Symplectic Reck: In-Situ Learning of Gaussian Quantum Operations

Janet F. Zhong[1] **Renwen Yu**[1,2] **Paul-Alexis Mor**[3,4] **Charles Roques-Carmes**[3]
Aviv Karnieli[5] **David A. B. Miller**[1] **Shanhui Fan**[1]
[1]Stanford [2]CNRS [3]ISTA [4]Corps des Mines [5]Technion

Abstract

Programmable photonic circuits are a natural platform for physical learning, where hardware parameters are trained directly from measured feedback. We propose a physical learning algorithm and architecture for continuous-variable quantum systems where a photonic circuit is trained in situ to invert an unknown lossless Gaussian quantum operation. We prove that a generic symplectic matrix $S \in \mathrm{Sp}(2N, \mathbb{R})$ admits a decomposition into a triangular array of two-mode symplectic gates on neighboring modes, followed by single-mode symplectic gates. We then introduce the symplectic Reck mesh, which physically realizes this factorization, generalizing the unitary Reck mesh widely used in optical neural networks. When placed after an unknown Gaussian unitary, the mesh is trained in situ by nulling local response blocks one at a time using optical probes and measurements. We identify a local condition number that quantifies the squeezing required for a nulling step and its sensitivity to measurement error. We numerically demonstrate finite-shot training and robustness to systematic gate-calibration errors. Our work extends self-configuring optical networks from unitary to symplectic transformations, bringing physical learning to continuous-variable quantum photonic hardware.[1]

1 Introduction

Physical learning trains a physical system directly from measured responses, rather than learning a digital model and compiling it onto the hardware afterward [1–4]. Training from measured responses can compensate for fixed calibration errors. In photonics, an example is *self-configuring optics* [5], where a programmable optical mesh is tuned using measurements of its response to optical probes. Self-configuring photonic circuits have been developed for undoing unwanted mode mixing [6, 7], finding optimal communication channels [8], spectral filtering [9, 10], and variational optical diagonalization [11–14]. Optical inversion or mode unscrambling can be useful for recovering fields, images, or communication channels after a multimode system has mixed them [15, 16].

Many self-configuring meshes are based on the unitary Reck architecture, which realizes an arbitrary unitary transformation using a triangular array of two-mode unitary gates together with single-mode phase shifts [5, 17, 18]. These circuits are built from phase shifters and beam splitters and perform passive (photon-number-conserving) linear optics. Continuous-variable quantum optics also involves squeezing, which reduces quantum uncertainty in one field quadrature while increasing it in the conjugate quadrature. Squeezing is an active (photon-number-nonconserving) operation. Both passive mode mixing and squeezing belong to the class of lossless Gaussian operations [19–22]. The linear part of a lossless Gaussian operation is described by a real symplectic matrix. The unitary Reck mesh cannot implement general symplectic transformations. Extending self-configuration to symplectic transformations would allow an optical mesh to implement or reverse both mode mixing

[1]Code is available at `https://github.com/janetzhong/symplectic-reck`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and squeezing. This could help correct unwanted mode coupling in continuous-variable quantum communication and squeezed-light processing [23–25].

In this paper, we develop a method for learning the inverse of an unknown lossless Gaussian operation directly in hardware. A conventional approach would be to first characterize the unknown transformation using tomography, where one sends optical probes to estimate the full symplectic matrix of the lossless Gaussian operation [26–30]. From this, one can compute the inverse of the measured symplectic matrix digitally and use this to decompose the inverse into available optical gates. One would then need to compile the resulting parameters into the hardware [31–34]. This route is natural when the hardware parameters are accurately calibrated. In practice, this is rarely the case as fabrication variations, unintended phase shifts, heat transfer between nearby components (thermal crosstalk), inaccurately calibrated squeezing strengths and more can cause implemented gates to differ from their intended settings [35–37]. An in-situ method instead programs the inverse from feedback measurements of the system. This allows for the training of the inverse to account for static, repeatable implementation errors, similar to other self-configuring photonic meshes [38].

We introduce the *symplectic Reck mesh*, a triangular array of two-mode symplectic gates followed by a layer of single-mode symplectic gates. Placed after an unknown device S_{unk}, the mesh, with transformation S_θ, is trained so that $M = S_\theta S_{\mathrm{unk}} \approx I$. The training is performed by sequentially nulling 2×2 quadrature-response blocks, setting one gate at a time from local measurements of the system. Gate settings are obtained either by optimization using measured feedback or by a closed-form update calculated from optical probe measurements. Both approaches use fresh measurements as training proceeds, so later steps can partly compensate for errors in earlier gates.

Contributions.

1. **Physical learning task.** We formulate in-situ inversion of an unknown lossless Gaussian operation as a physical learning problem. The learned object is the trained circuit itself rather than a digital model of S_{unk}.
2. **Theory and architecture.** We show that the Reck factorization extends to generic matrices in $\mathrm{Sp}(2N, \mathbb{R})$ through sequential two-mode block nulling on neighboring modes and a final layer of single-mode gates. We identify exactly when each local nulling step is possible and derive a condition number κ that quantifies how difficult it is. We propose a symplectic Reck mesh that realizes this factorization in hardware.
3. **Local training algorithm.** We show how to carry out this factorization using only measured responses of the physical system. Each gate is set either by feedback optimization of a measured loss or by a closed-form update from response blocks estimated with optical probes and measurements.
4. **Experiments.** We numerically show that the condition number κ tracks trends in residual error as we vary the mode number, as the target approaches a nulling failure, and as we vary the squeezing strength of the target matrix. We evaluate the effects of finite-shot noise and compare measured closed-form updates, gradient-based feedback, and gradient-free feedback up to $N = 30$. We also show that in-situ nulling can outperform tomography followed by offline compilation in the presence of gate-calibration errors.

2 Background and problem setup

We first review the continuous-variable concepts used in this paper and then define the in-situ inversion task.

Quadratures and Gaussian operations. Each optical mode m is described by two quadratures, x_m and p_m, which correspond to the real and imaginary parts of the field amplitude. For N modes we can define the quadrature vector $r = (x_1, p_1, \ldots, x_N, p_N)^\top$. A Gaussian operation transforms the mean quadratures as $\langle r_{\mathrm{out}} \rangle = S \langle r_{\mathrm{in}} \rangle + d$, where S is a real $2N \times 2N$ matrix and d is a fixed displacement vector that shifts the mean quadratures. Not every matrix S is physically allowed. The set of physically allowable S are those that preserve the canonical commutation relations. This holds exactly when $S^\top J_{2N} S = J_{2N}$, where $J_{2N} = \bigoplus_{m=1}^{N} J_2$ is the symplectic form and $J_2 = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. Matrices satisfying this condition are called *symplectic* and form the group $\mathrm{Sp}(2N, \mathbb{R})$ [19–22]. Symplectic matrices can be built from three kinds of elementary optical components. Phase shifters rotate

the quadratures of individual modes, beam splitters mix quadratures between modes and squeezers increase the uncertainty in one quadrature while decreasing it by the same factor in the conjugate quadrature. Phase shifters and beam splitters are orthogonal and symplectic (orthosymplectic) in quadrature space and squeezers are symplectic but not orthogonal.

Task and access model. We consider an unknown lossless Gaussian operation $S_{\mathrm{unk}} \in \mathrm{Sp}(2N, \mathbb{R})$, with unknown displacement d, followed by a programmable mesh S_θ. Our goal is to choose mesh settings in S_θ so that the combined transformation $M = S_\theta S_{\mathrm{unk}} \approx I$, and then to cancel the remaining displacement. The learned S_θ performs the inverse of S_{unk} up to measurement and implementation errors. We assume the unknown device stays fixed during training. To learn the settings, we use optical probes through the combined system and measure its quadrature response [21, 22, 39]. The gates that have not been set are set to identity. We assume the response is measured directly after the gate that is being set. This is similar to the local monitors used in unitary meshes [6, 7].

3 Related work

Self-configuring photonics and in-situ inversion. Self-configuring photonics uses measured feedback to tune an optical circuit directly [5]. Sequential in-situ nulling with two-mode unitary gates has been demonstrated experimentally to undo unwanted mode mixing and separate optical beams [6, 7]. These tasks are closely related to the unitary analogue of our in-situ inverse task, $U_\theta U_{\mathrm{unk}} \approx I$, but may leave residual output phase shifts, which do not affect channel separation. Optically inverting the mode mixing in a multimode fiber can enable imaging through it, with potential applications in optical communications and computing [15, 16]. Removing unwanted mode mixing is also relevant to continuous-variable quantum communication and squeezed-light processing [23–25]. Our task extends in-situ inversion from $\mathrm{U}(N)$ to $\mathrm{Sp}(2N, \mathbb{R})$, allowing squeezing.

Learning Gaussian transformations and compiling inverses. A conventional approach to inverting an unknown optical operation is to reconstruct the transformation using tomography [26–30], compute its inverse, decompose it into available gates, and program those gates into the hardware [31–34]. The final step can be affected by calibration errors, including fabrication variations, phase offsets, thermal crosstalk, and squeezer miscalibration [35–37]. We typically need detailed models of how the hardware responds to control settings [36, 37, 40] to account for such errors. Self-configuring algorithms use feedback to adjust circuit parameters, which can avoid the need for detailed calibration models and simplify experimental methods.

Symplectic QR-like factorizations. We derive a symplectic QR-like factorization that generalizes the Reck construction [5, 17], decomposing a generic matrix in $\mathrm{Sp}(2N, \mathbb{R})$ into two-mode $\mathrm{Sp}(4, \mathbb{R})$ gates on neighboring modes and single-mode $\mathrm{Sp}(2, \mathbb{R})$ gates. Similar algorithms for the symplectic QR factorization, using local symplectic transformations, were derived earlier in Refs. [41, 42]. Here, we give a different construction of each nulling step, prove that our nulling sequence succeeds for generic symplectic matrices, and show that the factorization can be realized and trained in hardware. Other symplectic QR-like and SR decompositions exist but differ from ours in their objectives and resulting matrix forms [43–46].

4 The symplectic Reck mesh

We now describe the mesh architecture, its gate training order, and how each gate is set from measured responses.

