



CHORUS

This is the accepted manuscript made available via CHORUS. The article has been published as:

Matrix Optimization on Universal Unitary Photonic Devices

Sunil Pai, Ben Bartlett, Olav Solgaard, and David A. B. Miller

Phys. Rev. Applied **11**, 064044 — Published 19 June 2019

DOI: [10.1103/PhysRevApplied.11.064044](https://doi.org/10.1103/PhysRevApplied.11.064044)

Matrix optimization on universal unitary photonic devices

Sunil Pai,^{1,*} Ben Bartlett,² Olav Solgaard,¹ and David A. B. Miller^{1,†}

¹*Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA*

²*Department of Applied Physics, Stanford University, Stanford, CA 94305, USA*

Universal unitary photonic 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.

I. 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) [1–4], mode unscramblers [5], photonic neural networks [6–8], and finding optimal channels through lossy scatterers [9]. While universal photonic devices have been experimentally realized at a relatively small scale [5, 6], 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.

One canonical universal photonic device is the rectangular multiport interferometer mesh [10] shown in Figure 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 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 [11] (shown in Appendix C), the universal cascaded binary tree architecture [12], and lattice architectures where light does not move in a forward-only direction [13–15].

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.

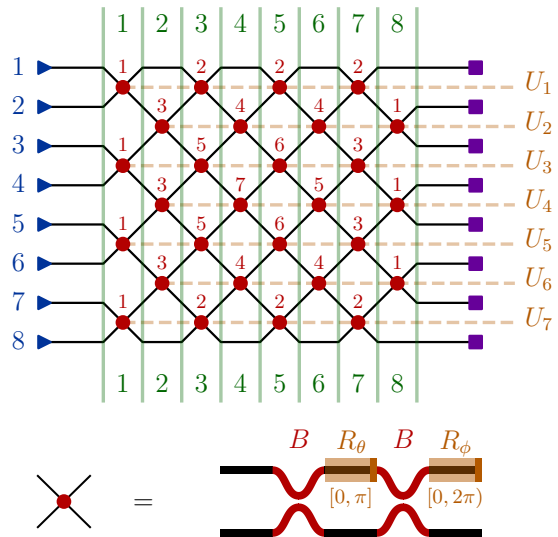


FIG. 1. Mesh diagram representing the locally interacting rectangular mesh for $N = 8$. The inputs (and single-mode phase shifts at the inputs) are represented by blue triangles. Outputs are represented by purple squares. The MZI nodes are represented by red dots labelled with sensitivity index $\alpha_{n\ell}$ (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 $0 \leq \phi < 2\pi$.

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 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 [6] and quantum computing [2, 16], we avoid this banded unitary

* sunilpai@stanford.edu

† dabm@stanford.edu

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 [16–20]. 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 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 II, we introduce the photonic mesh architecture and sources of error that can exacerbate the banded unitary matrix problem. In Section III, 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 IV and V, 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.

II. PHOTONIC MESH

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

A. 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:

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

where B is the beamsplitter operator, R_θ, R_ϕ are upper phase shift operators. Equation 1 is represented diagram-

matically by the configuration in Figure 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:

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

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 1 in N -dimensional space with a locally-interacting unitary “Givens rotation” U_n defined as:

$$U_n := \begin{matrix} & & n & n+1 & & \\ \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & U_{11} & U_{12} & \cdots & 0 \\ 0 & \cdots & U_{21} & U_{22} & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix} & \begin{matrix} n \\ n+1 \end{matrix} & \end{matrix}. \quad (3)$$

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} .

Arbitrary unitary transformations can be implemented on a photonic chip using only locally interacting MZIs [11]. In this paper, we focus on optimizing a rectangular mesh [10] of MZIs; however, our ideas can be extended to other universal schemes, such as the triangular mesh [22], as well.

In the rectangular mesh scheme [10] of Figure 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 $D(\gamma_1, \gamma_2, \dots, \gamma_N)$:

$$\hat{U}_R := \prod_{\ell=1}^N \prod_{n \in \mathcal{S}_{\ell, N}} U_n(\theta_{n\ell}, \phi_{n\ell}) \cdot D(\gamma_1, \gamma_2, \dots, \gamma_N), \quad (4)$$

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

¹ Other configurations with two independent phase shifters between the beamsplitters B are ultimately equivalent for photonic meshes [21].

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

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. [7, 23] and is depicted in Figure 1, where ℓ represents the index of the vertical layer.

B. Beamsplitter error tolerances

The expressions in Equations 1 and 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 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 1 or B in Equation 1. The resultant scattering matrix U_ϵ with imperfect beamsplitters B_ϵ can be written as:

$$\begin{aligned} B_\epsilon &:= \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{1+\epsilon} & i\sqrt{1-\epsilon} \\ i\sqrt{1-\epsilon} & \sqrt{1+\epsilon} \end{bmatrix} \\ U_\epsilon &:= R_\phi B_\epsilon R_\theta B_\epsilon. \end{aligned} \quad (5)$$

As shown in Appendix B, 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 2.

The unitary matrices in Equation 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 [24–26]. Our Haar phase theory, which we develop in the following section, determines acceptable interferometer tolerances for calibration of a “perfect mesh” consisting of imperfect beamsplitters [21] given large N . We will additionally show that simulated photonic backpropagation [7] 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 $N = 128$.

C. 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 1, shown in orange in Figure 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.

III. HAAR INITIALIZATION

A. 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 paper 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) [16, 24]. 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(b).

For an implemented Haar random unitary matrix, low-tolerance, transmissive MZIs are located towards the center of a rectangular mesh [16, 24] and the apex of a triangular mesh as proven in Appendix C. 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 waveguide ports, as proven in Appendix I. 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 1 and Figure 2(a) show the sensitivity index for the rectangular mesh, which clearly increases towards the center MZI.

B. Phase shift distributions and Haar phase

The external $\phi_{n\ell}, \gamma_n$ phase shifts do not affect the transmissivity $t_{n\ell}$ and therefore obey uniform random distributions [16]. In contrast, the $\theta_{n\ell}$ phase shifts have a probability density function (PDF) that depends on

³ Since $\gamma_n, \phi_{n\ell}$ 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 $t_{n\ell} \in [0, 1]$.

⁴ A Haar random unitary is defined as Gram-Schmidt orthogonalization of N standard normal complex vectors [16, 20].

