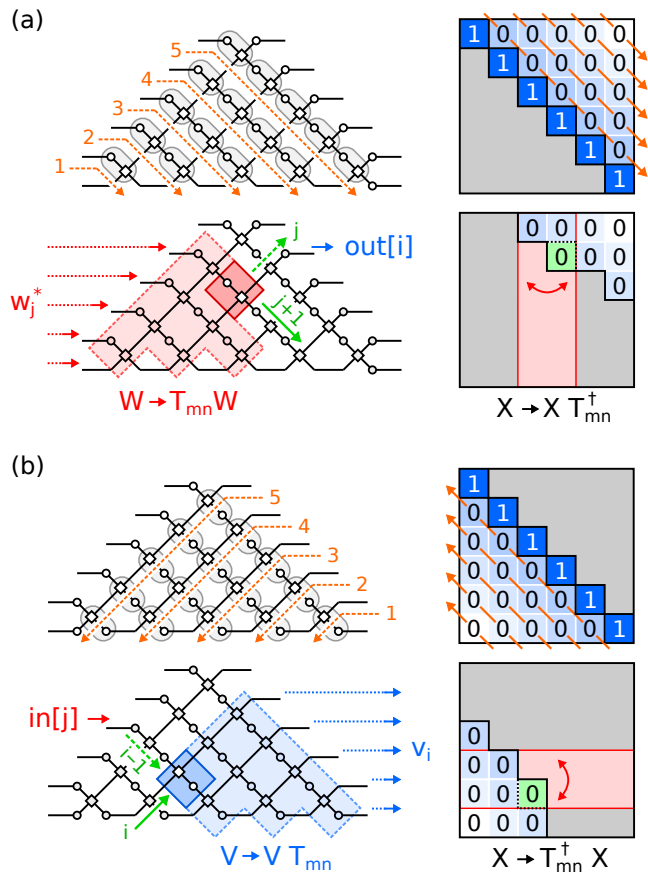

FIG. 7. Self-configuration procedure for the triangular RECK mesh. The procedure depends on whether the mesh layout has (a) an output phase screen, or (b) an input phase screen.

The first case $U = DW$ corresponds to a mesh with an output phase screen (Fig. 7(a)), while the mesh for $U = VD$ has an input phase screen (Fig. 7(b)). The algorithm in Sec. II simplifies because we may substitute $V = I$ [resp. $W = I$]. This leads to two cases:

1. For the first case (Fig. 7(a)), we work downstream from the leftmost MZI. For example, we can work down the falling diagonals, with the order given in the figure. Each step adds an MZI to $W \to T_{mn}W$, which updates the target matrix $X \to XT_{mn}^\dagger$ to zero an element in the upper triangle. With input field $a_{\rm in} = w_j^*$ (the $j^{\rm th}$ column of $W^\dagger$), the parameters (θ, ϕ) are set to zero the field at output i, where (i, j) is the index of the element X_{ij} being zeroed.

2. For the second case (Fig. 7(b)), we work upstream from the rightmost MZI, which performs the update $V \to T_{mn}V$, $X \to T_{mn}^\dagger X$. With light sent into the $i^{\rm th}$ input, (θ, ϕ) are chosen to zero the dot product between the output field and v_i (the $i^{\rm th}$ column of V).

After the crossings are configured, the phases of D can be obtained by inspection.

The MZI order in Fig. 7 is not unique. Any order that preserves causality (the set of configured and unconfigured MZIs must be causally separated) will produce a valid self-configuration.

Like the configuration of the CLEMENTS rectangle, the procedure above is designed, at each time step, to properly set the configured MZIs in the red [resp. blue] block to realize W [resp. V]. The procedure in Fig. 1(a) is closely related to the Reversed Local Light Interference Method (RELLIM) [22]. However, there are two important differences. First, RELLIM inputs columns of $U^\dagger$, while our scheme inputs columns of $W^\dagger$. As the MZIs are configured, W changes. Second, RELLIM assumes the existence of internal power detectors after each MZI (or the ability to pre-calibrate the downstream mesh so that internal fields can be read off from the outputs). By inputting a row of the target $W^\dagger$, light will exit the red block along only the $j^{\rm th}$ and $(j+1)^{\rm th}$ channels (provided all upstream MZIs are correctly set). The former, which is set to zero by varying (θ, ϕ), is read off directly from the $i^{\rm th}$ output.

The procedures in Fig. 7(a-b) are reciprocal to each other and can also be related to the "ratio method" described in Ref. [24], since the process of zeroing matrix elements never depends on absolute amplitudes, only on their ratios. Moreover, the ratio method is also robust in the presence of large errors and achieves the same quadratic error reduction. However, the when configuring each MZI, the ratio method required a sweep of the upstream phase shifter in order to subtract the common amplitude $\vec{a}$, a procedure that is not necessary here.

Fig. 8 plots the matrix error $\mathcal{E} = \|\Delta U\|/\sqrt{N}$ as a function of MZI error σ for the RECK mesh. The result is
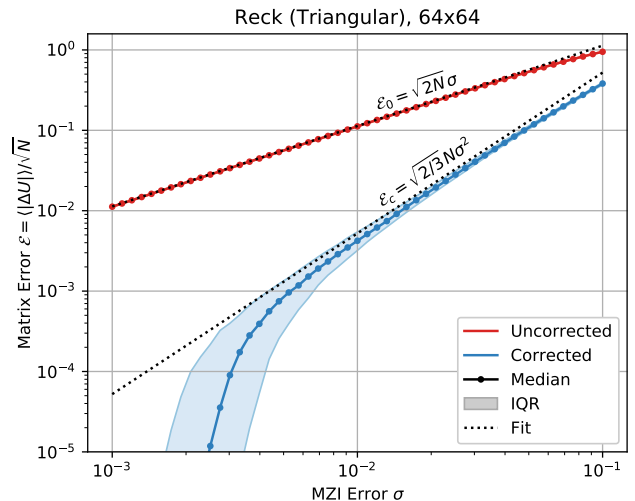

FIG. 8. Corrected and uncorrected matrix error as a function of MZI error σ, 64×64 RECK mesh.

identical to the case of the CLEMENTS mesh, Fig. 4, following the curves $\mathcal{E}_0 = \sqrt{2N}\sigma$, $\mathcal{E}_c = \sqrt{2/3N}\sigma^2$ predicted by theory [24]. Over the Haar measure, the distribution of MZI splitting angles is the same for both mesh geometries up to a reordering of the MZIs [29]; therefore, this correspondence is unsurprising.