Block notation. Because we use the quadrature ordering $r = (x_1, p_1, \ldots, x_N, p_N)^\top$ which is grouped by mode, we can view the $2N \times 2N$ response M as an $N \times N$ grid of 2×2 blocks, with one block row and one block column per mode. Block $M_{j,k} = \begin{pmatrix} M_{x_j,x_k} & M_{x_j,p_k} \\ M_{p_j,x_k} & M_{p_j,p_k} \end{pmatrix}$ describes how the output quadratures of mode j respond to the input quadratures of mode k. Here, M_{x_j,p_k} is the response of output x_j to input p_k.

Architecture. The mesh contains a triangular array of $N(N-1)/2$ two-mode symplectic gates acting on neighboring modes, followed by one single-mode symplectic gate per mode (Fig. 1). Each

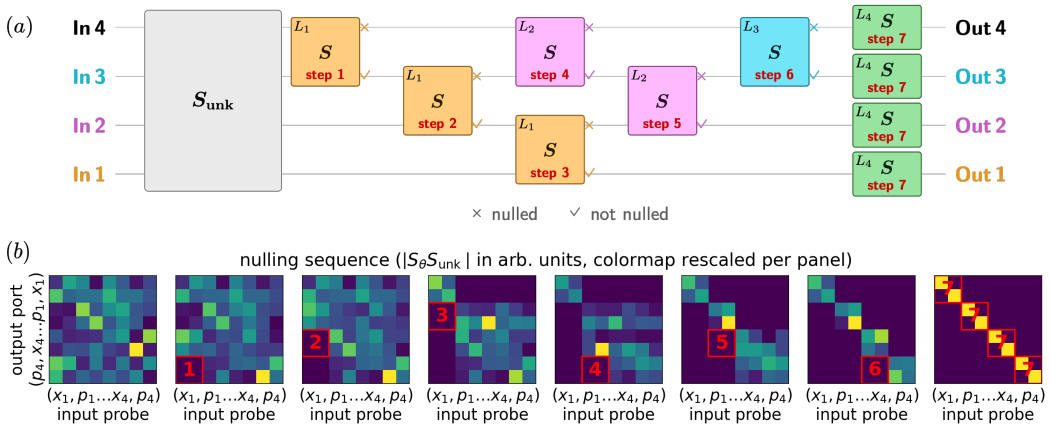


Figure 1: (a) Symplectic Reck mesh for $N = 4$ modes. Red step numbers show the gate training order. The final single-mode gates can be trained in parallel. (b) Simulated response $|S_\theta S_{\text{unk}}|$ during the training steps. Red boxes mark the blocks being nulled and correspond to the step numbers in (a). Two-mode gates bring the response close to block diagonal, and single-mode gates bring it close to the identity, with a final residual of 1.1×10^{-4}. Each panel uses its own color scale.

Algorithm 1 In-situ training of the symplectic Reck mesh. Gate settings are obtained using either measured feedback or closed-form updates from measured response blocks.

Set every gate to the identity, so $S_\theta = I$.
for $k = 1, \ldots, N-1$ **do** ▷ block columns, left to right
  for $j = N, \ldots, k+1$ **do** ▷ blocks in column k, bottom to top
    Probe input mode k and estimate $\widehat{M}_{j,k}$ (and $\widehat{M}_{j-1,k}$ for closed-form training).
    Set the two-mode gate on modes $(j-1, j)$ so that $\widehat{M}_{j,k} \approx 0$.
  end for
end for
for $m = 1, \ldots, N$ **do** ▷ blocks along diagonal, parallelizable
  Set the single-mode gate on mode m so that $\widehat{M}_{m,m} \approx I_2$.
end for
Measure the output mean with no input displacement and cancel it.
return the trained mesh S_θ, with $S_\theta S_{\text{unk}} \approx I$.

two-mode gate can implement any element of $\mathrm{Sp}(4, \mathbb{R})$. In hardware, it can be built from two passive two-mode interferometers with two single-mode squeezers between them (a Bloch–Messiah decomposition [22, 31, 32]). Each single-mode gate is a squeezer between two phase shifters. Appendix A gives explicit parameterizations and illustrates these implementations in Fig. 4.

Gate training order. We null quadrature-response blocks of M one gate at a time, leaving each gate fixed after training. The nulling order goes through block columns $k = 1, \ldots, N-1$, nulling the blocks below the diagonal from bottom to top. The gates used for column k form layer L_k in the triangular array. A two-mode gate acts on modes $(j-1, j)$ to null $M_{j,k}$, in the order $j = N, N-1, \ldots, k+1$. With exact nulling, this order preserves the zeros in earlier columns. Once all blocks below the diagonal are zero, symplecticity forces the blocks above the diagonal to also be zero (Appendix B.2). The single-mode gates then invert the remaining diagonal blocks. We denote the measured estimate of each response block by $\widehat{M}_{j,k}$. Algorithm 1 summarizes the procedure and Fig. 1 depicts it for four modes. Any remaining displacement can be removed by applying an opposite displacement to each output mode (not shown in Fig. 1) [47].

Measuring a response block. We use coherent optical pulses (essentially laser pulses) as probes. To estimate M_{x_j,p_k}, we set the mean input p_k to $+\eta$ and $-\eta$ and measure the mean output x_j, giving $\widehat{M}_{x_j,p_k} = (\langle x_j \rangle_{+\eta p_k} - \langle x_j \rangle_{-\eta p_k})/(2\eta)$. Taking the difference cancels the unknown displacement.

The other three entries of $M_{j,k}$ are measured in the same way. Shifting either x_k or p_k by $\pm\eta$ gives four inputs. Measuring x_j and p_j separately for each input using homodyne detection gives eight measurement settings. Heterodyne detection instead measures both output quadratures together [21, 22, 39].

Measured closed-form or feedback-based updates. The algorithm can use either feedback training or measured closed-form updates. In *feedback* training, an optimizer adjusts the gate's controls to minimize a measured loss: $\|\widehat{M}_{j,k}\|_F^2$ for a two-mode gate, and $\|\widehat{M}_{m,m} - I_2\|_F^2$ for a single-mode gate in the final layer. This does not require a calibration model of the gate, only the ability to change its controls and observe the result. In *closed-form* training, we measure both blocks in the two active rows, $\widehat{M}_{j-1,k}$ and $\widehat{M}_{j,k}$, and compute the nulling gate directly from them (Section 5). For the final layer, we instead invert each measured diagonal block $\widehat{M}_{m,m}$ after projecting it onto a symplectic matrix (Appendix G). This avoids iterative optimization but requires a calibration model to set the current gate. In both methods, each update uses only local 2×2 blocks, independent of N. The effects of earlier gates are measured rather than predicted from a model, so later gates can partly compensate for their errors (Section 6). For fixed shot and evaluation budgets per gate, training uses $O(N^2)$ measurement shots in total (Appendix E).

5 Theory

We now establish when the symplectic Reck mesh succeeds. We first determine when a two-mode gate can null a response block, then prove that the full sequence succeeds for generic symplectic matrices. We also derive a local condition number that quantifies how difficult each nulling step is.

5.1 When is a nulling step possible?

Consider a step that aims to set the block $M_{j,k}$ to zero. We apply a gate to output modes $j-1$ and j, which mixes the corresponding two block rows of M. In column k, let

$$P = M_{j-1,k}, \qquad E = M_{j,k}, \tag{1}$$

where E is the block we want to eliminate and P is the block directly above it. We seek a two-mode symplectic gate G such that

$$G \begin{pmatrix} P \\ E \end{pmatrix} = \begin{pmatrix} Q \\ 0 \end{pmatrix}. \tag{2}$$

Thus, G replaces E with zero while allowing P to change to a new block Q.

In the unitary Reck mesh, a beam splitter and phase shifters can always make one of the two output responses zero. For a general symplectic response, this is not always possible. To describe when it fails, we define

$$\Delta = \det P + \det E, \tag{3}$$

and call it the *pivot*, as it plays a similar role to pivots in Gaussian elimination. Because symplectic gates preserve the symplectic form, Δ is unchanged by any two-mode symplectic gate (Appendix B.1). After nulling, the blocks become Q and 0, so $\det Q = \Delta$.

Local nulling criterion. If the two columns of $\left(\begin{smallmatrix} P \\ E \end{smallmatrix}\right)$ are linearly independent, nulling is possible if and only if $\Delta \neq 0$. To see why a zero pivot[2] prevents nulling, recall that a symplectic gate is invertible and therefore preserves linear independence. After nulling, the lower block is zero, so the columns of Q must remain linearly independent. Hence, Q must be invertible. This is incompatible with $\det Q = \Delta = 0$. Conversely, when $\Delta \neq 0$, the two response columns can be mapped into a single mode by a symplectic gate, as shown by the explicit construction in Appendix B.1. Similar nulling criteria were derived in Refs. [42, 49] using different explicit gates, assuming invertible P or invertible P and E, respectively. If the two columns are linearly dependent, there is at most one independent response to move into the upper mode. Nulling in this case is always possible using only a beam splitter and phase shifters. Thus, the only obstruction is a zero pivot with two independent response columns. Appendix B gives the full proof.

[2]Geometrically, a zero pivot with independent columns means that the columns span an isotropic subspace, on which the symplectic form vanishes [48].

5.2 When does the full sequence succeed?

A nonzero pivot means one nulling step succeeds. However, each gate changes the blocks used in later steps. We must therefore show that the full sequence does not generically hit a zero pivot.

Theorem 1 (Symplectic Reck decomposition)**.** *For a generic symplectic matrix $S_{\mathrm{unk}} \in \mathrm{Sp}(2N, \mathbb{R})$, the nulling sequence of Algorithm 1 succeeds, giving $S_\theta S_{\mathrm{unk}} = I$. Equivalently, a generic symplectic matrix can be implemented by a triangular mesh of two-mode symplectic gates on neighboring modes, followed by one single-mode symplectic gate on each mode.*

Here, generic means that the exceptions form a set of measure zero, so a randomly chosen target fails with probability zero. We give an outline of the proof. We show that the choice of gate at each step does not affect later pivots. This lets us follow an explicit gate construction, given in Appendix B.1, which turns every pivot into a ratio of polynomials in the entries of S_{unk}. A target can fail only if one of these polynomials is zero. To show that they are not all identically zero, we construct a passive target from $50{:}50$ beam splitters for which every pivot is strictly positive (Appendix B.4). The failing targets therefore form a set of measure zero. Since a mesh that inverts S_{unk}^{-1} equals S_{unk}, generic symplectic matrices also admit a symplectic Reck factorization.

The symplectic Reck mesh does not invert every symplectic matrix, but a modified version can. Appendix C gives one such construction, which adds passive gates before each nulling step to avoid zero pivots, similar to the passive rotations in the symplectic QR algorithm of Ref. [41].