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

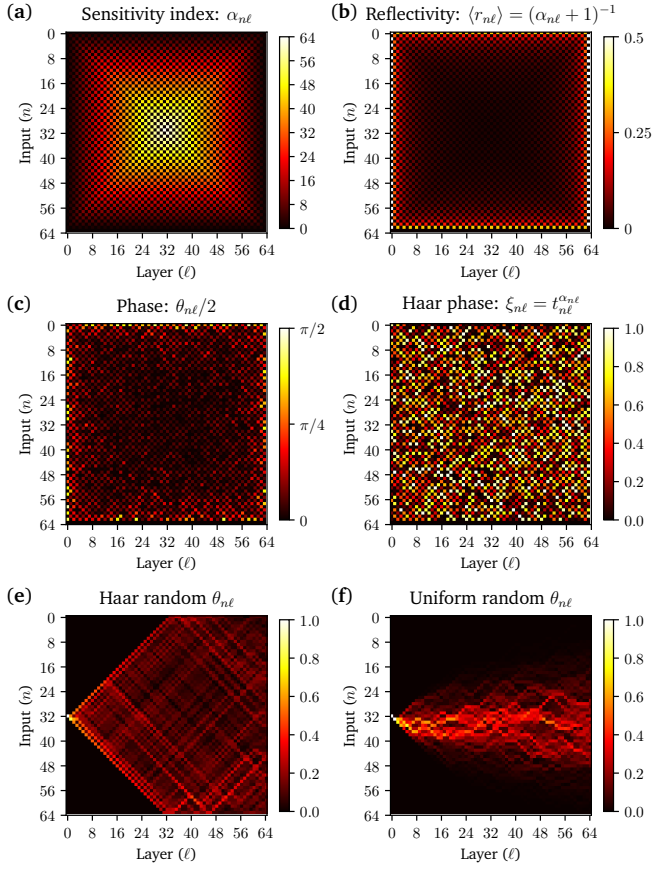


FIG. 2. (a) The sensitivity index $\alpha_{n\ell}$ for $N = 64$. (b) Checkerboard plot for the average reflectivity $\langle r_{n\ell} \rangle$ in a rectangular mesh. (c) Haar-random matrix and run the decomposition in Ref. [10] to find phases approaching cross state in the middle of the mesh. (d) The 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 $N = 64$.

$\alpha_{n\ell}$ [16]:

$$\mathcal{P}_{\alpha_{n\ell}}\left(\frac{\theta_{n\ell}}{2}\right) = \alpha_{n\ell} \sin\left(\frac{\theta_{n\ell}}{2}\right) \left[\cos\left(\frac{\theta_{n\ell}}{2}\right)\right]^{2\alpha_{n\ell}-1}. \quad (6)$$

The general shape of this distribution is presented in Figure 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 $\theta_{n\ell}/2 = \pi/2$:

$$\xi_{n\ell} := \int_{\pi/2}^{\theta_{n\ell}/2} \mathcal{P}_{\alpha_{n\ell}}(\theta) d\theta. \quad (7)$$

Using Equations 6 and 7, we can define $\xi_{n\ell}(\theta_{n\ell}) \in [0, 1]$ that yields a Haar random matrix:

$$\xi_{n\ell} = \left[\cos^2\left(\frac{\theta_{n\ell}}{2}\right)\right]^{\alpha_{n\ell}} = t_{n\ell}^{\alpha_{n\ell}}, \quad (8)$$

where $t_{n\ell}$ represents the transmissivity of the MZI, which is a function of $\theta_{n\ell}$ as defined in Equation 2.

C. Haar initialization

In the physical setting, it is useful to find the inverse of Equation 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 2(c, d):

$$\begin{aligned} t_{n\ell} &= \alpha_{n\ell} \sqrt{\xi_{n\ell}} \\ \theta_{n\ell} &= 2 \arccos \sqrt{t_{n\ell}} = 2 \arccos \sqrt[2\alpha_{n\ell}]{\xi_{n\ell}}, \end{aligned} \quad (9)$$

where the expression for $\theta_{n\ell}$ is just a rearrangement of Equation 2.

Haar initialization can be achieved progressively using a procedure similar to that in Ref. [25]. If the phase shifters in the mesh are all well-characterized, the transmissivities can be directly set [16]. We will show in Section V that Haar initialization improves the convergence speed of gradient descent optimization significantly.

We can also use Equation 9 to find the average transmissivity and reflectivity for an MZI parameterized by $\alpha_{n\ell}$ as is found through simulation in Ref. [24]:

$$\begin{aligned} \langle t_{n\ell} \rangle &= \int_0^1 d\xi_{n\ell} \alpha_{n\ell} \sqrt{\xi_{n\ell}} = \frac{\alpha_{n\ell}}{\alpha_{n\ell} + 1} \\ \langle r_{n\ell} \rangle &= \frac{1}{\alpha_{n\ell} + 1} = \frac{1}{|I_{n\ell}| + |O_{n\ell}| - N}. \end{aligned} \quad (10)$$

The average reflectivity $\langle r_{n\ell} \rangle$ shown in Figure 2(b) gives a simple interpretation for the sensitivity index shown in Figure 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 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 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 2(f).

D. 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

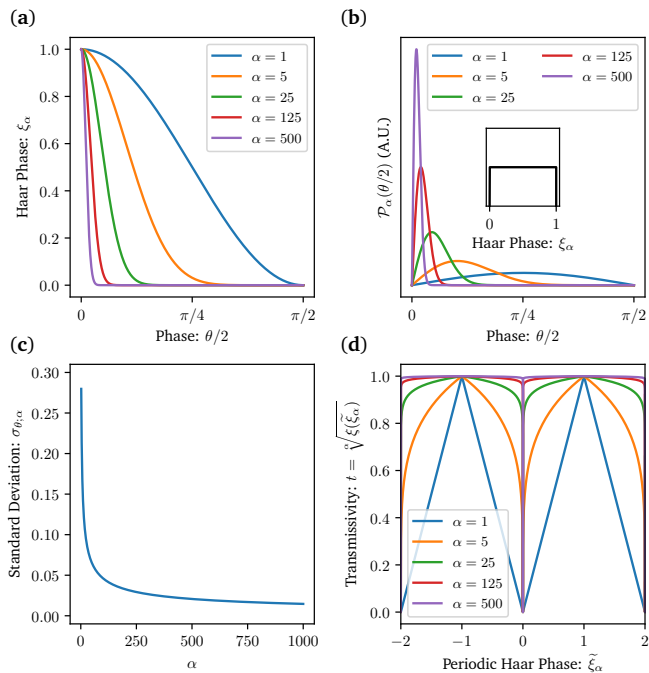


FIG. 3. (a) Plot of the relationship between ξ_α and θ . (b) We show that phase shift standard deviation $\sigma_{\theta;\alpha}$ decreases as α increases. (c) A plot of $\sigma_{\theta;\alpha}$ as α increases. (d) The transmissivity of an MZI component as a function of a periodic Haar phase has a power law relationship. The periodic Haar phase ξ_α is mapped to the Haar phase by a function $\xi: \mathbb{R} \rightarrow [0, 1]$ as discussed in Appendix G.