The self-configuration procedures for RECK and CLEMENTS can be mapped to a general-purpose subroutine that self-configures any MZI mesh of the form $U = VDW$, provided that the geometry admits a matrix diagonalization by way of Givens rotations. This algorithm has been implemented in PYTHON with NUMBA extensions for numerical efficiency, and is available as part of the MESHES package [26]

Appendix B: Correlated Errors

In realistic MZI meshes, the splitter errors α_n, β_n will be strongly correlated, since the process variations that lead to errors (waveguide thickness and spacing, partial etch depth, slab height) all have correlation lengths much longer than the size of an MZI. In general, the matrix error $\langle \|\Delta U\|^2 \rangle$ will depend both on the error amplitudes $(\langle \alpha_n^2 \rangle, \langle \beta_n^2 \rangle)$ as well as their correlations $(\langle \alpha_m \alpha_n \rangle$, etc.). For an individual matrix U, the dependence on correlations can be significant. However, we have previously shown that, in an *ensemble average* of matrices uniformly sampled over the Haar measure, this dependence becomes very small for most inter-MZI correlations because of the random phase shifts between pairs of MZIs [24, Appendix A]. Only intra-MZI correlations $\langle \alpha_n \beta_n \rangle$ have a significant effect on the ensemble-averaged matrix error.

Consider the extreme case of full correlation $\alpha_n = \beta_n = \mu$. This case is realized, for instance, when the
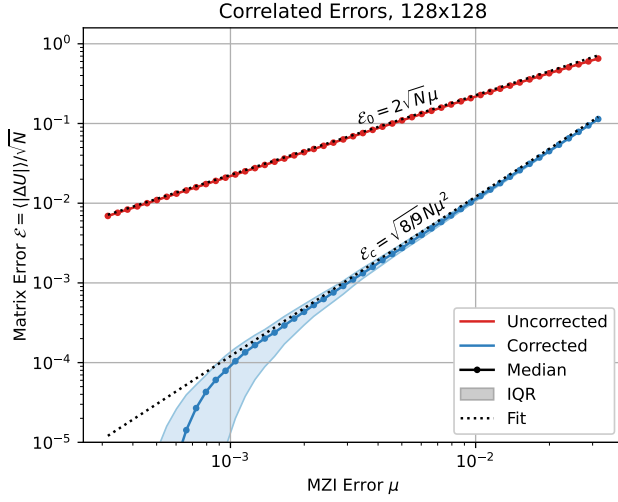
FIG. 9. Corrected and uncorrected matrix error as a function of correlated MZI error amplitude μ, 128×128 CLEMENTS mesh.

dominant error source arises from operating the mesh away from the coupler design wavelength. In Ref. [24], the coverage, uncorrected error, and corrected error are calculated to be:

$$\text{cov}(N) = e^{-(2/3)N^3\mu^2} \tag{B1}$$

$$\mathcal{E}_0 = 2\sqrt{N}\mu \tag{B2}$$

$$\mathcal{E}_c = \sqrt{8/9}N\mu^2 \tag{B3}$$

Fig. 9 plots $\mathcal{E}_0$ and $\mathcal{E}_c$ against the error amplitude μ for a 128×128 CLEMENTS mesh. Small residual dependences on correlation (proportional to $\theta_m\theta_n$) lead to a slight deviation in $\mathcal{E}_0$, while the theoretical curve for $\mathcal{E}_c$ matches very accurately. The behavior is qualitatively very close to that in the uncorrelated case.

Appendix C: Neural Network Model

The optical neural network model is based on the architecture described in Ref. [16]. Images from the MNIST digit dataset are preprocessed with a Fourier transform, which is cropped to a $\sqrt{N} \times \sqrt{N}$ window, where N is a model parameter that quantifies the size of the neural network. The light from this window (N input neurons) is fed into a two-layer optically accelerated DNN. This DNN consists of a single inner layer and two $N \times N$ unitary circuits, represented by CLEMENTS meshes (Fig. 10(a)).

The activation function at the inner layer is realized with an electro-optic nonlinearity: a fraction of each output field is fed into a detector that drives a Mach-Zehnder modulator, while the remaining light passes through the
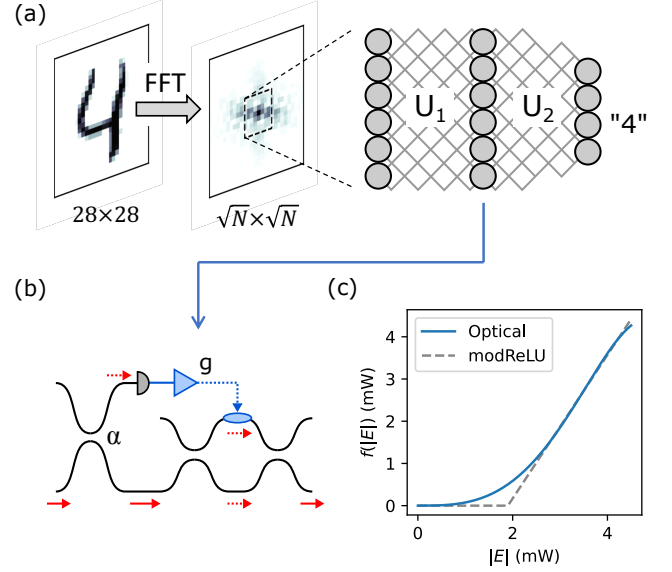

FIG. 10. Neural network model. (a) Dataflow: a 28×28 image is Fourier transformed and cropped to an $N \times N$ window; the complex amplitudes are then fed through a two-layer ONN. (b) Electro-optic nonlinearity implementing Eq. (C1). (c) Input-output relation for the nonlinearity, which approximates a complex modReLU.

modulator [31]. This is shown in Fig. 10(b), and implements the activation function:

$$f(E) = \sqrt{1-\alpha}\, e^{-i(g|E|^2+\phi-\pi)/2} \cos\left(\tfrac{1}{2}(g|E|^2+\phi)\right) \tag{C1}$$

where α is the power tap fraction, g is the modulator phase induced per unit optical power, and ϕ is the phase in the absence of power. Here, we choose $\alpha = 0.1$, $g = \pi/20$, and $\phi = \pi$, which causes $f(E)$ to approximate the form of a complex modReLU [49] in the right power regime (Fig. 10(c)).