5.3 Conditioning: how difficult is a nulling step?

The nonzero pivot condition above guarantees a nulling gate exists, but it does not say how much squeezing is needed or how sensitive the nulling step is to measurement errors. A small $|\Delta|$ may indicate that the nulling step is close to a failure point, but it can also mean that the whole response is small.

To distinguish these cases, we compare $|\Delta|$ with the overall size of the response, measured by the Euclidean area spanned by its two columns. Let $V = \binom{P}{E}$. For linearly independent response columns, we define the local condition number

$$\kappa = \frac{\sqrt{\det(V^\top V)}}{|\Delta|} \geq 1. \tag{4}$$

The columns of V are the responses to inputs x_k and p_k, and the numerator is the ordinary area of the parallelogram they span. The pivot Δ can be thought of as a symplectic area. When the columns are projected onto the (x, p) plane of each mode, they form parallelograms with signed areas $\det P$ and $\det E$, and their sum is Δ. Symplectic gates preserve this sum, although the individual areas can change. After nulling, both columns lie in one mode, so the ordinary area equals $|\det Q| = |\Delta|$. The condition number κ is the ratio of the ordinary area before nulling to this area after nulling. Because both areas scale in the same way, κ is also independent of the overall scale of the response, whereas $|\Delta|$ alone is not.

Passive gates preserve both areas, so when $\kappa = 1$, the nulling step can be achieved using beam splitters and phase shifters with no squeezers. A large κ means that $|\Delta|$ is small relative to the area spanned by the response columns. Nulling then requires strong squeezing and can amplify measurement errors. See Appendix D for more details.

Minimum-squeezing gate. It is possible to state the minimum squeezing needed for a nulling step. For any gate that completes the nulling step, at least one of the two single-mode squeezers in its Bloch–Messiah decomposition must have squeezing strength at least $r_{\min} = \frac{1}{2} \operatorname{arccosh} \kappa$. A passive two-mode transformation (built from beam splitters and phase shifters) followed by a two-mode squeezer of strength $r_{\min}$ can achieve this bound (Appendix D). We call this the *minimum-squeezing gate* and use it in all closed-form simulations in Section 6. It has only four control parameters, so a symplectic Reck mesh built from these gates has $N(2N + 1)$ parameters, which matches $\dim(\mathrm{Sp}(2N, \mathbb{R}))$ (Appendix D.3).

Noise sensitivity and sample complexity. The condition number κ also measures how strongly a nulling step amplifies noise. If a measured block differs from the true response by δ (in Frobenius

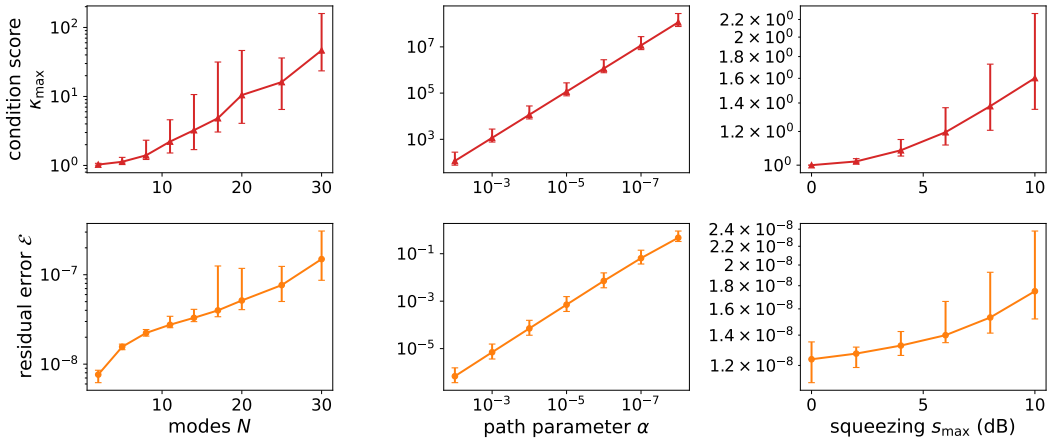


Figure 2: The condition number tracks how much the mesh amplifies small errors. Top row: $\kappa_{\max}$, the largest condition number along the nulling sequence. Bottom row: final residual when each measured block has a small random error of size $\epsilon = 10^{-8}$. Left: increasing the number of modes N. Middle: moving the target matrix toward one where nulling fails (smaller α is closer). Right: increasing the maximum target squeezing $s_{\max}$ for $N = 4$. The residual trend is similar to the $\kappa_{\max}$ trend in all three cases. Markers show medians, and error bars show the 25th–75th percentiles over 60 trials (left) or 30 trials (middle and right). See Appendix G for full experimental details.

norm), the squeezing used for nulling amplifies this error, and the minimum-squeezing gate leaves a residual of at most $\sqrt{\widehat{\kappa}}\,\delta$ in the block it was meant to null, where $\widehat{\kappa}$ is the condition number of the measured blocks (Appendix D.4). Averaging over q shots per setting reduces δ as $1/\sqrt{q}$, so the residual after one nulling step scales as $\sqrt{\kappa/q}$. Since κ depends only on P and E, it can be estimated from the measured responses and monitored during training. We report its largest value over all nulling steps, $\kappa_{\max}$. The closed-form method already measures both blocks, whereas feedback training would need an extra measurement of P. By combining the nulling-step bound over the full sequence, we can derive a conditional probably approximately correct (PAC) guarantee for closed-form training. Under the conditions stated in Appendix E.1, a polynomial number of shots suffices to reach a target residual with high probability.

6 Numerical experiments

We address the following questions in numerical simulations. First, does the condition number predict how much errors are amplified (Fig. 2)? Second, how well does training work with a finite number of measurement shots (Tables 1 and 2)? Third, what advantage does in-situ training offer over tomography followed by offline compilation (Fig. 3)?

Setup. Unless stated otherwise, we sample targets in Bloch–Messiah form, $S_{\text{unk}} = O_1 K(\mathbf{r}) O_2$, where O_1, O_2 are Haar-random passive transformations and $K(\mathbf{r})$ squeezes mode i by $r_i \sim \text{Uniform}[0, 0.5]$ [22, 31]. This gives up to 4.3 dB per mode, which is within experimentally demonstrated squeezing levels [50, 51]. We report the normalized residual $\mathcal{E} = \|S_\theta S_{\text{unk}} - I\|_F / \sqrt{2N}$, which vanishes for perfect inversion. Finite-shot simulations include noise in homodyne measurements, with each of the settings in Section 4 averaged over q shots. Appendix G gives the noise model and simulation details of all experiments.

Conditioning tracks error amplification (Fig. 2). To separate the effect of conditioning from optimizer behavior and shot noise, we use a standard sensitivity test based on small input perturbations [52]. We add a random error of fixed Frobenius norm $\epsilon = 10^{-8}$ to each measured block, compute the closed-form minimum-squeezing gate from the perturbed blocks, and apply it to the true response. We run three sweeps, shown in the three columns of Fig. 2. The first increases the number of modes N. The second follows paths $S(\alpha) = e^{\alpha L} S_{\text{iso}}$, where S_{iso} has an exactly failing pivot and L is a symplectic generator constructed from a random Bloch–Messiah target. This

Table 1: Local loss $\|M_{j,k}\|_F^2$ left by each two-mode gate for $N = 3$, measured on the exact response right after the gate is set. Here q is the number of shots per homodyne setting, and $q = \infty$ means noiseless measurements, where the closed-form loss is at machine precision. All four methods use matched total measurement budgets (Appendix G.3). Entries show the median [25th, 75th percentile] over 90 gates from 30 targets, and each power of ten applies to all three values.

q	closed-form	finite differences	dithering	genetic algorithm
10^2	$8.2\,[4.5, 12]\,10^{-6}$	$5.4\,[1.3, 17]\,10^{-3}$	$3.2\,[1.1, 7.9]\,10^{-3}$	$6.7\,[3, 15]\,10^{-3}$
10^3	$7.7\,[4.7, 13]\,10^{-7}$	$2.5\,[0.83, 5.3]\,10^{-4}$	$2.2\,[1.4, 3.6]\,10^{-4}$	$6.5\,[3.3, 12]\,10^{-4}$
10^4	$6.3\,[4.3, 12]\,10^{-8}$	$5.5\,[3, 12]\,10^{-5}$	$1\,[0.5, 1.9]\,10^{-4}$	$1.6\,[0.92, 2.4]\,10^{-4}$
10^5	$8.3\,[4.5, 12]\,10^{-9}$	$2.9\,[1.3, 7.8]\,10^{-5}$	$6.5\,[3.2, 12]\,10^{-5}$	$1.2\,[0.49, 1.8]\,10^{-4}$
∞	$7.2\,[3.1, 26]\,10^{-31}$	$2.2\,[1.2, 6.6]\,10^{-5}$	$6.3\,[3.3, 15]\,10^{-5}$	$9\,[5.8, 16]\,10^{-5}$

Table 2: Final full-mesh residual $\mathcal{E}$ for increasing N, with $q = 10^3$ shots per homodyne setting. All four methods use matched total measurement budgets (Appendix G.3). Entries show the median [25th, 75th percentile] over 30 targets.

N	closed-form	finite differences	dithering	genetic algorithm
3	$0.0011\,[0.00099, 0.0012]$	$0.035\,[0.024, 0.058]$	$0.057\,[0.033, 0.087]$	$0.056\,[0.038, 0.11]$
5	$0.0016\,[0.0016, 0.0019]$	$0.098\,[0.069, 0.12]$	$0.086\,[0.07, 0.11]$	$0.11\,[0.073, 0.16]$
7	$0.0022\,[0.0021, 0.0025]$	$0.12\,[0.097, 0.15]$	$0.12\,[0.1, 0.18]$	$0.14\,[0.1, 0.2]$
9	$0.0027\,[0.0023, 0.0029]$	$0.16\,[0.14, 0.23]$	$0.16\,[0.12, 0.18]$	$0.18\,[0.12, 0.23]$
11	$0.0032\,[0.0027, 0.0054]$	$0.2\,[0.17, 0.26]$	$0.16\,[0.14, 0.2]$	$0.18\,[0.15, 0.26]$
13	$0.004\,[0.0031, 0.011]$	$0.22\,[0.19, 0.28]$	$0.22\,[0.16, 0.24]$	$0.21\,[0.17, 0.3]$
15	$0.0046\,[0.0036, 0.013]$	$0.26\,[0.21, 0.29]$	$0.24\,[0.2, 0.29]$	$0.23\,[0.19, 0.29]$
20	$0.013\,[0.0053, 0.057]$	$0.32\,[0.27, 0.37]$	$0.32\,[0.24, 0.35]$	$0.3\,[0.25, 0.38]$
25	$0.021\,[0.0069, 0.061]$	$0.39\,[0.31, 0.44]$	$0.34\,[0.31, 0.36]$	$0.33\,[0.29, 0.38]$
30	$0.047\,[0.02, 0.22]$	$0.47\,[0.41, 0.52]$	$0.41\,[0.36, 0.45]$	$0.36\,[0.31, 0.4]$

construction preserves symplecticity. As α decreases toward zero, the target approaches S_{iso}. The third increases $s_{\max}$, the upper bound on the target's single-mode Bloch–Messiah squeezing in dB, at $N = 4$.