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(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(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 [6], 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 V.

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(d) and Equation 9. The transmissivity dependence on a periodic Haar phase (shown in Figure 3(d) and discussed in Appendix G), is markedly different from the usual sinusoidal dependence on periodic $\theta_{n\ell}$. 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.

IV. ARCHITECTURE MODIFICATIONS

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

A. Redundant rectangular mesh (RRM)

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 redundant rectangular mesh is depicted in green in Figure 4(a). The authors in Ref. [24] 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.

B. Permuting rectangular mesh (PRM)

Another method to accelerate the optimization of a rectangular mesh is to shuffle outputs at regular intervals within the rectangular 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 4(b), or using low-loss waveguide crossings.

We now add permutation matrices P_1, P_2, \dots, P_{K-1}

⁶ If 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.

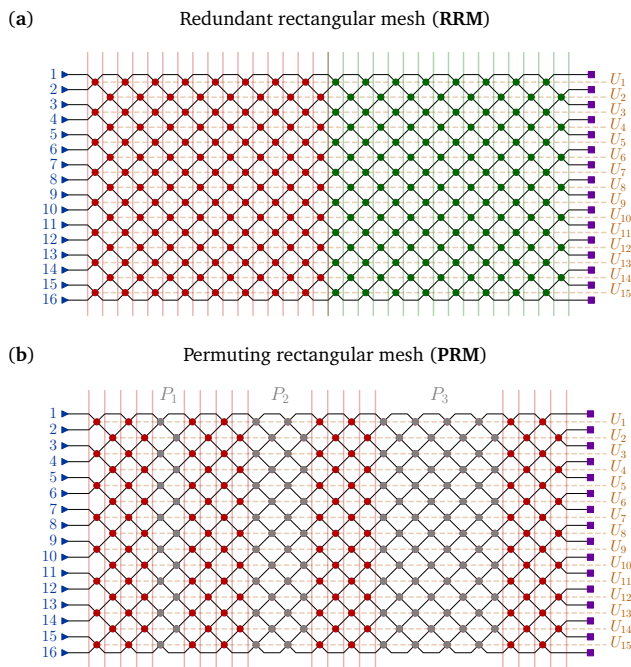


FIG. 4. (a) A 16×16 rectangular mesh (red). Extra tunable layers (green) may be added to significantly reduce convergence time. (b) A 16-input, 30-layer permuting rectangular mesh. The rectangular permutation layer is implemented using either waveguide crossings or cross state MZIs (gray).

into the middle of the rectangular mesh as follows

$$\hat{U}_{\text{PR}} := M_K \left(\prod_{k=1}^{K-1} P_k M_k \right) \quad (11)$$

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

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 4(b).

V. 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.

A. Mesh initialization

We begin by discussing the importance of initializing the mesh to respect the cross state bias and sensitivity of

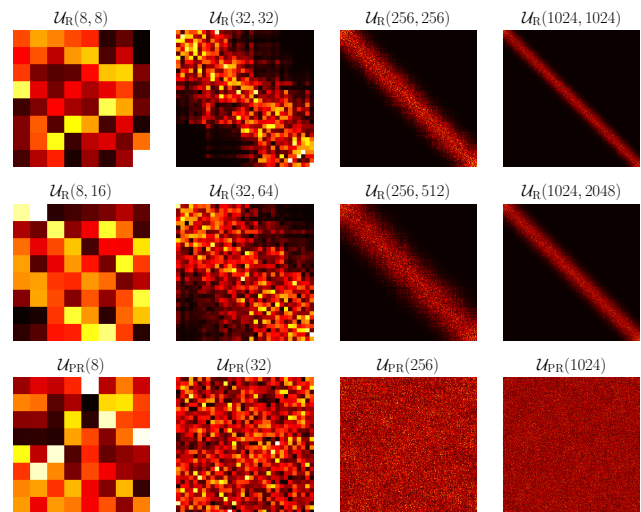


FIG. 5. Elementwise absolute values of unitary matrices resulting from rectangular ($U \sim \mathcal{U}_R$) and permuting rectangular ($U \sim \mathcal{U}_{\text{PR}}$) meshes where meshes are initialized with uniform-random phases.

each component for Haar random unitary matrices discussed in Section III. 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 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 5.

We compare the bandsizes of banded unitary matrices in simulations qualitatively as we do in Figure 5 or quantitatively as we do in Appendix D. We randomly generate $U \sim \mathcal{U}_R(N, N)$, $U \sim \mathcal{U}_{\text{PR}}(N)$ (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 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 5 and Appendix D. 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 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

(θ_{nl} 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 α_{nl} , 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.

B. 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 :

$$\underset{\theta_{nl}, \phi_{nl}, \gamma_n}{\text{minimize}} \quad \frac{1}{2N} \left\| \hat{U}(\theta_{nl}, \phi_{nl}, \gamma_n) - U \right\|_F^2, \quad (12)$$

where the estimated unitary matrix function \hat{U} maps N^2 phase shift parameters $\theta_{nl}, \phi_{nl}, \gamma_n$ to $U(N)$ via Equations 4 or 11, and $\|\cdot\|_F$ denotes the Frobenius norm. Since trigonometric functions parameterizing \hat{U} are non-convex, we know that Equation 12 is a non-convex problem. The non-convexity of Equation 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 12.

C. 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. [7]. This optical backpropagation procedure physically measures $\partial \mathcal{L}_{\text{train}} / \partial \theta_{nl}$ using interferometric techniques, which can be extended to any of the architectures we discuss in this paper.

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 [6]. We find empirically that the Adam update rule (a popular first-order

adaptive update rule [27]) 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. [24] 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. [7, 23]. The models were programmed in `tensorflow` [28] and run on an NVIDIA GeForce GTX1080 GPU to improve optimization performance.

D. Results

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 θ_{nl} are randomly or Haar-initialized (according to the PDF in Equation 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 6.
2. Rectangular meshes converge faster when Haar-initialized than when uniformly random initialized, as in Figure 6, in which case the estimated matrix converges towards a banded configuration shown in Appendix H.
3. Permuting rectangular meshes converge faster than rectangular meshes despite having the same number of total parameters as shown in Figure 6.
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 7.

⁷ See <https://github.com/solgaardlab/neurophox>.

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 6 and Appendix E.)