Models of sizes $N = 64$ and $N = 256$ were trained using the NEUROPHOX package [34]. Code and model parameters are provided in the Supplementary Material [35].

Appendix D: Additional Hardware and Calibration Errors

We find that beamsplitter imperfections are the dominant source of error in most situations. Additional hardware imperfections include: (1) random phase shifts, which are perfectly correctable provided the error is below a given threshold, and (2) unbalanced loss and finite programming precision, which are in general uncorrectable. The sections below analyze these effects in detail, concluding that for realistic hardware, errors in the former category are well below the threshold for perfect correction, while errors in the latter category are small enough that they can be ignored.

1. Perfectly Correctable Errors: Phase Shifts

While multiport interferometers are length-balanced to ensure that all paths accumulate the same phase shift (in the absence of heater power), in practice there will be phase errors due to localized imperfections. If these errors are known, they can be calibrated away and absorbed into the programmable phase shifts (θ, ϕ) of each MZI. Various calibration procedures have been reported in the literature [50, 51]. However, even without pre-calibration, the method reported in this paper can exactly correct for phase-shifter errors provided that their magnitude falls below a given threshold.

To start, consider an error model with random phase shifts on both the internal and external arms, i.e. a given MZI has the 2×2 unitary

$$T(\theta,\phi) \rightarrow$$
$$S(\tfrac{\pi}{4}+\beta)\begin{bmatrix} e^{i(\theta+\psi_1)} & 0 \\ 0 & e^{i\psi_2} \end{bmatrix} S(\tfrac{\pi}{4}+\alpha)\begin{bmatrix} e^{i(\phi+\psi_3)} & 0 \\ 0 & e^{i\psi_4} \end{bmatrix}$$
$$(D1)$$

where errors are uncorrelated and Gaussian: $(\alpha, \beta) \sim N(0, \sigma_{bs})$ and $\psi_i \sim N(0, \sigma_{ph})$. In Fig. 11, we numerically perform self-configuration on such faulty CLEMENTS meshes, varying both mesh size N and phase error σ_{ph}, for the cases of ideal ($\sigma_{bs} = 0$) and non-ideal ($\sigma_{bs} = 0.03$) splitters. In the absence of error correction, the matrix error is calculate to be $\mathcal{E}_0 = \sqrt{2N(\sigma_{ph}^2+\sigma_{bs}^2)}$, which is dominated by the phase shifts when $\sigma_{ph} > \sigma_{bs}$. With self-configuration, we see a sharp phase transition between a regime where error correction succeeds ($\mathcal{E}_c = \sqrt{2/3N\sigma_{bs}^2}$) and fails ($\mathcal{E}_c \approx 1$). This threshold scales as $\sigma_{ph} \sim 3/\sqrt{N}$, and varies only slightly with the splitter error σ_{bs}.

This threshold arises from imperfect initialization of the mesh, as phase errors cause the initialized MZIs to deviate from the cross state. While the algorithm does not rely on these MZIs being exact crossings, the measurement steps consist of nulling optical signals that depend on fields propagating down a long chain of cross-like MZIs (dashed lines in Fig. 3). As the (amplitude) transmission of an imperfect crossing goes as $t_{MZI} \sim e^{-(\psi_1-\psi_2)^2/2}$ for small errors, the average transmission after N stages (there are approximately N crossings during the initial self-configuration steps) is $t = e^{-N\sigma_{ph}^2}$.

The algorithm fails for $\sigma_{ph} > 4/\sqrt{N}$ even with perfect 50:50 splitters because $t < 10^{-7}$, which is below 32-bit machine precision. In practice, zeroing to such weak signals is impractical in the presence of noise, so realistically one requires $N\sigma_{ph}^2 \lesssim 2$ in order to maintain a reasonable amplitude for the nulling signal $t \gtrsim 0.1$. This is a factor of 2–3 below the thresholds in Fig. 11, so self-configuration should work to high accuracy.

In a recent 32×32 SOI switch network, Suzuki et al. measured a trimming power variability ($\sigma = 1.52$ mW) an order of magnitude lower than the switching power ($P_\pi = 18.1$ mW) [52]. Putting these together, we cal-
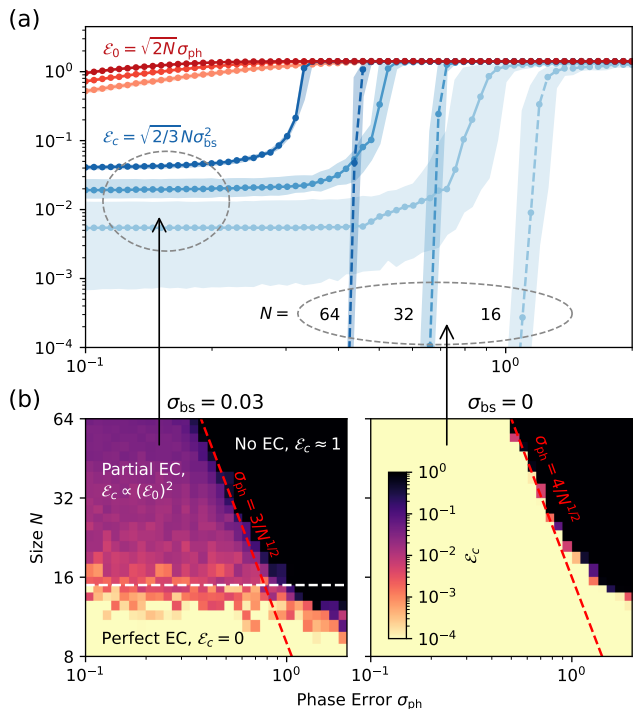

FIG. 11. Effect of phase errors on self-configuration. (a) Error as a function of σ_{ph} for $\sigma_{bs} \in \{0, 0.03\}$, both uncorrected and corrected. Note that uncorrected error saturates to unity for $\sigma_{ph} \gtrsim 0.2$. (b) Corrected error as a function of N and σ_{ph}, showing the sharp error correction threshold at $\sigma_{ph} \sim 3/N^{1/2}$.

culate $\sigma_{ph} = 0.2$. This is small enough to comfortably admit self-configuration on meshes up to size $N \approx 64$ ($N\sigma^2 \approx 2.5$). To configure larger meshes, an initial calibration step [50, 51] will be needed to comfortably reduce the phase errors so that $N\sigma_{ph}^2 \lesssim 2$. This calibration need only be performed once, as the phase shifts can be saved as the initial "cross" state for self-configuring to any matrix U.