In all three sweeps, the final residual largely follows $\kappa_{\max}$. Near a zero pivot, both the residual and $\kappa_{\max}$ grow roughly as α^{-1}, which is steeper than the nulling-step factor $\sqrt{\kappa}$ discussed above because later gates in the sequence can further amplify errors left by earlier steps. Raising $s_{\max}$ to 10 dB at $N = 4$ increases the residual only mildly, whereas increasing N leads to substantially larger errors, consistent with error accumulation and amplification over more nulling steps. Conditioning also sets a hardware requirement through the largest local minimum-squeezing parameter over the two-mode nulling gates, $\frac{1}{2}\operatorname{arccosh}\kappa_{\max}$. For each target, we take the largest squeezing required by any nulling step. For targets with $r_i \sim \text{Uniform}[0, 0.5]$ (up to 4.3 dB per Bloch–Messiah mode), the median of this value increases from 1.1 dB at $N = 2$ to 19.7 dB at $N = 30$ (25th–75th percentiles 16.7–25.0 dB). This exceeds the squeezing levels available in current experimental setups [51, 53, 54]. Although this squeezing requirement becomes challenging at larger N, current squeezing-gate platforms involve only a few modes [55, 56]. At $N = 4$ with $s_{\max} = 10$ dB, the median requirement is about 4.5 dB, comparable to demonstrated squeezing gates. The squeezing requirement also grows near nulling failures. Monitoring κ during training can identify these poorly conditioned steps, and passive mode mixing before nulling can reduce the squeezing required for nulling (Appendix C).

Finite-shot training (Tables 1 and 2). We compare measured closed-form updates with three feedback methods using matched total measurement budgets. We estimate gradients using finite differences or dithering, which applies small oscillations to the gate parameters and uses demodulation (measuring how the loss responds to each oscillation) to extract gradients, as demonstrated in unitary meshes [12]. Both use Adam [57] for optimization. The third method is a gradient-free genetic algorithm. All three feedback methods need only the measured loss $\|\widehat{M}_{j,k}\|_F^2$. Other gradient methods include automatic differentiation in simulation, parameter-shift rules [58, 59], and in-situ backpropagation [60–62].

Table 1 compares the four methods on a single nulling step. For $N = 3$, the closed-form squared loss $\|M_{j,k}\|_F^2$ decreases roughly as $1/q$. The residual norm $\|M_{j,k}\|_F$ instead scales as $1/\sqrt{q}$, consistent

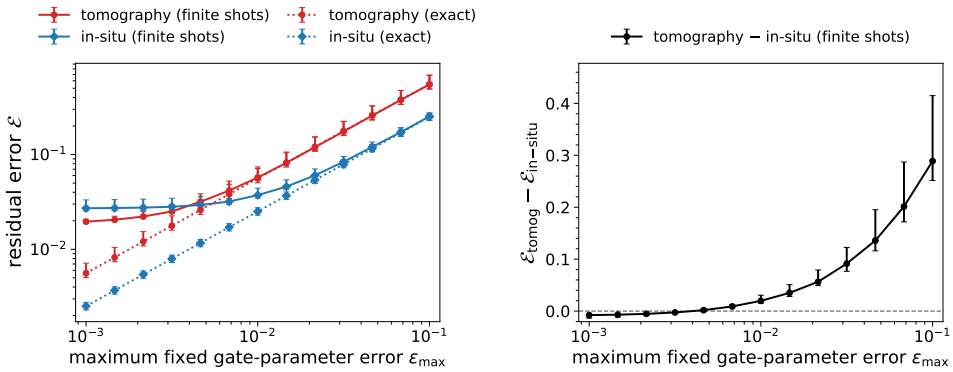


Figure 3: In-situ training is more accurate than tomography followed by offline compilation once fixed gate errors dominate measurement noise ($N = 10$). Each gate is followed by a small symplectic error whose parameters are drawn from $\mathrm{Uniform}[-\epsilon_{\max}, \epsilon_{\max}]$ and held fixed. Left: final residual $\mathcal{E}$. Solid curves use $q = 10^4$ shots per homodyne setting, with the same total budget of $8N^2q$ samples for both methods. Dotted curves use noiseless measurements with the same gate errors. Right: $\mathcal{E}_{\text{tomog}} - \mathcal{E}_{\text{in-situ}}$ for the finite-shot runs, where positive values mean in-situ training is better. Markers show medians and error bars span the 25th–75th percentiles over 30 targets.

with the noise-sensitivity bound in Section 5.3 for fixed conditioning. Dithering gives lower median local loss than finite differences at $q = 10^2$ and 10^3, but higher loss at larger q. The feedback optimizers retain a higher loss even without measurement noise ($q = \infty$), so their accuracy is limited by the optimizer or its budget rather than by shot noise. Table 2 reports full-mesh training for $N = 3$ to 30. The residual grows with N for all four methods. The median closed-form residual remains below 10^{-2} up to $N = 15$, then reaches 0.013 at $N = 20$ and 0.047 at $N = 30$. With noiseless measurements, the median stays between 1.6×10^{-15} and 1.4×10^{-14} for all N. The growth with N is consistent with the growth of $\kappa_{\max}$ in Fig. 2. In both tables, total shot budgets are matched to compare the methods. Because the closed-form method measures each gate once rather than iterating, it uses about 10^3 times more shots per setting than each feedback loss evaluation (Appendix G.3). In practice, the closed-form method may need far fewer shots to reach a useful residual. A practical approach may combine the methods, using the closed-form gate as a starting point and refining it with measured feedback to correct errors from an imperfect calibration model.

Fixed calibration errors (Fig. 3). We now test when in-situ training can outperform tomography followed by offline compilation. Offline compilation computes all gate settings from a hardware model and cannot account for unmodeled deviations from those settings. In-situ training instead measures the response after the real, imperfect gates have acted. To represent static errors such as fabrication variations and phase offsets, we add a small symplectic error to each gate, with each error parameter drawn from $\mathrm{Uniform}[-\epsilon_{\max}, \epsilon_{\max}]$ and fixed during each trial (Appendix G.4). The tomography baseline measures the full response once, projects it onto a symplectic matrix, and computes all gates offline. In-situ training computes each gate from blocks measured after the earlier gates are set, using sequential access (Section 2). Both use the closed-form minimum-squeezing gate and the same measurement budget. With noiseless measurements (dotted curves in Fig. 3), in-situ training gives about half the residual of tomography at every $\epsilon_{\max}$ we test. We also add a moderate level of shot noise ($q = 10^4$) to test when gate errors outweigh measurement noise.

For small gate errors, measurement noise dominates and tomography is slightly more accurate. In the left panel of Fig. 3, the median residuals at $\epsilon_{\max} = 10^{-3}$ are 0.020 for tomography and 0.027 for in-situ training. This advantage comes mainly from the global symplectic projection of the tomography estimate. Without it, the median tomography residual rises to 0.026. In-situ training has no global projection, but its gates are symplectic by construction. As $\epsilon_{\max}$ grows, gate errors overtake shot noise. The two methods cross over at $\epsilon_{\max} \approx 4 \times 10^{-3}$, and from $\epsilon_{\max} \approx 2 \times 10^{-2}$ onward the in-situ median residual is about half that of tomography, matching the noiseless trend. The right panel shows $\mathcal{E}_{\text{tomog}} - \mathcal{E}_{\text{in-situ}}$, the tomography residual minus the in-situ residual for each target. It stays close to zero for small gate errors and grows steadily once gate errors dominate.

7 Discussion and limitations

Limitations. A key obstacle to a hardware demonstration is likely to be loss, since the theory assumes lossless operations, whereas loss remains a major challenge for integrated continuous-variable platforms [50]. Uniform loss across modes does not prevent the trained mesh from separating the modes, but unequal loss leaves residual mixing. This mixing can be removed by equalizing the loss with dummy components [38], at the cost of lower transmission (Appendix F). Loss also removes quantum information and adds noise that no mesh can undo [20, 21]. Another difficulty is that, although nulling fails only for a measure-zero set of targets, targets near this set have near-zero pivots, which require strong squeezing and amplify measurement errors. Errors and the required squeezing also grow with N, which may limit scaling to larger meshes (Section 6). Adding extra passive gates before each nulling step can reduce the required squeezing, but it costs more components (Appendix C). If the response can only be measured at the output of the full mesh rather than directly after each gate, every measurement passes through the gates that have not yet been trained, which may also increase errors. Hardware error-correction methods developed for unitary meshes [38, 63–65] may also help with these challenges, and adapting them to symplectic meshes is an open direction.

Architecture variants. Other nulling meshes are possible. For example, a whole column could be nulled at once rather than one block at a time. Adding extra passive gates before each nulling step, similar to row pivoting in Gaussian elimination [66], can make the mesh universal [41] (Appendix C). Other architectures used for unitary and general linear meshes may also be worth exploring in the symplectic setting, such as rectangular layouts [67], SVD-style architectures [5], and more.

Beyond in-situ inversion. Self-configuring unitary meshes have been used for more than inversion. They can implement arbitrary linear transformations [5] and perform variational diagonalization of scrambled light [11–14]. Symplectic versions of these tasks, which would also allow squeezing, are a natural next step. More broadly, machine learning often uses structured factorizations and structure-preserving parameterizations of large linear transformations [68–76]. Unitary Reck meshes provide a photonic realization of this idea and are used as linear layers in optical neural networks [77, 78]. A symplectic Reck mesh may play a similar role in Gaussian quantum optics and other problems involving symplectic transformations [21, 79–81].