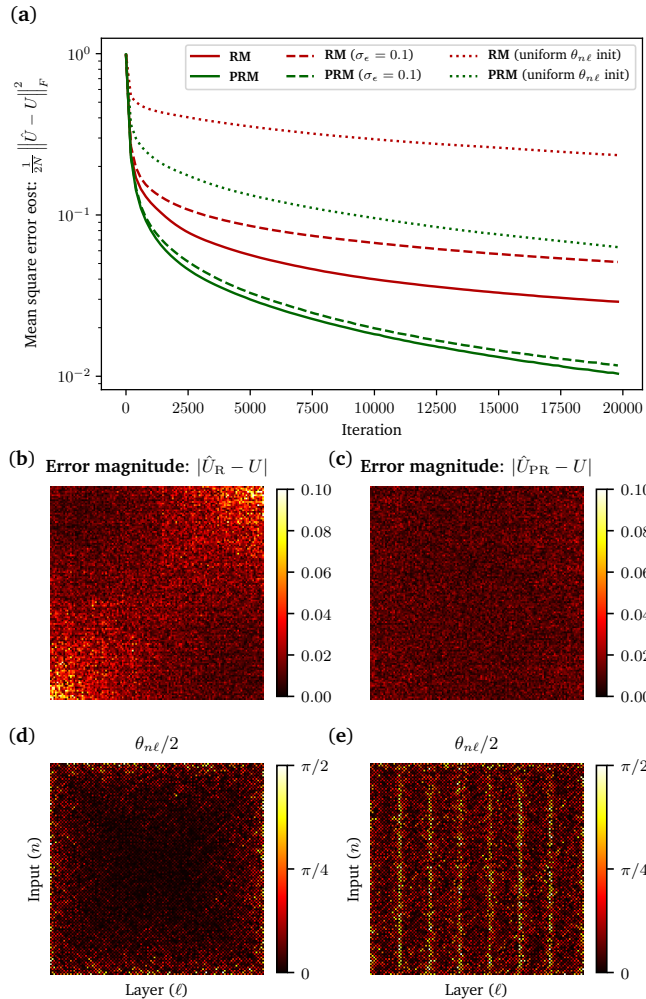


FIG. 6. We implement six different optimizations for $N = 128$ where we vary the choice of permuting rectangular mesh (PRM) or rectangular mesh (RM); the initialization (random $\theta_{n\ell}$ or Haar-initialized $\theta_{n\ell}$); and photonic transmissivity error displacements ($\epsilon = 0$ or $\epsilon \sim \mathcal{N}(0, 0.01)$, 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 G) 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.

The singular value decomposition (SVD) architecture

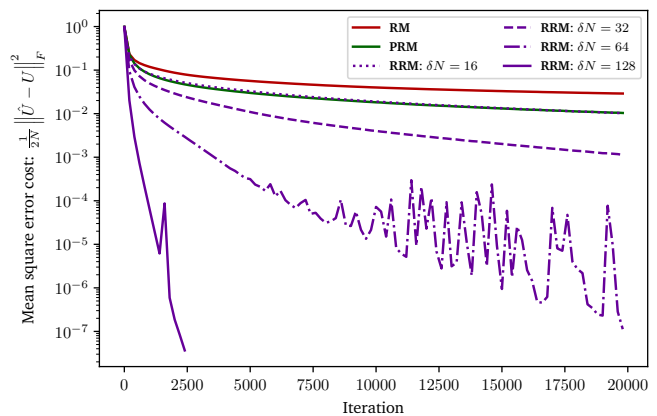


FIG. 7. A comparison of test error in `tensorflow` for $N = 128$ between rectangular (RM), permuting rectangular (PRM), and redundant rectangular (RRM) meshes for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256. Ideal = Haar random initialized $\theta_{n\ell}$ with $\epsilon = 0$. 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.

discussed in Refs. [6, 22] 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. [7]), 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 F.

VI. DISCUSSION

A. 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 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 9.

The average reflectivities given by Haar initialization may be useful for inverse design approaches [29] 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 [6]. 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 V, 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 [6], 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(c). As mentioned in Section III, the ideal transmissivity-voltage dependence with this consideration would be identical to the transmissivity vs Haar phase dependence in Figure 3(d).

B. Applications of mesh optimization

Meshes can be tuned using either self-configuration [11, 22] or global optimizations (gradient-based [7] or derivative-free [30]). The algorithmic optimizations proposed in Refs. [11, 22] 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 [21, 31] or a vertical layer-wise progressive algorithm [25]. 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 [7]. This model-free property makes gradient-based optimization robust to fabrication error; we show in Figure 6(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.

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 non-local waveguide interactions [23, 26, 32, 33]; 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. [24], simply adds more tunable paths for the light to achieve a desired output. As shown in Figure 6, 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 [34, 35]. 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 [34, 35]. A similar concept is used in unitary spatial mode manipulation, where stochastic optimization of deformable mirror settings allow for efficient mode conversion [36]. 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 [30]. The success of simulated annealing in experimentally training small unitary photonic devices [30] (rather than gradient descent as is used in this work) 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 [37–39]. 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.

⁸ Ref. [6] propose a standard deviation of $\sim 10^{-3}$ might be possible with further circuit characterization, which might be scalable based on Figure 3(c).

Such “unitary RNNs” (URNs) have already been simulated on conventional computers and show some promise in synthetic long-term memory tasks [23, 40]. Unitary RNNs are physically implementable using a single mesh with optical nonlinearities and recurrent optoelectronic feedback, suggesting that the architecture discussed in this work 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.

VII. 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 III, 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 V 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 IV, 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 7. 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 [3, 5, 6, 12, 22, 37–39, 41].

ACKNOWLEDGMENTS

This work was funded by the Air Force Office of Scientific Research, specifically for the Center for Energy-Efficient 3D Neuromorphic Nanocomputing (CEE3N²), Grant No. FA9550-181-1-0186. We would like to thank Tyler Hughes, Momchil Minkov, Nate Abebe, Dylan Black, and Ian Williamson for illuminating discussions.

-
- [1] Markus Gräfe, Ren Heilmann, Maxime Lebugle, Diego Guzman-Silva, Armando Perez-Leija, and Alexander Szameit, “Integrated photonic quantum walks,” *Journal of Optics* (2016), 10.1088/2040-8978/18/10/103002.
 - [2] Jacques Carolan, Christopher Harrold, Chris Sparrow, Enrique Martín-López, Nicholas J. Russell, Joshua W. Silverstone, Peter J. Shadbolt, Nobuyuki Matsuda, Manabu Oguma, Mikitaka Itoh, Graham D. Marshall, Mark G. Thompson, Jonathan C.F. Matthews, Toshikazu Hashimoto, Jeremy L. O’Brien, and Anthony Laing, “Universal linear optics,” *Science* (2015), 10.1126/science.aab3642.
 - [3] 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* (2013), 10.1126/science.1231692.
 - [4] 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**, 447–452 (2017).
 - [5] 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** (2017), 10.1038/lsa.2017.110.
 - [6] 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**, 441–446 (2017).
 - [7] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shan-hui Fan, “Training of photonic neural networks through in situ backpropagation and gradient measurement,” *Optica* **5**, 864 (2018).
 - [8] 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 (2019).
 - [9] David A. B. Miller, “Establishing Optimal Wave Communication Channels Automatically,” *Journal of Lightwave Technology*, Vol. 31, Issue 24, pp. 3987–3994 **31**, 3987–3994 (2013).
 - [10] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley, “An Optimal Design for Universal Multiport Interferom-