2. Uncorrectable Errors: Loss and DAC Precision

Loss Errors. A uniform waveguide loss does not lead to errors in balanced MZI structures; however, in practice losses can be slightly non-uniform. We can model loss errors by introducing additional amplitude scalings to the internal and external arms of the MZI, as follows:

$$T(\theta,\phi) \rightarrow S(\tfrac{\pi}{4})\begin{bmatrix} e^{i\theta-\ell_1/2} & 0 \\ 0 & e^{-\ell_2/2} \end{bmatrix} S(\tfrac{\pi}{4})\begin{bmatrix} e^{i\phi-\ell_3/2} & 0 \\ 0 & e^{-\ell_4/2} \end{bmatrix}$$
$$(D2)$$

where $\ell_i \sim N(0, \sigma_\ell)$ are the loss errors. The uncorrected error is $\mathcal{E}_0 = \sqrt{N/2}\sigma_\ell$. We can perform self-configuration on the mesh to attempt to correct the errors, but in practice the error increases by a factor of $\sqrt{2}$ to $\mathcal{E}_c = \sqrt{N}\sigma_\ell$

(Fig. 12(a)). The matrix error increases because self-configuration is attempting to correct a non-unitary perturbation with a unitary one, in order to correctly set the splitting ratios of the MZIs. This procedure is generally successful at setting the MZIs to the target splitting ratios. However, a nonunitary matrix is not defined solely by its splitting ratio and output phases, so the logic behind error correction breaks down. In fact, the loss error and the corresponding unitary "correction" are orthogonal, so the overall matrix errors add up, leading to the additional factor of $\sqrt{2}$.

Loss errors likely rule out the use of doped-Si thermal phase shifters [43], as alignment and doping density variations lead to a significant wafer-scale variation (0.23 ± 0.13) dB ($\sigma_\ell = 0.03$). However, most SOI platforms use TiN phase shifters, where the heating element is placed sufficiently far above ($\Delta y \gtrsim 0.5\ \mu$m) as to not interact with the waveguide mode. In such phase shifters, loss is determined entirely by the waveguide loss. Wafer-level statistics for typical SOI processes show $\alpha = (2.1 \pm 0.25)$ dB/cm for standard processing, while H_2 thermal annealing can reduce this to (0.1 ± 0.04) dB/cm. Using these figures, for a 200 μm thermal tuner, $\sigma_\ell = 1.1\times10^{-3}$ and 1.8×10^{-4} respectively [38]. These values are based on wafer-scale variations, so the actual die-scale values relevant to moderate-sized meshes may be much smaller.

Fundamentally, loss variations are limited by the statistics of sidewall scattering. Consider a waveguide segment of length L with scatterers of size $\approx \lambda/n_{\text{eff}}$; if these scatterers are randomly and independently placed, the number in a given segment will follow a Poisson distribution with a mean $\langle n \rangle = L/\lambda$ and a standard deviation $\langle \Delta n \rangle_{\text{rms}} = \sqrt{L/\lambda}$. Dividing these quantities, ratio of loss variation to average loss will be $(\Delta\alpha/\alpha)_L \sim \sqrt{\lambda/L}$. A more rigorous calculation based on the scattering theory of Lacey and Payne [37] (see Sec. D 3) gives

$$\frac{\langle \Delta\alpha \rangle_{\text{rms}}}{\langle \alpha \rangle} = \frac{1}{\pi}\sqrt{\frac{\log(4n_0 L/\lambda)}{n_0 L/\lambda}} \qquad \text{(D3)}$$

where n_0 is the cladding index. For a 200 μm phase shifter at 1.55 μm with $\alpha = 2$ dB/cm, Eq. (D3) gives $\sigma_\ell = 5 \times 10^{-4}$, roughly consistent with the wafer-scale variations in Ref. [38]. These loss variations are low enough to allow scaling up to $N \approx 1000$ with matrix errors of at most a few percent.

Programming / DAC Errors. The phase shifts (θ, ϕ) are always programmed to finite bit precision; moreover, finite signal-to-noise ratios in the nulling signals may limit the programming accuracy. To model these effects, in Fig. 12(b) we self-configure the mesh while introducing a random perturbation to every θ and ϕ after setting it. For example, if the phase is set by a DAC with bounds $[0, 2\pi]$ and bit precision B, truncation to B bits leads to a phase error $\sigma_{\text{dac}} = 2\pi/\sqrt{12} \times 2^{-B}$.

Naively, we would expect these errors to add randomly and in quadrature to give $\sqrt{N}\sigma_{\text{dac}}$. However, subsequent
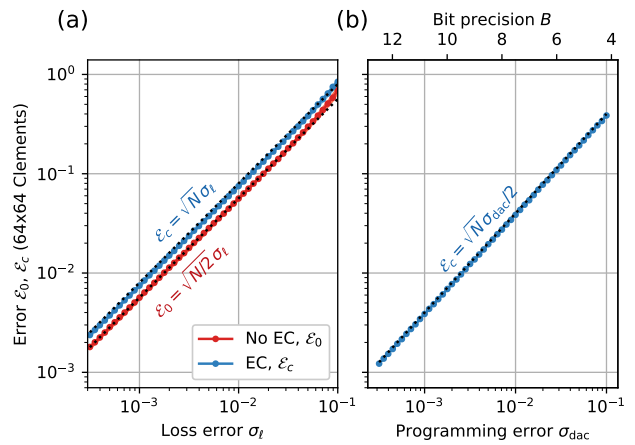

FIG. 12. Uncorrectable errors, 64×64 CLEMENTS mesh. (a) Effect of loss error on mesh with and without correction. (b) Programming error due to finite DAC bit precision or noise.

steps in self-configuration correct for exactly half of this error. To see how, note that analogous to Ref. [18], for any physical MZI

$$T(\theta, \phi) = S(\tfrac{\pi}{4})\begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix} S(\tfrac{\pi}{4})\begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{(D4)}$$

there exist (ψ_1, ψ_2) such that T is represented by a symmetric splitter with phase shifts on its outputs:

$$T = \begin{bmatrix} e^{i\psi_1} & 0 \\ 0 & e^{i\psi_2} \end{bmatrix} \underbrace{\begin{bmatrix} \cos(\theta/2) & i\sin(\theta/2) \\ i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}}_{T_{\text{sym}}(\theta)} \qquad \text{(D5)}$$

These output phase shifts get corrected in subsequent self-configuration steps. Therefore, the residual error to matrix U comes entirely from the errors in the T_{sym} blocks

$$\|\Delta U\|^2 = \sum_{mn}\|T'_{\text{sym}}(\theta_{mn})\Delta\theta_{mn}\|^2 = \frac{1}{2}\sum_{mn}\Delta\theta_{mn}^2 \quad \text{(D6)}$$

which averages to $\mathcal{E}_c = \sqrt{N}\sigma_{\text{dac}}/2$. With $B \geq 10$ bits of precision, one can scale to $N \approx 1000$ with matrix errors of at most a few percent.