8 Conclusion

We introduced the symplectic Reck mesh for learning the inverse of an unknown lossless Gaussian operation through local optical measurements. The mesh extends the Reck architecture to include squeezing. We proved that sequential block nulling in the adjacent-mode Reck ordering succeeds for generic symplectic matrices and identified exactly when a nulling step fails. We showed that the condition number κ determines the minimum squeezing required for nulling and bounds the amplification of measurement errors in one nulling step. Our simulations demonstrate finite-shot training and show that in-situ updates can partly compensate for fixed gate errors, giving lower residuals than tomography followed by offline compilation when these gate errors are sufficiently large. Potential applications include compensating unwanted mode mixing and squeezing in continuous-variable optical links [6, 23, 24, 82]. More broadly, the architecture and training methods may enable new self-configuring tasks in continuous-variable quantum optics.

Acknowledgments and Disclosure of Funding

This work is supported by the Toyota Research Institute of North America. The authors are seeking patent protection for inventions related to this work (US Provisional Patent Application 64/075,643, Stanford University).

References

[1] Menachem Stern, Daniel Hexner, Jason W. Rocks, and Andrea J. Liu. Supervised learning in physical networks: From machine learning to learning machines. *Phys. Rev. X*, 11:021045, May 2021. doi: 10.1103/PhysRevX.11.021045. URL `https://link.aps.org/doi/10.1103/PhysRevX.11.021045`.

[2] Sam Dillavou, Menachem Stern, Andrea J. Liu, and Douglas J. Durian. Demonstration of decentralized physics-driven learning. *Phys. Rev. Appl.*, 18:014040, Jul 2022. doi: 10.1103/PhysRevApplied.18.014040. URL https://link.aps.org/doi/10.1103/PhysRevApplied.18.014040.

[3] Logan G. Wright, Tatsuhiro Onodera, Martin M. Stein, Tianyu Wang, Darren T. Schachter, Zoey Hu, and Peter L. McMahon. Deep physical neural networks trained with backpropagation. *Nature*, 601(7894): 549–555, January 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04223-6. URL http://dx.doi.org/10.1038/s41586-021-04223-6.

[4] Ali Momeni, Babak Rahmani, Benjamin Scellier, Logan G. Wright, Peter L. McMahon, Clara C. Wanjura, Yuhang Li, Anas Skalli, Natalia G. Berloff, Tatsuhiro Onodera, Ilker Oguz, Francesco Morichetti, Philipp del Hougne, Manuel Le Gallo, Abu Sebastian, Azalia Mirhoseini, Cheng Zhang, Danijela Marković, Daniel Brunner, Christophe Moser, Sylvain Gigan, Florian Marquardt, Aydogan Ozcan, Julie Grollier, Andrea J. Liu, Demetri Psaltis, Andrea Alù, and Romain Fleury. Training of physical neural networks. *Nature*, 645(8079):53–61, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09384-2. URL http://dx.doi.org/10.1038/s41586-025-09384-2.

[5] David A. B. Miller. Self-configuring universal linear optical component [invited]. *Photon. Res.*, 1(1): 1–15, Jun 2013. doi: 10.1364/PRJ.1.000001. URL https://opg.optica.org/prj/abstract.cfm?URI=prj-1-1-1.

[6] 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 & Applications*, 6(12):e17110–e17110, June 2017. ISSN 2047-7538. doi: 10.1038/lsa.2017.110. URL http://dx.doi.org/10.1038/lsa.2017.110.

[7] Maziyar Milanizadeh, SeyedMohammad SeyedinNavadeh, Francesco Zanetto, Vittorio Grimaldi, Christian De Vita, Charalambos Klitis, Marc Sorel, Giorgio Ferrari, David A. B. Miller, Andrea Melloni, and Francesco Morichetti. Separating arbitrary free-space beams with an integrated photonic processor. *Light: Science & Applications*, 11(1), July 2022. ISSN 2047-7538. doi: 10.1038/s41377-022-00884-8. URL http://dx.doi.org/10.1038/s41377-022-00884-8.

[8] SeyedMohammad SeyedinNavadeh, Maziyar Milanizadeh, Francesco Zanetto, Giorgio Ferrari, Marco Sampietro, Marc Sorel, David A. B. Miller, Andrea Melloni, and Francesco Morichetti. Determining the optimal communication channels of arbitrary optical systems using integrated photonic processors. *Nature Photonics*, 18(2):149–155, November 2023. ISSN 1749-4893. doi: 10.1038/s41566-023-01330-w. URL http://dx.doi.org/10.1038/s41566-023-01330-w.

[9] David A. B. Miller, Charles Roques-Carmes, Carson G. Valdez, Anne R. Kroo, Marek Vlk, Shanhui Fan, and Olav Solgaard. Universal programmable and self-configuring optical filter. *Optica*, 12(9):1417, August 2025. ISSN 2334-2536. doi: 10.1364/optica.557630. URL http://dx.doi.org/10.1364/OPTICA.557630.

[10] Carson G. Valdez, Anne R. Kroo, Marek Vlk, Charles Roques-Carmes, Shanhui Fan, David A. B. Miller, and Olav Solgaard. Programmable optical filters based on feed-forward photonic meshes. *Light: Science & Applications*, 15(1), August 2026. ISSN 2047-7538. doi: 10.1038/s41377-026-02461-9. URL http://dx.doi.org/10.1038/s41377-026-02461-9.

[11] Charles Roques-Carmes, Shanhui Fan, and David A. B. Miller. Measuring, processing, and generating partially coherent light with self-configuring optics. *Light: Science & Applications*, 13(1), September 2024. ISSN 2047-7538. doi: 10.1038/s41377-024-01622-y. URL http://dx.doi.org/10.1038/s41377-024-01622-y.

[12] Paul-Alexis Mor, Anne R. Kroo, Carson G. Valdez, Marko Šimić, Aviv Karnieli, Gabriele Cavicchioli, Zhanghao Sun, Vittorio Grimaldi, Shanhui Fan, Olav Solgaard, David A. B. Miller, and Charles Roques-Carmes. Automatic on-chip separation of partially coherent light into its mutually incoherent components. *Nature Photonics*, October 2026. ISSN 1749-4893. doi: 10.1038/s41566-026-02035-6. URL http://dx.doi.org/10.1038/s41566-026-02035-6.

[13] Aviv Karnieli, Paul-Alexis Mor, Charles Roques-Carmes, Eran Lustig, Jamison Sloan, Jelena Vučković, David A.B. Miller, and Shanhui Fan. Variational processing of multimode squeezed light. *PRX Quantum*, 7:020311, Apr 2026. doi: 10.1103/mcmq-qf4p. URL https://link.aps.org/doi/10.1103/mcmq-qf4p.

[14] Charles Roques-Carmes, Aviv Karnieli, David A. B. Miller, and Shanhui Fan. Automated modal analysis of entanglement with bipartite self-configuring optics. *ACS Photonics*, 12(6):3285–3294, 2025. doi: 10.1021/acsphotonics.5c00813. URL https://doi.org/10.1021/acsphotonics.5c00813.

[15] Unė G. Būtaitė, Hlib Kupianskyi, Tomáš Čižmár, and David B. Phillips. How to build the "optical inverse" of a multimode fibre. *Intelligent Computing*, 2022, January 2022. ISSN 2771-5892. doi: 10.34133/2022/9816026. URL http://dx.doi.org/10.34133/2022/9816026.

[16] Hlib Kupianskyi, Simon A. R. Horsley, and David B. Phillips. All-optically untangling light propagation through multimode fibers. *Optica*, 11(1):101, January 2024. ISSN 2334-2536. doi: 10.1364/optica.502144. URL http://dx.doi.org/10.1364/OPTICA.502144.

[17] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, Jul 1994. doi: 10.1103/PhysRevLett.73.58. URL https://link.aps.org/doi/10.1103/PhysRevLett.73.58.

[18] Daniel Pérez, Ivana Gasulla, Prometheus Das Mahapatra, and José Capmany. Principles, fundamentals, and applications of programmable integrated photonics. *Advances in Optics and Photonics*, 12(3):709, September 2020. ISSN 1943-8206. doi: 10.1364/aop.387155. URL http://dx.doi.org/10.1364/AOP.387155.

[19] Gerardo Adesso, Sammy Ragy, and Antony R. Lee. Continuous variable quantum information: Gaussian states and beyond. *Open Systems & Information Dynamics*, 21(01n02):1440001, March 2014. ISSN 1793-7191. doi: 10.1142/s1230161214400010. URL http://dx.doi.org/10.1142/S1230161214400010.

[20] Alessio Serafini. Quantum continuous variables: A primer of theoretical methods, July 2017. URL http://dx.doi.org/10.1201/9781315118727.

[21] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. Gaussian quantum information. *Rev. Mod. Phys.*, 84:621–669, May 2012. doi: 10.1103/RevModPhys.84.621. URL https://link.aps.org/doi/10.1103/RevModPhys.84.621.

[22] C. Fabre and N. Treps. Modes and states in quantum optics. *Rev. Mod. Phys.*, 92:035005, Sep 2020. doi: 10.1103/RevModPhys.92.035005. URL https://link.aps.org/doi/10.1103/RevModPhys.92.035005.

[23] Radim Filip, Ladislav Mišta, and Petr Marek. Elimination of mode coupling in multimode continuous-variable key distribution. *Phys. Rev. A*, 71:012323, Jan 2005. doi: 10.1103/PhysRevA.71.012323. URL https://link.aps.org/doi/10.1103/PhysRevA.71.012323.

[24] Olena Kovalenko, Vladyslav C. Usenko, and Radim Filip. Cross talk compensation in multimode continuous-variable entanglement distribution. *Optics Express*, 29(15):24083, July 2021. ISSN 1094-4087. doi: 10.1364/oe.428628. URL http://dx.doi.org/10.1364/OE.428628.

[25] Ruo-Jing Ren, Yong-Heng Lu, Ze-Kun Jiang, Jun Gao, Wen-Hao Zhou, Yao Wang, Zhi-Qiang Jiao, Xiao-Wei Wang, Alexander S. Solntsev, and Xian-Min Jin. Topologically protecting squeezed light on a photonic chip. *Photonics Research*, 10(2):456, January 2022. ISSN 2327-9125. doi: 10.1364/prj.445728. URL http://dx.doi.org/10.1364/PRJ.445728.