- eters,” *Optica*, 1–8 (2016).
- [11] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani, “Experimental realization of any discrete unitary operator,” *Physical Review Letters* **73**, 58–61 (1994).
- [12] David A. B. Miller, “Self-aligning universal beam coupler,” *Optics Express* **21**, 6360 (2013).
- [13] 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* (2017) pp. 285–286.
- [14] Daniel Pérez, Ivana Gasulla, Lee Crudgington, David J. Thomson, Ali Z. Khokhar, Ke Li, Wei Cao, Goran Z. Mashanovich, and Jos Capmany, “Silicon RF-Photonics Processor Reconfigurable Core,” in *European Conference on Optical Communication, ECOC* (2018).
- [15] Daniel Pérez, Ivana Gasulla, Lee Crudgington, David J. Thomson, Ali Z. Khokhar, Ke Li, Wei Cao, Goran Z. Mashanovich, and Jos Capmany, “Multipurpose silicon photonics signal processor core,” *Nature Communications* (2017), 10.1038/s41467-017-00714-1.
- [16] Nicholas J. Russell, Levon Chakhmakhchyan, Jeremy L. O’Brien, and Anthony Laing, “Direct dialling of Haar random unitary matrices,” *New Journal of Physics* (2017), 10.1088/1367-2630/aa60ed.
- [17] Adolf Hurwitz, “über die Erzeugung der Invarianten durch Integration,” *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 71–72 (1897).
- [18] K. Zyczkowski and M. Kus, “Random unitary matrices,” *Journal of Physics A: General Physics* (1994), 10.1088/0305-4470/27/12/028.
- [19] Persi Diaconis and Peter J. Forrester, “Hurwitz and the origins of random matrix theory in mathematics,” *Random Matrices: Theory and Applications* **06**, 1730001 (2017).
- [20] 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), 10.1063/1.3672064.
- [21] David A. B. Miller, “Perfect optics with imperfect components,” *Optica* **2**, 747 (2015).
- [22] David A. B. Miller, “Self-configuring universal linear optical component [Invited],” *Photonics Research* **1**, 1 (2013).
- [23] 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* (2017) pp. 1733–1741.
- [24] 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**, 28236 (2017).
- [25] David A. B. Miller, “Setting up meshes of interferometers reversed local light interference method,” *Optics Express* **25**, 29233 (2017).
- [26] Fulvio Flamini, Nicol Spagnolo, Niko Viggianiello, Andrea Crespi, Roberto Osellame, and Fabio Sciarrino, “Benchmarking integrated linear-optical architectures for quantum information processing,” *Scientific Reports* **7**, 15133 (2017).
- [27] Diederik P Kingma and Jimmy Lei Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations* (2015).
- [28] 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* (Savannah, GA, 2016) pp. 265–283.
- [29] Alexander Y. Piggott, Jan Petykiewicz, Logan Su, and Jelena Vučković, “Fabrication-constrained nanophotonic inverse design,” *Scientific Reports* (2017), 10.1038/s41598-017-01939-2.
- [30] Rui Tang, Takuo Tanemura, Samir Ghosh, Kei-jiro 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**, 1798 (2018).
- [31] 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 MachZehnder interferometer,” *Optics Letters* **41**, 5318 (2016).
- [32] Alan Genz, “Methods for Generating Random Orthogonal Matrices,” *Monte Carlo and Quasi-Monte Carlo Methods*, 199–213 (1998).
- [33] Michael Mathieu and Yann LeCun, “Fast Approximation of Rotations and Hessians matrices,” *arXiv preprint* (2014).
- [34] Guillaume Labroille, Bertrand Denolle, Pu Jian, Jean Francois 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).
- [35] Guillaume Labroille, Pu Jian, Nicolas Barré, Bertrand Denolle, and Jean-Francois Morizur, “Mode Selective 10-Mode Multiplexer based on Multi-Plane Light Conversion,” *Optical Fiber Communication Conference* (2016), 10.1364/OFC.2016.Th3E.5.
- [36] Jean-Francois 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**, 2524 (2010).
- [37] Maria Schuld and Nathan Killoran, “Quantum Machine Learning in Feature Hilbert Spaces,” *Physical Review Letters* **122**, 040504 (2019).
- [38] Maria Schuld, Alex Bocharov, Krysta Svore, and Nathan Wiebe, “Circuit-centric quantum classifiers,” *arXiv preprint* (2018).
- [39] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicols Quesada, and Seth Lloyd, “Continuous-variable quantum neural networks,” *arXiv preprint* (2018), 10.1016/j.pepi.2014.06.011.
- [40] Rumen Dangovski, Li Jing, and Marin Soljačić, “Rotational Unit of Memory,” in *International Conference on Learning Representations* (2017).
- [41] 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, 024004 (2019).

Appendix A: Software

To reproduce the results of this paper, 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.

Appendix B: 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 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 [21]. The unitary matrix that we implement in presence of beamsplitter errors becomes

$$\begin{aligned} U_\epsilon &:= R_\phi B_{\epsilon_2} R_\theta B_{\epsilon_1} \\ t_\epsilon &:= |U_{\epsilon,12}|^2 = |U_{\epsilon,21}|^2 \\ r_\epsilon &:= |U_{\epsilon,11}|^2 = |U_{\epsilon,22}|^2. \end{aligned} \quad (\text{B1})$$

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 :

$$\begin{aligned} t_\epsilon &= 4|\rho|^2|\tau|^2t \\ &= 4t \left(\frac{1}{2} + \frac{\epsilon}{2} \right) \left(\frac{1}{2} - \frac{\epsilon}{2} \right) \\ &= t(1 - \epsilon^2). \end{aligned} \quad (\text{B2})$$

Similarly, we can solve for r_ϵ :

$$r_\epsilon = 1 - t_\epsilon = r + t \cdot \epsilon^2. \quad (\text{B3})$$

As we have discussed in this paper (and as we later show in Figure 12), 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 [21, 24]. 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 [7].