3. Statistical Origin of Scattering Loss Variations

Sidewall roughness scattering dominates loss in tightly confined silicon waveguides. This loss is calculated with the volume-current method, where boundary perturbations lead to current sources that scatter into the far field. Lacey and Payne show that, in two dimensions, the loss is given by [37, Eq. 16a]:

$$\alpha = C \int_0^\pi \tilde{R}(\beta - n_0 k_0 \cos\theta)\mathrm{d}\theta \qquad \text{(D7)}$$

$L =$	50	100	200	400	μm
$\Delta\alpha/\alpha$	0.106	0.080	0.060	0.044	
αL	0.01	0.02	0.04	0.08	dB
$\Delta\alpha L$	0.0011	0.0016	0.0023	0.0036	dB
σ_ℓ	2.5	3.7	5.5	8.2	$\times 10^{-4}$

TABLE I. Mean and statistical variation of waveguide loss for a range of lengths, assuming $\alpha = 2$ dB/cm loss. Ratio $\Delta\alpha/\alpha$ comes from Eq. (D18). Normalized nonunitary error magnitude σ_ℓ is given by $\sigma_\ell = \Delta\alpha L/(4.34 \text{ dB})$.

where $k_0 = \omega/c$, $\beta = n_{\text{eff}}k_0$, n_0 is the cladding index, and C is a constant of proportionality. The loss depends on the roughness statistics through $\tilde{R}(k)$, which is the Fourier transform of the autocorrelation function $R(u) = \langle f(z)f(z+u)\rangle$ of the sidewall perturbation $f(z)$. The integral in Eq. (D7) corresponds to the sum over the amplitudes of all scatterers $R(\Delta k_z)$ that phase-match the guided mode $k_z = \beta$ to free-space modes $k_z = n_0 k_0 \cos\theta$ that lie within the light cone $|k_z| < n_0 k_0$.

While Eq. (D7) is usually used to calculate the average waveguide loss, sidewall roughness is a statistical quantity and short waveguides will have large relative roughness variations. We can also use Eq. (D7) to calculate the statistical loss properties of finite-length waveguides by replacing the loss perturbation with a periodic function:

$$f^{(L)}(z) = \sum_{m>0} \sqrt{\pi\tilde{R}(k_m)}\, f_m e^{ik_m z} + \text{c.c.} \quad \text{(D8)}$$

Here $k_m = 2\pi m/L$ is the mth Fourier series mode, while f_m is a complex-valued Gaussian with unit norm in both quadratures, i.e. $\langle\text{Re}[f_m]^2\rangle = \langle\text{Im}[f_m]^2\rangle = 1$, so that $\langle|f_m|^2\rangle = 2$. We calculate the autocorrelation and its spectral function:

$$R^{(L)}(u) = \frac{2\pi}{L}\sum_m \tilde{R}(k_m)|f_m|^2 \cos(k_m u) \quad \text{(D9)}$$

$$\tilde{R}^{(L)}(k) = \frac{\pi}{L}\sum_m \tilde{R}(k_m)|f_m|^2 \delta(k - k_m) \quad \text{(D10)}$$

Note that in the limit $L \to \infty$, $\tilde{R}^{(L)}(k) \to \tilde{R}(k)$ in the sense of distributions, i.e. for any smooth $g(k)$, we have $\int \tilde{R}^{(L)}(k)g(k)\mathrm{d}k \to \int \tilde{R}(k)g(k)\mathrm{d}k$, as the former becomes a Riemann sum for the latter and the statistical fluctuations of $|f_m|^2$ average out when many modes are summed. We can rewrite Eq. (D7) as follows:

$$\alpha = C\int_{\beta-n_0 k_0}^{\beta+n_0 k_0} \tilde{R}(k) \underbrace{\frac{1}{\sqrt{(n_0 k_0)^2 - (\beta - k)^2}}}_{w(\beta-k)}\, \mathrm{d}k \quad \text{(D11)}$$

Following Eq. (D10), the scattering loss becomes a discrete sum:

$$\alpha = C\frac{\pi}{L}\sum_m \tilde{R}(k_m)|f_m|^2 w(\beta - k_m) \quad \text{(D12)}$$

Note that α depends linearly on the $|f_m|^2$, which are i.i.d. random variables. Applying the moments $\langle|f_m|^2\rangle = 2$, $\langle(\Delta|f_m|^2)^2\rangle = 4$, we can calculate the mean and variance of α. These yield discrete sums

$$\langle\alpha\rangle = C\frac{\pi}{L}\sum_m \tilde{R}(k_m)\langle|f_m|^2\rangle w(k_m) \quad \text{(D13)}$$

$$\langle\Delta\alpha^2\rangle = C^2\frac{\pi^2}{L^2}\sum_m \tilde{R}(k_m)^2\langle(\Delta|f_m|^2)^2\rangle w(k_m)^2 \quad \text{(D14)}$$

that may be replaced with integrals in the limit of moderately large L:

$$\langle\alpha\rangle \to C\int \tilde{R}(k)w(k)\mathrm{d}k, \quad \langle\Delta\alpha^2\rangle \to \frac{2\pi C^2}{L}\int \tilde{R}(k)^2 w(k)^2 \mathrm{d}k \quad \text{(D15)}$$