[26] Yong Siah Teo, Kimin Park, Seongwook Shin, Hyunseok Jeong, and Petr Marek. Highly accurate Gaussian process tomography with geometrical sets of coherent states. *New Journal of Physics*, 23(6):063024, June 2021. ISSN 1367-2630. doi: 10.1088/1367-2630/abf702. URL http://dx.doi.org/10.1088/1367-2630/abf702.

[27] Xiang-Bin Wang, Zong-Wen Yu, Jia-Zhong Hu, Adam Miranowicz, and Franco Nori. Efficient tomography of quantum-optical Gaussian processes probed with a few coherent states. *Phys. Rev. A*, 88:022101, Aug 2013. doi: 10.1103/PhysRevA.88.022101. URL https://link.aps.org/doi/10.1103/PhysRevA.88.022101.

[28] Chandan Kumar, Ritabrata Sengupta, and Arvind. Optimal characterization of Gaussian channels using photon-number-resolving detectors. *Phys. Rev. A*, 102:012616, Jul 2020. doi: 10.1103/PhysRevA.102.012616. URL https://link.aps.org/doi/10.1103/PhysRevA.102.012616.

[29] Logan W. Grove, Pratik J. Barge, and Kevin Valson Jacob. Characterizing Gaussian quantum processes with Gaussian resources. *Phys. Rev. A*, 112:032406, Sep 2025. doi: 10.1103/sxd2-5z5d. URL https://link.aps.org/doi/10.1103/sxd2-5z5d.

[30] Marco Fanizza, Vishnu Iyer, Junseo Lee, Antonio A. Mele, and Francesco A. Mele. Efficient learning of bosonic Gaussian unitaries, 2025. URL https://arxiv.org/abs/2510.05531.

[31] Claude Bloch and Albert Messiah. The canonical form of an antisymmetric tensor and its application to the theory of superconductivity. *Nuclear Physics*, 39:95–106, December 1962. ISSN 0029-5582. doi: 10.1016/0029-5582(62)90377-2. URL http://dx.doi.org/10.1016/0029-5582(62)90377-2.

[32] Gianfranco Cariolaro and Gianfranco Pierobon. Bloch-Messiah reduction of Gaussian unitaries by Takagi factorization. *Phys. Rev. A*, 94:062109, Dec 2016. doi: 10.1103/PhysRevA.94.062109. URL https://link.aps.org/doi/10.1103/PhysRevA.94.062109.

[33] Timjan Kalajdzievski and Nicolás Quesada. Exact and approximate continuous-variable gate decompositions. *Quantum*, 5:394, February 2021. ISSN 2521-327X. doi: 10.22331/q-2021-02-08-394. URL http://dx.doi.org/10.22331/q-2021-02-08-394.

[34] Tyler Volkoff, Zoë Holmes, and Andrew Sornborger. Universal compiling and (no-)free-lunch theorems for continuous-variable quantum learning. *PRX Quantum*, 2:040327, Nov 2021. doi: 10.1103/PRXQuantum.2.040327. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.040327.

[35] Christopher Alexiev, Jason C. C. Mak, Wesley D. Sacher, and Joyce K. S. Poon. Calibrating rectangular interferometer meshes with external photodetectors. *OSA Continuum*, 4(11):2892, November 2021. ISSN 2578-7519. doi: 10.1364/osac.437918. URL http://dx.doi.org/10.1364/OSAC.437918.

[36] Andreas Fyrillas, Olivier Faure, Nicolas Maring, Jean Senellart, and Nadia Belabas. Scalable machine learning-assisted clear-box characterization for optimally controlled photonic circuits. *Optica*, 11(3):427, March 2024. ISSN 2334-2536. doi: 10.1364/optica.512148. URL http://dx.doi.org/10.1364/OPTICA.512148.

[37] Jin-Hao Zheng, Qin-Qin Wang, Lan-Tian Feng, Yu-Yang Ding, Xiao-Ye Xu, Xi-Feng Ren, Chuan-Feng Li, and Guang-Can Guo. Global calibration of large-scale photonic integrated circuits. *Phys. Rev. Appl.*, 22:054011, Nov 2024. doi: 10.1103/PhysRevApplied.22.054011. URL https://link.aps.org/doi/10.1103/PhysRevApplied.22.054011.

[38] David A. B. Miller. Perfect optics with imperfect components. *Optica*, 2(8):747, August 2015. ISSN 2334-2536. doi: 10.1364/optica.2.000747. URL http://dx.doi.org/10.1364/OPTICA.2.000747.

[39] A. I. Lvovsky and M. G. Raymer. Continuous-variable optical quantum-state tomography. *Rev. Mod. Phys.*, 81:299–332, Mar 2009. doi: 10.1103/RevModPhys.81.299. URL https://link.aps.org/doi/10.1103/RevModPhys.81.299.

[40] M. Milanizadeh, T. Jonuzi, P. Borga, F. Toso, G. Ferrari, M. Sampietro, D.A.B. Miller, A. Melloni, and F. Morichetti. Control of programmable photonic integrated meshes for free-space optics applications. In *OSA Advanced Photonics Congress (AP) 2020 (IPR, NP, NOMA, Networks, PVLED, PSC, SPPCom, SOF)*, PSC, page PsM2F.1. Optica Publishing Group, 2020. doi: 10.1364/psc.2020.psm2f.1. URL http://dx.doi.org/10.1364/PSC.2020.PsM2F.1.

[41] Sanja Singer and Saša Singer. Rounding error and perturbation bounds for the symplectic QR factorization. *Linear Algebra and its Applications*, 358(1-3):255–279, January 2003. ISSN 0024-3795. doi: 10.1016/s0024-3795(02)00263-x. URL http://dx.doi.org/10.1016/S0024-3795(02)00263-X.

[42] Sanja Singer. Orthosymmetric block rotations. *The Electronic Journal of Linear Algebra*, 23, January 2012. ISSN 1081-3810. doi: 10.13001/1081-3810.1523. URL http://dx.doi.org/10.13001/1081-3810.1523.

[43] Peter Benner and Heike Faßbender. The symplectic eigenvalue problem, the butterfly form, the SR algorithm, and the Lanczos method. *Linear Algebra and its Applications*, 275-276:19–47, 1998. ISSN 0024-3795. doi: https://doi.org/10.1016/S0024-3795(97)10049-0. URL https://www.sciencedirect.com/science/article/pii/S0024379597100490. Proceedings of the Sixth Conference of the International Linear Algebra Society.

[44] Angelika Bunse-Gerstner. Matrix factorizations for symplectic QR-like methods. *Linear Algebra and its Applications*, 83:49–77, November 1986. ISSN 0024-3795. doi: 10.1016/0024-3795(86)90265-x. URL http://dx.doi.org/10.1016/0024-3795(86)90265-X.

[45] A. Salam, E. Al-Aidarous, and A. El Farouk. Optimal symplectic Householder transformations for SR decomposition. *Linear Algebra and its Applications*, 429(5–6):1334–1353, September 2008. ISSN 0024-3795. doi: 10.1016/j.laa.2008.02.029. URL http://dx.doi.org/10.1016/j.laa.2008.02.029.

[46] Miloud Sadkane and Ahmed Salam. A note on symplectic block reflectors. *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, 33:45–52, 2008. URL http://eudml.org/doc/130628.

[47] Christopher C. Gerry and Peter L. Knight. *Introductory Quantum Optics*. Cambridge University Press, November 2023. ISBN 9781107653948. doi: 10.1017/9781139151207. URL http://dx.doi.org/10.1017/9781139151207.

[48] Ana Cannas da Silva. *Lectures on Symplectic Geometry*. Springer Berlin Heidelberg, 2008. ISBN 9783540453307. doi: 10.1007/978-3-540-45330-7. URL http://dx.doi.org/10.1007/978-3-540-45330-7.

[49] Marc Langer, Raúl Morral-Yepes, Adam Gammon-Smith, Frank Pollmann, and Barbara Kraus. Match-gate circuit representation of fermionic Gaussian states: optimal preparation, approximation, and classical simulation, 2026. URL https://arxiv.org/abs/2603.05675.

[50] Rachel N. Clark, Bethany Puzio, Oliver M. Green, Siva T. Pradyumna, Oliver Trojak, Alberto Politi, and Jonathan C. F. Matthews. Integrated photonics for continuous-variable quantum optics. *Nature Photonics*, 20(5):489–503, April 2026. ISSN 1749-4893. doi: 10.1038/s41566-026-01864-9. URL http://dx.doi.org/10.1038/s41566-026-01864-9.

[51] Henning Vahlbruch, Moritz Mehmet, Karsten Danzmann, and Roman Schnabel. Detection of 15 dB squeezed states of light and their application for the absolute calibration of photoelectric quantum efficiency. *Phys. Rev. Lett.*, 117:110801, Sep 2016. doi: 10.1103/PhysRevLett.117.110801. URL https://link.aps.org/doi/10.1103/PhysRevLett.117.110801.

[52] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, January 2002. ISBN 9780898718027. doi: 10.1137/1.9780898718027. URL http://dx.doi.org/10.1137/1.9780898718027.

[53] Kazuki Hirota, Takahiro Kashiwazaki, Gyeongmin Ha, Taichi Yamashima, Pawaphat Jaturaphagorn, Takumi Suzuki, Kazuma Takahashi, Akito Kawasaki, Asuka Inoue, Warit Asavanant, Mamoru Endo, Takeshi Umeki, and Akira Furusawa. Generation of 10-dB squeezed light from a broadband waveguide optical parametric amplifier with improved phase locking method. *Optics Express*, 34(5):7958, February 2026. ISSN 1094-4087. doi: 10.1364/oe.585323. URL http://dx.doi.org/10.1364/OE.585323.

[54] Tushar Sanjay Karnik, Xinyi Ren, Chun-Ho Lee, Bo-Han Wu, Mihir Chaudhari, Clayton Cheung, James Wang, Shi-Yuan Ma, Mahmoud Jalali Mehrabad, Ian Christen, Reshma Kopparapu, Kiwon Kwon, Yue Yu, Sri Krishna Vadlamani, Kamila Kunes, Quntao Zhuang, Dirk Englund, Zaijun Chen, and Mengjie Yu. 18-dB on-chip vacuum squeezing in an adaptively poled lithium niobate waveguide, 2026. URL https://arxiv.org/abs/2605.27607.