Appendix C: Haar measure

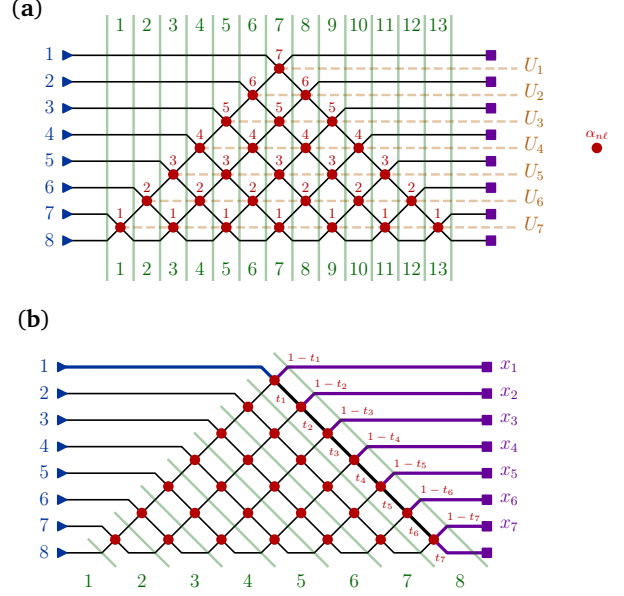


FIG. 8. Triangular mesh for $N = 8$ using (a) $2N - 3$ vertical layers ℓ showing the sensitivity index $\alpha_{n\ell}$ and (b) N diagonal layers m showing the transmissivity basis (t_n in red) and the measurement basis (x_n in purple).

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. [16]. 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. [11]. In this representation, we define the triangular mesh U_T as

$$\begin{aligned} U_T &= \prod_{m=0}^{N-1} U^{(N-m)} \\ U^{(m)} &= \prod_{n=1}^{m-1} U_{N-n}(\theta_{N-n}^{(m)}, \phi_{N-n}^{(m)}) \cdot D_m(\gamma_{N-m+1}), \end{aligned} \quad (\text{C1})$$

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 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 8(b). As mentioned in Refs. [11, 22], 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. [16] 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. [16], 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 .⁹ 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):

$$d\mu(U^{(m)}) \propto d\gamma_{N-m} \prod_{n=1}^{m-1} dx_n \prod_{n=1}^m d\phi_n. \quad (\text{C2})$$

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 8(b):

$$x_n = (1 - t_n) \prod_{k=1}^{n-1} t_k. \quad (\text{C3})$$

Intuitively, Equation C3 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 C3 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 $n \leq N - 1$

$$\mathcal{J}_{nn} = \frac{\partial x_n}{\partial t_n} = - \prod_{k=1}^{n-1} t_k. \quad (\text{C4})$$

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 C3.

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

$$\begin{aligned} d\mu(U^{(m)}) &\propto d\gamma_{N-m+1} \det \mathcal{J} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \\ &= d\gamma_{N-m+1} \prod_{n=1}^{m-1} \mathcal{J}_{nn} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \quad (\text{C5}) \\ &\propto d\gamma_{N-m+1} \prod_{n=2}^{m-1} t_{n-1}^{m-n} \prod_{n=1}^{m-1} dt_n \prod_{n=1}^m d\phi_n \end{aligned}$$

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 $\alpha_{n\ell} = N - n$ for a triangular mesh in Equation C5 (independent of ℓ), which can be seen using Figure 8. 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:

$$\begin{aligned} d\mu(U) &= \prod_{n=1}^N d\mu(U^{(n)}) \\ &= \prod_n d\gamma_n \prod_{n,\ell} \mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell}) dt_{n\ell} d\phi_{n\ell} \quad (\text{C6}) \\ &= \prod_n d\gamma_n \prod_{n,\ell} \mathcal{P}_{\alpha_{n\ell}}\left(\frac{\theta_{n\ell}}{2}\right) d\theta_{n\ell} d\phi_{n\ell} \end{aligned}$$

We can now normalize Equation C5 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 $\mathcal{P}_{\alpha_{n\ell}}(\theta_{n\ell}/2)$:

$$\begin{aligned} \mathcal{Q}_{\alpha_{n\ell}}(t_{n\ell}) &= \alpha_{n\ell} t_{n\ell}^{\alpha_{n\ell}-1} \\ \mathcal{P}_{\alpha_{n\ell}}\left(\frac{\theta_{n\ell}}{2}\right) &= \alpha_{n\ell} \sin\left(\frac{\theta_{n\ell}}{2}\right) \left[\cos\left(\frac{\theta_{n\ell}}{2}\right)\right]^{2\alpha_{n\ell}-1}. \quad (\text{C7}) \end{aligned}$$

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 $\theta_{n\ell}$:

$$\xi_{n\ell} = \left[\cos\left(\frac{\theta_{n\ell}}{2}\right)\right]^{2\alpha_{n\ell}} = t_{n\ell}^{\alpha_{n\ell}}. \quad (\text{C8})$$

Finally, as explained in Ref. [16], we can use the Clements decomposition [10] 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 C7 and C8 respectively.

⁹ This prefactor is exactly $\int_0^\infty dP e^{-P} P^{m-1} = (m-1)!$.

Appendix D: Unitary 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 9.

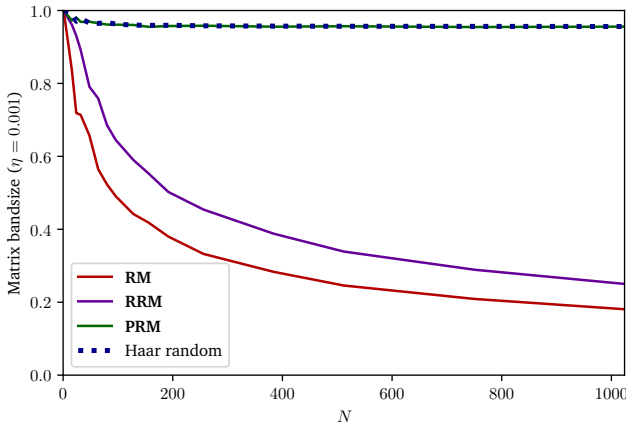


FIG. 9. Given $\eta = 0.001$, we compare bandsizes for rectangular ($U \sim \mathcal{U}_R(N, N)$), permuting rectangular ($U \sim \mathcal{U}_{PR}(N)$), and redundant meshes ($U \sim \mathcal{U}_R(N, 2N)$). Permuting rectangular meshes match the bandsize of Haar random matrices.

Appendix E: Introducing photonic errors 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 10, similar to what we observed for the rectangular and permuting rectangular meshes in Figure 7. 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.