To solve Eqs. (D15), we need to specify roughness model. For example, a Gaussian model $R(s) = e^{-s^2/2L_c^2}$ would yield $\tilde{R}(k) \sim e^{-k^2 L_c^2/2}$ while an exponential model $R(s) = e^{-|s|/L_c}$ would yield $\tilde{R}(k) \sim 1/(1+k^2 L_c^2)$. In both cases the model is characterized by a scatterer size L_c. Here we assume that the scatterers are much smaller than the optical wavelength $L_c \ll \lambda$, so that we can approximate the spectral density as a constant $\tilde{R}(k) \to R_0$, and the model details become irrelevant (in practice $L_c \sim \lambda$ is possible, but the results will be qualitatively similar). Making this assumption, we find $\langle\alpha\rangle = \pi R_0 C$, while the expression for $\langle\Delta\alpha^2\rangle$ formally diverges:

$$\langle\Delta\alpha^2\rangle = \frac{2\pi R_0^2}{L}\int_{-n_0 k_0}^{n_0 k_0} \frac{1}{(n_0 k_0)^2 - k^2}\mathrm{d}k \to \infty \quad \text{(D16)}$$

However, this divergence is merely logarithmic, and the integral is actually just the approximation to a discrete sum Eq. (D14) over a mode set with spacing $2\pi/L$. Therefore, one should actually impose a cutoff and shrink the bounds of the integral to $\pm(n_0 k_0 - \pi/L)$, in which case Eq. (D16) reduces to

$$\langle\Delta\alpha^2\rangle = \frac{2\pi R_0^2 C^2}{n_0 k_0 L}\log(2n_0 k_0 L/\pi) = \frac{\lambda R_0^2 C^2}{n_0 L}\log(4n_0 L/\lambda) \quad \text{(D17)}$$

where $\lambda = 2\pi/k_0$ is the optical wavelength. Although $\langle\alpha\rangle$ and $\langle\Delta\alpha^2\rangle$ are functions of model-dependent constants R_0 and C, their ratio depends only on the refractive index, wavelength, and length of the waveguide segment:

$$\frac{\langle\Delta\alpha\rangle_{\text{rms}}}{\langle\alpha\rangle} = \frac{1}{\pi}\sqrt{\frac{\log(4n_0 L/\lambda)}{n_0 L/\lambda}} \quad \text{(D18)}$$

Table I lists expected losses and their variance for a range of phase shifter lengths and a typical waveguide loss of 2 dB/cm. The ratio $\Delta\alpha/\alpha \lesssim 0.1$ is also consistent with wafer-scale measurements of waveguide loss variations [38]. Note that $\sigma_\ell < 10^{-3}$ for reasonable phase shifter lengths, so the expected error $\mathcal{E}_c = \sqrt{N}\sigma_\ell$ is at most a few percent even for meshes as large as $N = 256$.

Appendix E: Algorithmic Efficiency

A large number of photonic mesh programming algorithms have been proposed in the literature; see generally Appendix B of Ref. [24]. Table II summarizes some of the leading approaches in terms of both their computational cost, generality (applicable mesh geometries). This list is not exhaustive, but gives a representative sample that illustrates the tradeoffs in the field. With the exception of the algorithms recently reported by us [24], to date all optimization schemes require either (1) accurate pre-calibration of the hardware errors, or (2) $O(N^2)$ internal photodetectors used to monitor power at intermediate points on the mesh. Our algorithm uniquely lacks both requirements, so it can be performed on uncalibrated "zero-change" photonic hardware, requiring only coherent control of the input fields and coherent detection at the outputs, and works for both RECK and CLEMENTS meshes.

Even in absence of these considerations, our algorithm is competitive against in terms of computational resources. The required resources of an algorithm depend on whether it is performed in situ (for correction of unknown errors) or in silico (for known, calibrated errors). For in-situ algorithms, the number of matrix-vector multiplication (MVM) calls is most relevant. Our algorithm takes approximately $2N^2$ calls (4 calls per MZI), which is comparable to progressive self-configuration and the initial RELLIM proposal [22, 53] (although RELLIM with internal detectors can be parallelized to run in $O(N)$ time [16]). Alternatively, numerical optimization with T time steps takes $3NT$ calls (three calls per step, first to estimate U, next to back-propagate $U - \hat{U}$, and finally a forward-pass step to compute the gradient with respect to phase shifts [17]). Fig. 13(a) shows the convergence of a 64×64 CLEMENTS mesh in terms of MVMs under the L-BFGS-B algorithm [54]. Convergence depends

FIG. 13. Convergence of L-BFGS-B for an imperfect CLEMENTS mesh (64×64, $\sigma = 0.02$), minimizing the L_2 norm. The convergence curve depends on the initial solution, and is compared with the result from self-configuration. (a) Plot in terms of in-situ MVMs to solution. (b) In-silico CFLOPs to solution.

		Mesh*		MVMs	CFLOPs	Caveats†
Progressive	[53]	R		N^2	N^3	(D)
RELLIM	[16, 22]	R	C	N–N^2	N^3	(D)
In-situ	[17]	R	C	NT	N^3T	(D)
SGD	[15]	R	C	BT	N^2BT	(C)
Numerical	[13, 14]	R	C	NT	N^3T	(C)
Local	[18]	R	C	–	N^2	(C)
Direct	[24]	R		N^2	N^3	(S)
Ratio	[24]	R		N^2	N^3	
This work		R	C	N^2	N^3	