[55] Jun-ichi Yoshikawa, Yoshichika Miwa, Alexander Huck, Ulrik L. Andersen, Peter van Loock, and Akira Furusawa. Demonstration of a quantum nondemolition sum gate. *Phys. Rev. Lett.*, 101:250501, Dec 2008. doi: 10.1103/PhysRevLett.101.250501. URL https://link.aps.org/doi/10.1103/PhysRevLett.101.250501.

[56] Mikkel V. Larsen, Xueshi Guo, Casper R. Breum, Jonas S. Neergaard-Nielsen, and Ulrik L. Andersen. Deterministic multi-mode gates on a scalable photonic quantum computing platform. *Nature Physics*, 17 (9):1018–1023, July 2021. ISSN 1745-2481. doi: 10.1038/s41567-021-01296-y. URL http://dx.doi.org/10.1038/s41567-021-01296-y.

[57] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[58] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99:032331, Mar 2019. doi: 10.1103/PhysRevA.99.032331. URL https://link.aps.org/doi/10.1103/PhysRevA.99.032331.

[59] Jinzhe Jiang, Yaqian Zhao, Xin Zhang, Chen Li, Yunlong Yu, and Hailing Liu. Gradients of unitary optical neural networks using parameter-shift rule, 2025. URL https://arxiv.org/abs/2506.11565.

[60] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7):864, July 2018. ISSN 2334-2536. doi: 10.1364/optica.5.000864. URL http://dx.doi.org/10.1364/OPTICA.5.000864.

[61] Sunil Pai, Zhanghao Sun, Tyler W. Hughes, Taewon Park, Ben Bartlett, Ian A. D. Williamson, Momchil Minkov, Maziyar Milanizadeh, Nathnael Abebe, Francesco Morichetti, Andrea Melloni, Shanhui Fan, Olav Solgaard, and David A. B. Miller. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science*, 380(6643):398–404, April 2023. ISSN 1095-9203. doi: 10.1126/science.ade8450. URL http://dx.doi.org/10.1126/science.ade8450.

[62] Farshid Ashtiani, Mohamad Hossein Idjadi, and Kwangwoong Kim. Integrated photonic neural network with on-chip backpropagation training. *Nature*, 651(8107):927–932, March 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10262-8. URL http://dx.doi.org/10.1038/s41586-026-10262-8.

[63] Saumil Bandyopadhyay, Ryan Hamerly, and Dirk Englund. Hardware error correction for programmable photonics. *Optica*, 8(10):1247, September 2021. ISSN 2334-2536. doi: 10.1364/optica.424052. URL http://dx.doi.org/10.1364/OPTICA.424052.

[64] Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund. Accurate self-configuration of rectangular multiport interferometers. *Phys. Rev. Appl.*, 18:024019, Aug 2022. doi: 10.1103/PhysRevApplied.18.024019. URL https://link.aps.org/doi/10.1103/PhysRevApplied.18.024019.

[65] Ryan Hamerly, Saumil Bandyopadhyay, and Dirk Englund. Stability of self-configuring large multiport interferometers. *Phys. Rev. Appl.*, 18:024018, Aug 2022. doi: 10.1103/PhysRevApplied.18.024018. URL https://link.aps.org/doi/10.1103/PhysRevApplied.18.024018.

[66] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4 edition, 2013.

[67] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460, December 2016. ISSN 2334-2536. doi: 10.1364/optica.3.001460. URL http://dx.doi.org/10.1364/OPTICA.3.001460.

[68] Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Re. Learning fast algorithms for linear transforms using butterfly factorizations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1517–1527. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/dao19a.html.

[69] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1120–1128, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/arjovsky16.html.

[70] Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/d9ff90f4000eacd3a6c9cb27f78994cf-Paper.pdf.

[71] 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 Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1733–1741. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jing17a.html.

[72] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using Householder reflections. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2401–2409. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/mhammedi17a.html.

[73] Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3794–3803. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lezcano-casado19a.html.

[74] Thomas Frerix and Joan Bruna. Approximating orthogonal matrices with effective Givens factorization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1993–2001. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/frerix19a.html.

[75] Uri Shalit and Gal Chechik. Coordinate-descent for learning orthogonal matrices through Givens rotations. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 548–556, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/shalit14.html.

[76] Andi Han, Pratik Jawanpuria, and Bamdev Mishra. Riemannian coordinate descent algorithms on matrix manifolds. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17393–17415. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/han24c.html.

[77] 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(7):441–446, June 2017. ISSN 1749-4893. doi: 10.1038/nphoton.2017.93. URL http://dx.doi.org/10.1038/nphoton.2017.93.

[78] Nicholas C. Harris, Jacques Carolan, Darius Bunandar, Mihika Prabhu, Michael Hochberg, Tom Baehr-Jones, Michael L. Fanto, A. Matthew Smith, Christopher C. Tison, Paul M. Alsing, and Dirk Englund. Linear programmable nanophotonic processors. *Optica*, 5(12):1623, December 2018. ISSN 2334-2536. doi: 10.1364/optica.5.001623. URL http://dx.doi.org/10.1364/OPTICA.5.001623.

[79] Martin Houde, Will McCutcheon, and Nicolás Quesada. Matrix decompositions in quantum optics: Takagi/Autonne, Bloch–Messiah/Euler, Iwasawa, and Williamson. *Canadian Journal of Physics*, 102(10):497–507, October 2024. ISSN 1208-6045. doi: 10.1139/cjp-2024-0070. URL http://dx.doi.org/10.1139/cjp-2024-0070.

[80] Gianfranco Cariolaro and Gianfranco Pierobon. Implementation of multimode Gaussian unitaries using primitive components. *Phys. Rev. A*, 98:032111, Sep 2018. doi: 10.1103/PhysRevA.98.032111. URL https://link.aps.org/doi/10.1103/PhysRevA.98.032111.

[81] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Phys. Rev. Res.*, 1:033063, Oct 2019. doi: 10.1103/PhysRevResearch.1.033063. URL https://link.aps.org/doi/10.1103/PhysRevResearch.1.033063.

[82] Vladyslav C. Usenko, Antonio Acín, Romain Alléaume, Ulrik L. Andersen, Eleni Diamanti, Tobias Gehring, Adnan A. E. Hajomer, Florian Kanitschar, Christoph Pacher, Stefano Pirandola, and Valerio Pruneri. Continuous-variable quantum communication. *Rev. Mod. Phys.*, 98:015003, Mar 2026. doi: 10.1103/mgj7-t6d3. URL https://link.aps.org/doi/10.1103/mgj7-t6d3.

[83] Sunil Pai, Ben Bartlett, Olav Solgaard, and David A. B. Miller. Matrix optimization on universal unitary photonic devices. *Phys. Rev. Appl.*, 11:064044, Jun 2019. doi: 10.1103/PhysRevApplied.11.064044. URL https://link.aps.org/doi/10.1103/PhysRevApplied.11.064044.

[84] Gianfranco Cariolaro and Roberto Corvaja. Implementation of two-mode Gaussian states whose covariance matrix has the standard form. *Symmetry*, 14(7):1485, July 2022. ISSN 2073-8994. doi: 10.3390/sym14071485. URL http://dx.doi.org/10.3390/sym14071485.

[85] Arvind, B Dutta, N Mukunda, and R Simon. The real symplectic groups in quantum mechanics and optics. *Pramana*, 45(6):471–497, December 1995. ISSN 0973-7111. doi: 10.1007/bf02848172. URL http://dx.doi.org/10.1007/BF02848172.

Appendix contents

A Explicit gate parameterizations, and the N-mode mesh **17**

B Proof of the symplectic Reck factorization **18**

C Universal architecture **20**

D Conditioning and minimum squeezing **20**

E Training cost **22**

F Effect of loss **23**

G Experimental details **23**

A Explicit gate parameterizations, and the N-mode mesh

A.1 Explicit gate parameterizations

We use the mode-grouped quadrature ordering of Section 2, so a two-mode gate on modes a and b acts on $(x_a, p_a, x_b, p_b)^\top$. Every gate has independently adjustable parameters. We write $S^{(a,b)}$ for a generic two-mode gate on modes a and b, and $S^{(m)}$ for a generic single-mode gate on mode m.

Following Ref. [83], an arbitrary two-mode unitary can be written as

$$U(\theta,\phi,\alpha,\beta) = D(\phi,0)BD(\theta,0)BD(\alpha,\beta) = ie^{i\theta/2}\begin{pmatrix} e^{i(\phi+\alpha)}\sin\frac{\theta}{2} & e^{i(\phi+\beta)}\cos\frac{\theta}{2} \\ e^{i\alpha}\cos\frac{\theta}{2} & -e^{i\beta}\sin\frac{\theta}{2}\end{pmatrix}, \quad (5)$$

where $D(\phi,\psi) = \mathrm{diag}(e^{i\phi}, e^{i\psi})$ and $B = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & i \\ i & 1\end{smallmatrix}\right)$. A unitary $U = (u_{mn})$ corresponds in the real quadrature basis to the orthosymplectic matrix

$$O(U) = \begin{pmatrix} \mathrm{Re}\,u_{11} & -\mathrm{Im}\,u_{11} & \mathrm{Re}\,u_{12} & -\mathrm{Im}\,u_{12} \\ \mathrm{Im}\,u_{11} & \mathrm{Re}\,u_{11} & \mathrm{Im}\,u_{12} & \mathrm{Re}\,u_{12} \\ \mathrm{Re}\,u_{21} & -\mathrm{Im}\,u_{21} & \mathrm{Re}\,u_{22} & -\mathrm{Im}\,u_{22} \\ \mathrm{Im}\,u_{21} & \mathrm{Re}\,u_{21} & \mathrm{Im}\,u_{22} & \mathrm{Re}\,u_{22}\end{pmatrix}. \quad (6)$$

A general lossless two-mode symplectic gate admits the Bloch–Messiah form

$$S^{(a,b)} = O_1^{(a,b)}\,\mathrm{diag}(e^{r_1}, e^{-r_1}, e^{r_2}, e^{-r_2})O_2^{(a,b)}, \quad (7)$$

where the passive factors are implemented with beam splitters and phase shifters [22, 31, 84]. The two passive factors and two squeezers give ten real parameters, matching $\dim \mathrm{Sp}(4,\mathbb{R}) = 10$. Similarly,

$$S^{(m)} = \mathcal{R}(\varphi_1)\begin{pmatrix} e^r & 0 \\ 0 & e^{-r}\end{pmatrix}\mathcal{R}(\varphi_2), \qquad \mathcal{R}(\varphi) = \begin{pmatrix}\cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi\end{pmatrix}, \quad (8)$$

which gives the three parameters of $\mathrm{Sp}(2,\mathbb{R})$. Figure 4 illustrates these Bloch–Messiah implementations.