Appendix F: Photonic singular value decomposition simulations

We compare the simulated performance of such rectangular and permuting rectangular architectures in the

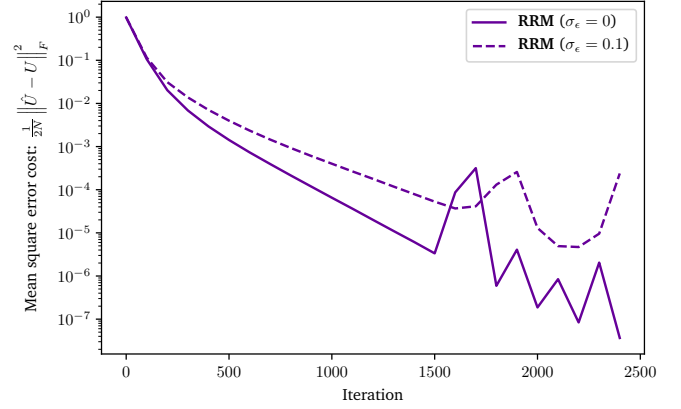


FIG. 10. A comparison of test mean square error for $N = 128$ between redundant rectangular meshes with error ϵ for 256-layer mesh for: 20000 iterations, Adam update, learning rate of 0.0025, batch size of 256, simulated in `tensorflow`.

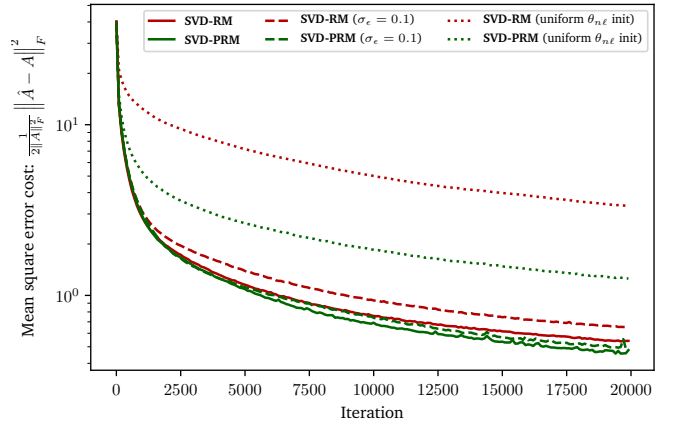


FIG. 11. A comparison of test mean square error for $N = 64$ between SVD devices using rectangular (SVD-RM) and permuting rectangular (SVD-PRM) meshes for: 20000 iterations, Adam update, learning rate of 0.005, batch size of 128, simulated in `tensorflow`. Unless otherwise noted, the default setting is Haar random initialized $\theta_{n\ell}$ with $\sigma_\epsilon = 0$.

singular value decomposition (SVD) configuration discussed in Refs. [6, 22]. 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. [22], the total com-

plexity 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

$$\begin{aligned} \mathcal{L}_{\text{test}} &= \frac{N\|\hat{A} - A\|_F^2}{2\|A\|_F^2} \\ \mathcal{L}_{\text{train}} &= \|\hat{A}X - AX\|_F^2, \end{aligned} \quad (\text{F1})$$

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

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 [7] 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 11. Both optimizations are minimally affected by beamsplitter error, unlike what is seen in the unitary optimization case.

Appendix G: Periodic parameters

We comment on our reported values of $\theta_{n\ell}$ in the checkerboard plots in Figures 3 (of the main text) and 12. 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 corresponds to the transformation (assuming $\theta_{n\ell}$ is originally between 0 and 2π):

$$\theta_{n\ell} \rightarrow \begin{cases} \theta_{n\ell} & \theta_{n\ell} \leq \pi \\ 2\pi - \theta_{n\ell} & \theta_{n\ell} > \pi \end{cases}. \quad (\text{G1})$$

Note a similar treatment as Equation G1 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:

$$\xi(\tilde{\xi}) = \begin{cases} \tilde{\xi} & \tilde{\xi} \leq 1 \\ 2 - \tilde{\xi} & \tilde{\xi} > 1 \end{cases}. \quad (\text{G2})$$

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 G1 and G2 based on symmetry arguments.

Appendix H: Training simulation comparisons

In Figure 12, we compare the performance for our unitary network experiment over our aforementioned conditions in Section V. 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.¹⁰

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 II.

Our simulated permuting rectangular implementation uses the same layer definitions as defined in Equation 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 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 11 gives improvements over the rectangular mesh.

Appendix I: 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 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. [16], 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

¹⁰ See <https://av.tib.eu/series/520/photonic+optimization>.