TABLE II. Comparison of a representative sample of optimization algorithms. Scaling of computational cost plotted for in-situ resources (number of MVM calls), and in-silico resources (CFLOPs), where N is the mesh size, T is the number of optimization steps, and B is the SGD batch size. *Mesh types: (R)eck, (C)lements. †Caveats include (D) requires internal detectors, (C) requires pre-calibration of errors, (S) stability issues in presence of errors.
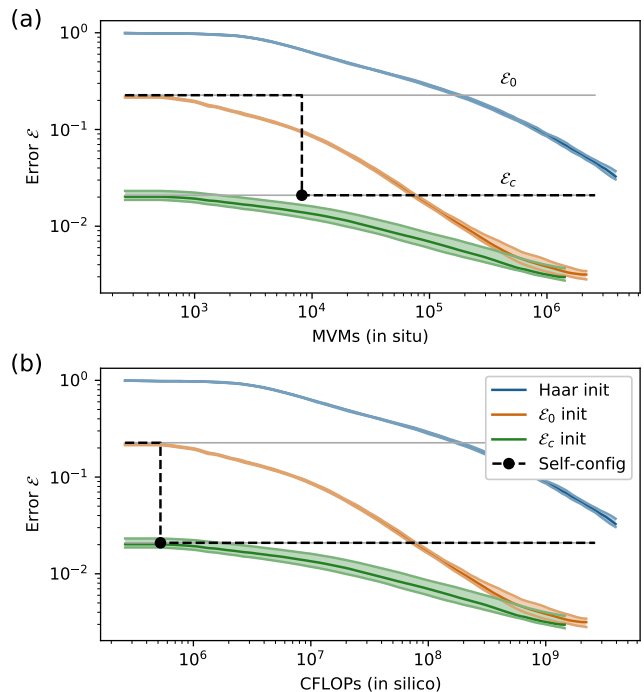
strongly on the mesh initialization [15]. Initializing to Haar-random unitaries is a significant improvement over random phase shifts, since a mesh with random phase shifts has a banded structure that leads to vanishing gradients with respect to matrix elements far from the diagonal. Even with Haar initialization, the algorithm takes thousands of steps and millions of MVMs to converge to the accuracy $\mathcal{E}_c$ reached by our algorithm. Initializing to the phases of an ideal (error-free) CLEMENTS mesh helps considerably, but optimization still takes $10\times$ more calls. However, for CLEMENTS meshes it appears that the global minimum can be as much as $5\times$ lower than the self-configuration result; as a result, numerical optimization may still be helpful as a refinement technique, where the mesh is initialized using self-configuration, followed by a numerical procedure to improve this solution.

Since a single optimization step requires N MVM calls, stochastic gradient descent (SGD) has been proposed as a solution to speed up matrix optimization, where a batch of $B < N$ random columns are used rather than the whole matrix. However, SGD tends to trade off batch size for iteration count, so the overall resource requirement is higher [15].

Fig. 13(b) shows the performance gap with respect to the in-silico case, where accurate calibration of the errors (α, β) allows the phases (θ, ϕ) to be computed nu-

merically. Here, optimization protocols require $O(N^3)$ complex floating point operations (CFLOPs), equivalent to one matrix-matrix multiplication per time step, leading to a scaling of $N^3 T$ (SGD scales as $N^2 BT$). Self-configuration runs by performing approximately $N^2/2$ Givens rotations with $4N$ CFLOPs each, for a total of $2N^3$ CFLOPs, and thus runs about $10^2 \times$ faster. As before, numerical optimization can still be helpful as a means for further refinement of the solution.

However, for in silico optimization, the local correction method [18] is superior, as it takes only N^2 CFLOPs and is parallelizable to N time steps.

[1] J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh, *et al.*, Universal linear optics, Science **349**, 711 (2015).

[2] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Quantum computational advantage using photons, Science 10.1126/science.abe8770 (2020).

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

[4] A. N. Tait, T. F. Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, Neuromorphic photonic networks using silicon photonic weight banks, Scientific Reports **7**, 7430 (2017).

[5] R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, Large-scale optical neural networks based on photoelectric multiplication, Physical Review X **9**, 021032 (2019).

[6] D. Marpaung, C. Roeloffzen, R. Heideman, A. Leinse, S. Sales, and J. Capmany, Integrated microwave photonics, Laser & Photonics Reviews **7**, 506 (2013).

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

[8] N. C. Harris, J. Carolan, D. Bunandar, M. Prabhu, M. Hochberg, T. Baehr-Jones, M. L. Fanto, A. M. Smith, C. C. Tison, P. M. Alsing, *et al.*, Linear programmable nanophotonic processors, Optica **5**, 1623 (2018).

[9] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Experimental realization of any discrete unitary operator, Physical Review Letters **73**, 58 (1994).

[10] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optimal design for universal multiport interferometers, Optica **3**, 1460 (2016).

[11] N. C. Harris, R. Braid, D. Bunandar, J. Carr, B. Dobbie, C. Dorta-Quinones, J. Elmhurst, M. Forsythe, M. Gould, S. Gupta, *et al.*, Accelerating artificial intelligence with silicon photonics, in *2020 Optical Fiber Communications Conference and Exhibition (OFC)* (IEEE, 2020) pp. 1–4.

[12] C. Ramey, Silicon photonics for artificial intelligence acceleration, in *2020 IEEE Hot Chips 32 Symposium (HCS)* (IEEE Computer Society, 2020) pp. 1–26.

[13] R. Burgwal, W. R. Clements, D. H. Smith, J. C. Gates, W. S. Kolthammer, J. J. Renema, and I. A. Walmsley, Using an imperfect photonic network to implement random unitaries, Optics Express **25**, 28236 (2017).

[14] J. Mower, N. C. Harris, G. R. Steinbrecher, Y. Lahini, and D. Englund, High-fidelity quantum state evolution in imperfect photonic integrated circuits, Physical Review A **92**, 032322 (2015).

[15] S. Pai, B. Bartlett, O. Solgaard, and D. A. B. Miller, Matrix optimization on universal unitary photonic devices, Physical Review Applied **11**, 064044 (2019).

[16] S. Pai, I. Williamson, T. W. Hughes, M. Minkov, O. Solgaard, S. Fan, and D. A. B. Miller, Parallel programming of an arbitrary feedforward photonic network, IEEE Journal of Selected Topics in Quantum Electronics (2020).

[17] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, Training of photonic neural networks through in situ backpropagation and gradient measurement, Optica **5**, 864 (2018).

[18] S. Bandyopadhyay, R. Hamerly, and D. Englund, Hardware error correction for programmable photonics, Optica **8**, 1247 (2021).

[19] S. P. Kumar, L. Neuhaus, L. G. Helt, H. Qi, B. Morrison, D. H. Mahler, and I. Dhand, Mitigating linear optics imperfections via port allocation and compilation, arXiv preprint arXiv:2103.03183 (2021).

[20] D. A. Miller, Self-aligning universal beam coupler, Optics Express **21**, 6360 (2013).

[21] D. A. Miller, Self-configuring universal linear optical component, Photonics Research **1**, 1 (2013).

[22] D. A. Miller, Setting up meshes of interferometers–reversed local light interference method, Optics Express **25**, 29233 (2017).

[23] S. Grillanda, M. Carminati, F. Morichetti, P. Ciccarella, A. Annoni, G. Ferrari, M. Strain, M. Sorel, M. Sampietro, and A. Melloni, Non-invasive monitoring and control in silicon photonics using CMOS integrated electronics, Optica **1**, 129 (2014).