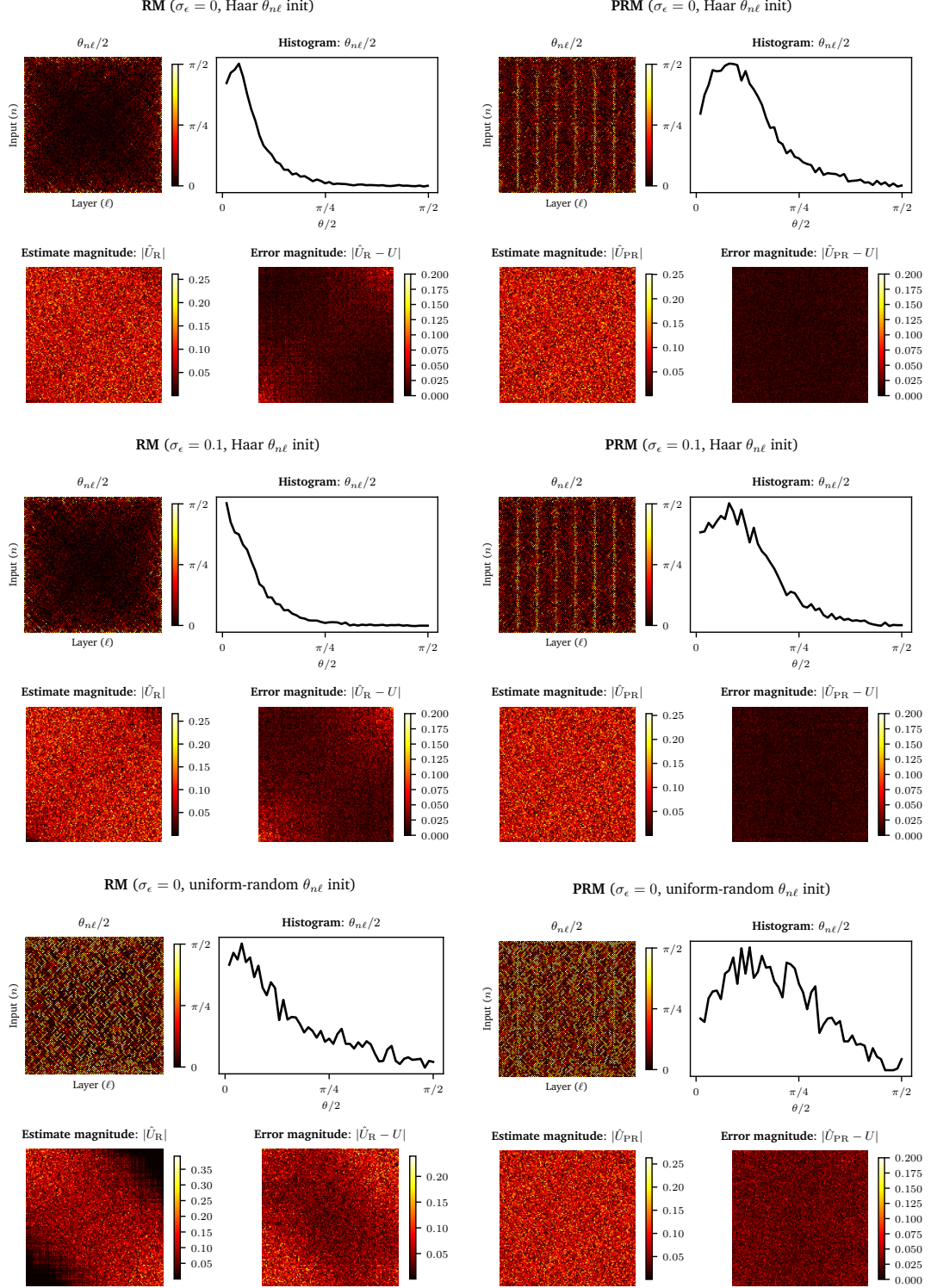


FIG. 12. Comparison of learned matrix errors and learned $\theta_{n\ell}$ weights after 20000 iterations for the Adam update at learning rate 0.0025 and batch size 256 for the simple unitary network. We consider two meshes: (1) rectangular mesh (RM), and (2) permuting rectangular mesh (PRM). We consider 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.

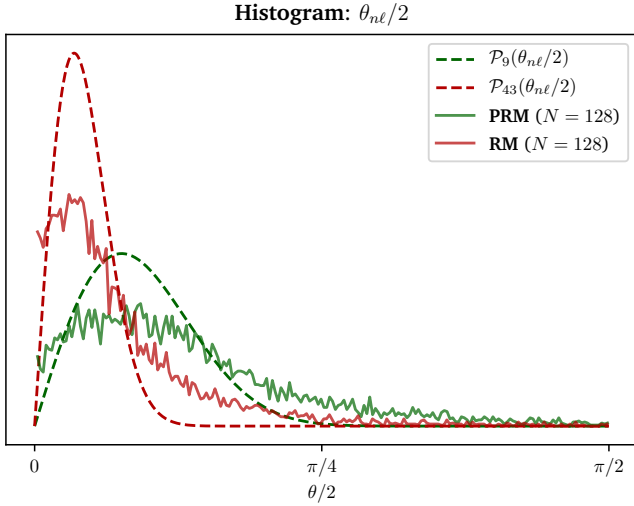


FIG. 13. Comparison of learned, normalized $\theta_{n\ell}$ distributions for $N = 128$ rectangular (RM) and permuting rectangular (PRM) meshes with $\mathcal{P}_\alpha(\theta/2)$ PDFs for $\alpha = \frac{N+1}{3} = 43$ (the average sensitivity index) and $\alpha = \lfloor \frac{N}{2 \log N} \rfloor = 9$ respectively. Note that permuting meshes have a larger tolerance, which eventually results in faster mesh optimization.

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 $2 \leq k_{\text{even}} \leq d(n, \ell)$, as defined in [16]. 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 8) $\alpha_{n\ell} := N - n$, so we wish to show that $N - n = |I_{n\ell}| + |O_{n\ell}| - N - 1$, or:

$$2N + 1 = |I_{n\ell}| + |O_{n\ell}| + n. \quad (\text{I1})$$

Suppose Equation I1 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 I1 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 I1 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. [16]. 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 14. 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 14, 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:

$$d(x) = \begin{cases} 2x - 1 & x \leq \lfloor \frac{N}{2} \rfloor \\ 2(N - x) & x > \lfloor \frac{N}{2} \rfloor \end{cases}. \quad (\text{I2})$$

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. [16] to an explicit lattice form:

$$s_x[y] = \begin{cases} 2(\lfloor \frac{N}{2} \rfloor - y) + 1 & y \leq \lfloor \frac{N}{2} \rfloor \\ 2(y - \lfloor \frac{N}{2} \rfloor) & y > \lfloor \frac{N}{2} \rfloor \end{cases}. \quad (\text{I3})$$

In this diagonal basis, we want to show that

$$d(x) + 1 - s_x[y] = |I_{xy}| + |O_{xy}| - N - 1. \quad (\text{I4})$$

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 14. 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 I4 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 I. 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 I4 remains satisfied.

Next, we induct across the $x, y = \lfloor \frac{N}{2} \rfloor$ boundaries, shown in Table I. 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 I4.

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

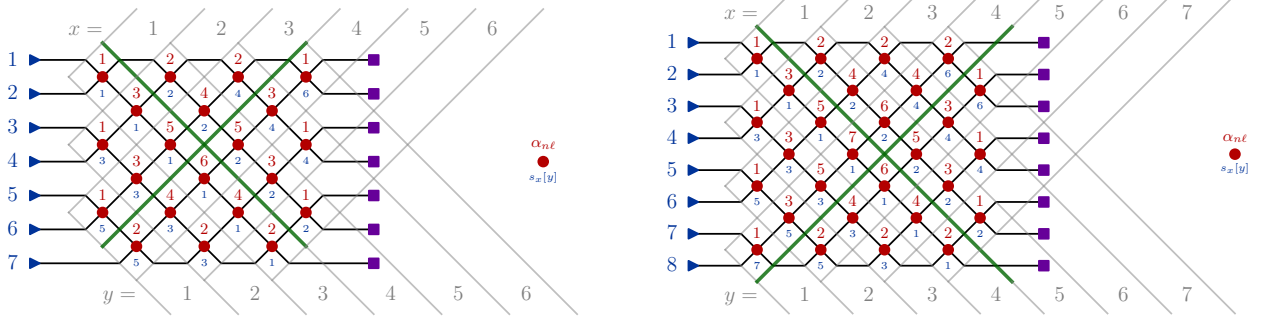


FIG. 14. Rectangular decomposition for even ($N = 8$) and odd ($N = 7$) meshes, showing the diagonal x, y basis. Values for $\alpha_{n\ell}$ are shown in red above each MZI, with values for $s_x[y]$ shown in blue below. The critical boundaries of $x, y = \lfloor \frac{N}{2} \rfloor$ separating the different quadrants are drawn in green. (Boundaries are offset for visual clarity.)

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 I. Induction on x and y within each of the quadrants in the mesh.

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