[24] R. Hamerly, S. Bandyopadhyay, and D. Englund, Stability of self-configuring large multiport interferometers, arXiv preprint arXiv:2106.04363 (2021).

[25] M. Y.-S. Fang, S. Manipatruni, C. Wierzynski, A. Khosrowshahi, and M. R. DeWeese, Design of optical neural networks with component imprecisions, Optics Express **27**, 14009 (2019).

[26] R. Hamerly, Meshes: tools for modeling photonic beam-splitter mesh networks, Online at: `https://github.com/QPG-MIT/meshes` (2021).

[27] A. Haar, Der massbegriff in der theorie der kontinuierlichen gruppen, Annals of Mathematics , 147 (1933).

[28] W.-K. Tung, *Group theory in physics: an introduction to symmetry principles, group representations, and special functions in classical and quantum physics* (World Scientific Publishing Company, 1985).

[29] N. J. Russell, L. Chakhmakhchyan, J. L. O'Brien, and A. Laing, Direct dialling of Haar random unitary matrices, New Journal of Physics **19**, 033007 (2017).

[30] L. Bernstein, A. Sludds, R. Hamerly, V. Sze, J. Emer, and D. Englund, Freely scalable and reconfigurable opti-

cal hardware for deep learning, Scientific Reports **11**, 1 (2021).

[31] I. A. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, Reprogrammable electro-optic nonlinear activation functions for optical neural networks, IEEE Journal of Selected Topics in Quantum Electronics **26**, 1 (2019).

[32] H. Zhang, M. Gu, X. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. Yung, *et al.*, An optical neural chip for implementing complex-valued neural network, Nature Communications **12**, 1 (2021).

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE **86**, 2278 (1998).

[34] S. Pai, Neurophox: a simulation framework for unitary neural networks and photonic devices, Online at: `https://github.com/solgaardlab/neurophox` (2020).

[35] See supplemental material at `localhost:631/printers` for neural network models and source code (todo: fix link on publication).

[36] J. C. Mikkelsen, W. D. Sacher, and J. K. Poon, Dimensional variation tolerant silicon-on-insulator directional couplers, Optics Express **22**, 3145 (2014).

[37] J. Lacey and F. Payne, Radiation loss from planar waveguides with random wall imperfections, IEE Proceedings J-Optoelectronics **137**, 282 (1990).

[38] Q. Wilmart, S. Brision, J.-M. Hartmann, A. Myko, K. Ribaud, C. Petit-Etienne, L. Youssef, D. Fowler, B. Charbonnier, C. Sciancalepore, *et al.*, A complete Si photonics platform embedding ultra-low loss waveguides for o-and c-band, Journal of Lightwave Technology **39**, 532 (2020).

[39] X. Chen, M. M. Milosevic, D. J. Thomson, A. Z. Khokhar, Y. Franz, A. F. Runge, S. Mailis, A. C. Peacock, and G. T. Reed, Post-fabrication phase trimming of mach–zehnder interferometers by laser annealing of germanium implanted waveguides, Photonics Research **5**, 578 (2017).

[40] D. A. Miller, Perfect optics with imperfect components, Optica **2**, 747 (2015).

[41] X. Li, N. Youngblood, C. Ríos, Z. Cheng, C. D. Wright, W. H. Pernice, and H. Bhaskaran, Fast and reliable storage using a 5 bit, nonvolatile photonic memory cell, Optica **6**, 1 (2019).

[42] D. M. Gill, S. S. Patel, M. Rasras, K.-Y. Tu, A. E. White, Y.-K. Chen, A. Pomerene, D. Carothers, R. L. Kamocsai, C. M. Hill, *et al.*, CMOS-compatible Si-ring-assisted Mach–Zehnder interferometer with internal bandwidth equalization, IEEE Journal of Selected Topics in Quantum Electronics **16**, 45 (2009).

[43] N. C. Harris, Y. Ma, J. Mower, T. Baehr-Jones, D. Englund, M. Hochberg, and C. Galland, Efficient, compact and low loss thermo-optic phase shifter in silicon, Optics Express **22**, 10487 (2014).

[44] V. J. López-Pastor, J. S. Lundeen, and F. Marquardt, Arbitrary optical wave evolution with Fourier transforms and phase masks, arXiv preprint arXiv:1912.04721 (2019).

[45] S. Fldzhyan, M. Y. Saygin, and S. Kulik, Optimal design of error-tolerant reprogrammable multiport interferometers, Optics Letters **45**, 2632 (2020).

[46] M. Y. Saygin, I. V. Kondratyev, I. V. Dyakonov, S. A. Mironov, S. S. Straupe, and S. P. Kulik, Robust architecture for programmable universal unitaries, Physical Review Letters **124**, 010501 (2020).

[47] R. Tanomura, R. Tang, S. Ghosh, T. Tanemura, and Y. Nakano, Robust integrated optical unitary converter using multiport directional couplers, Journal of Lightwave Technology **38**, 60 (2019).

[48] J. Polcari, Generalizing the butterfly structure of the FFT, in *Advanced Research in Naval Engineering* (Springer, 2018) pp. 35–52.

[49] M. Arjovsky, A. Shah, and Y. Bengio, Unitary evolution recurrent neural networks, in *International conference on machine learning* (PMLR, 2016) pp. 1120–1128.

[50] S. Suda, H. Matsuura, K. Tanizawa, K. Suzuki, K. Ikeda, H. Kawashima, and S. Namiki, Fast and accurate automatic calibration of a 32×32 silicon photonic strictly-non-blocking switch, in *Photonics in Switching* (Optica Publishing Group, 2017) pp. PTu3C–5.

[51] C. Alexiev, J. C. Mak, W. D. Sacher, and J. K. Poon, Calibrating rectangular interferometer meshes with external photodetectors, OSA Continuum **4**, 2892 (2021).

[52] K. Suzuki, R. Konoike, J. Hasegawa, S. Suda, H. Matsuura, K. Ikeda, S. Namiki, and H. Kawashima, Low-insertion-loss and power-efficient 32×32 silicon photonics switch with extremely high-Δ silica PLC connector, Journal of Lightwave Technology **37**, 116 (2018).

[53] D. A. Miller, Self-configuring universal linear optical component, Photonics Research **1**, 1 (2013).

[54] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM Journal on Scientific Computing **16**, 1190 (1995).