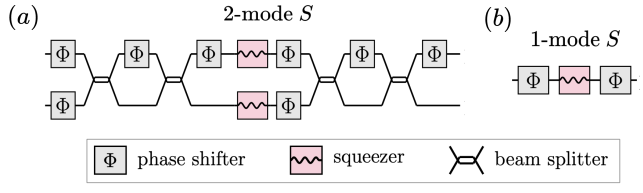


Figure 4: Bloch–Messiah implementations of (a) a two-mode symplectic gate and (b) a single-mode symplectic gate, using phase shifters, squeezers, and beam splitters.

A.2 Symplectic Reck mesh for N modes

Using these single-mode and two-mode gates, we now write the full mesh explicitly as a product of its layers. Layer L_k, for $k = 1, \ldots, N-1$, contains the $N-k$ two-mode gates that null column k (Section 4), each a gate $S^{(a,a+1)}$ on adjacent modes, and the final layer L_N applies one single-mode gate $S^{(m)}$ to each mode m, with both gate types defined in Appendix A.1. With the rightmost factor acting first,

$$L_k = S_{L_k}^{(k,k+1)} S_{L_k}^{(k+1,k+2)} \cdots S_{L_k}^{(N-1,N)}, \qquad L_N = \bigoplus_{m=1}^{N} S_{L_N}^{(m)}. \tag{9}$$

The full mesh is then

$$S_\theta = L_N L_{N-1} \cdots L_1. \tag{10}$$

B Proof of the symplectic Reck factorization

This appendix proves the local nulling criterion (Section 5.1) and Theorem 1. We first establish when one step succeeds, then show that the full sequence completes the inversion for generic targets.

B.1 One nulling step

With P, E and Δ as in Section 5.1, let $V = \left(\begin{smallmatrix} P \\ E \end{smallmatrix}\right)$. We seek a symplectic gate G with $GV = \left(\begin{smallmatrix} Q \\ 0 \end{smallmatrix}\right)$. For any real 2×2 matrix X,

$$X^\top J_2 X = (\det X)\, J_2. \tag{11}$$

Applying this to P and E gives

$$V^\top J_4 V = P^\top J_2 P + E^\top J_2 E = \Delta J_2, \tag{12}$$

which is preserved by any G with $G^\top J_4 G = J_4$.

If $V = 0$, the target block is already zero. If V has rank one, both columns are multiples of the same real four-vector. Pairing its x and p components gives one complex two-mode amplitude, which a beam splitter with phase shifters moves into the upper mode.

Now suppose V has rank two. If $GV = \left(\begin{smallmatrix} Q \\ 0 \end{smallmatrix}\right)$, then $Q^\top J_2 Q = \Delta J_2$, so $\det Q = \Delta$. Since G is invertible, Q has rank two, so nulling requires $\Delta \neq 0$.

Conversely, suppose $\Delta \neq 0$ and that P is invertible. The gate $\left(\begin{smallmatrix} I_2 & 0 \\ C & I_2 \end{smallmatrix}\right)$ with $C = -EP^{-1}$ nulls E but is not in general symplectic. We therefore add a free block B,

$$G_0 = \begin{pmatrix} I_2 & B \\ C & I_2 \end{pmatrix}, \qquad G_0 V = \begin{pmatrix} P + BE \\ 0 \end{pmatrix}, \tag{13}$$

which still nulls E for any B, and choose B to make G_0 symplectic. We have

$$G_0^\top J_4 G_0 = \begin{pmatrix} J_2 + C^\top J_2 C & J_2 B + C^\top J_2 \\ B^\top J_2 + J_2 C & B^\top J_2 B + J_2 \end{pmatrix}. \tag{14}$$

Choosing $B = J_2 C^\top J_2$ makes the off-diagonal blocks vanish. Using Eq. (11) and $\det B = \det C$ then gives

$$G_0^\top J_4 G_0 = \lambda J_4, \qquad \lambda = 1 + \det C = 1 + \frac{\det E}{\det P} = \frac{\Delta}{\det P} \neq 0. \tag{15}$$

To remove this scalar factor, set $L = \mathrm{diag}(1, \lambda^{-1})$, so $L^\top J_2 L = \lambda^{-1} J_2$, and define

$$G = (L \oplus L) G_0. \tag{16}$$

Then $G^\top J_4 G = J_4$ and $GV = \left(\begin{smallmatrix} L(P+BE) \\ 0 \end{smallmatrix}\right)$, so G is symplectic and nulls E. If P is singular, then E is invertible since $\det E = \Delta \neq 0$. A passive swap of the two modes, absorbed into the same gate, reduces this case to the one above.

Hence, for rank-two V, a symplectic nulling gate exists if and only if $\Delta \neq 0$, and for rank at most one it always exists. Similar criteria were derived in Refs. [42, 49] using different explicit gates, assuming invertible P or invertible P and E, respectively.

B.2 Preservation of zeros and block diagonality

Having shown that each step can null its target block, we now show that later steps do not undo earlier ones, and that the result is block diagonal. At nulling step (k, j) the gate mixes only block rows $j-1$ and j. In column k, the blocks below row j are already zero and are not touched again. In every earlier column, the blocks in rows $j-1$ and j are already zero, so mixing these rows preserves both zeros. After all $N(N-1)/2$ two-mode steps, the response is block upper triangular. A block upper-triangular symplectic matrix is block diagonal. Write $T = \left(\begin{smallmatrix} T_1 & R \\ 0 & T_{\mathrm{rem}} \end{smallmatrix}\right)$. The top-left and top-right parts of $T^\top J_{2N} T = J_{2N}$ give $T_1^\top J_2 T_1 = J_2$ and $T_1^\top J_2 R = 0$. The first makes T_1 invertible, so the second gives $R = 0$. Repeating this for T_{rem} gives $T = T_1 \oplus \cdots \oplus T_N$ with each $T_m \in \mathrm{Sp}(2, \mathbb{R})$, and the final layer applies T_m^{-1} on each mode, giving $S_\theta S_{\mathrm{unk}} = I$.

B.3 The choice of nulling gate does not change later pivots

The gate that completes a nulling step is not unique. We now show that the choice of nulling gate does not affect whether later steps succeed, a result we use in the genericity proof below. Suppose G and $\widetilde{G}$ both null the same rank-two response V, giving invertible upper blocks Q and $\widetilde{Q}$. Then $H = G\widetilde{G}^{-1}$ satisfies

$$H \begin{pmatrix} \widetilde{Q} \\ 0 \end{pmatrix} = \begin{pmatrix} Q \\ 0 \end{pmatrix}. \tag{17}$$

Since $\widetilde{Q}$ is invertible, H maps the upper mode into itself, so H is block upper triangular. By Appendix B.2, $H = A \oplus D$ with $A, D \in \mathrm{Sp}(2, \mathbb{R})$, so two valid nulling gates differ only by single-mode gates on their outputs. Thus, changing the gate replaces each block X it acts on with AX or DX. Since $\det A = \det D = 1$, every block determinant (and hence every later pivot) is unchanged. This means that whether the full sequence succeeds depends only on the entries of S_{unk}. The choice of gate does still affect the condition number κ of later steps, because single-mode squeezing changes the Euclidean area spanned by the response columns.

B.4 Generic success: proof of Theorem 1

The previous subsections show that the mesh inverts a target whenever every nulling step has $\Delta_t \neq 0$. We now show that this holds for almost every target. We do this by showing that every failing target lies on the zero set of a single nonzero polynomial. By Appendix B.3, we may use the gate of Eq. (16) at every step. We call a target *exceptional* if $\det P_t = 0$ or $\Delta_t = 0$ at some step t. This set contains every failure. It also contains some targets that can still be nulled (those with $\det P_t = 0$ but $\Delta_t \neq 0$). This does not matter since including these only enlarges the exceptional set. Showing that this larger set has measure zero still shows that the failures have measure zero. Outside this set, each gate is built from the current response using only additions, multiplications, determinants and P_t^{-1}. Every later entry, and every $\det P_t$ and Δ_t, is therefore a ratio of polynomials in the entries of S_{unk}. Let Π be the product of all these numerators and denominators. Every exceptional target satisfies $\Pi(S_{\mathrm{unk}}) = 0$.

It remains to show that Π is not identically zero. One target with $\Pi \neq 0$ is enough to show this. To build such an example, let us start from a diagonal unitary and apply 50:50 beam splitters in the reverse order of the nulling steps. The nulling sequence then removes these beam splitters one at a time. Let p_t and e_t be the complex entries corresponding to P_t and E_t. Each 50:50 beam splitter maps a nonzero entry to two nonzero entries of equal magnitude, so $p_t \neq 0$ at every step. For a passive matrix, a complex entry z becomes a 2×2 block with determinant $|z|^2$, so

$$\det P_t = |p_t|^2 > 0, \qquad \Delta_t = |p_t|^2 + |e_t|^2 > 0. \tag{18}$$

By Appendix B.3, the gate of Eq. (16) encounters the same block determinants, so $\Pi \neq 0$ for this target. The zero set of Π, which contains every exceptional target, therefore has measure zero in $\mathrm{Sp}(2N, \mathbb{R})$.

Finally, the same result gives a factorization theorem. Applying the inversion procedure to S^{-1} gives a mesh with $S_\theta S^{-1} = I$, so $S_\theta = S$. Since inversion is a diffeomorphism of $\mathrm{Sp}(2N, \mathbb{R})$, it preserves genericity, so generic symplectic matrices admit a symplectic Reck factorization.

C Universal architecture

Theorem 1 shows that the symplectic Reck mesh inverts generic targets. Similar to Ref. [41], we add passive gates before each two-mode symplectic nulling step, which avoids zero pivots. Here we give one example of a modified version of the symplectic Reck mesh that inverts every $S \in \mathrm{Sp}(2N, \mathbb{R})$. Before nulling $M_{j,k}$ with $j > k+1$, we add extra passive two-mode gates on modes $(k, k+1), (k+1, k+2), \ldots, (j-2, j-1)$, applied in that order, and then the usual symplectic gate acts on modes $(j-1, j)$. Each extra gate is a tunable beam splitter with a phase shifter, a special case of the two-mode unitary of Eq. (5). The gate on modes $(a, a+1)$ is set so that mode a no longer responds to the x_k probe, moving that response into mode $a+1$, exactly as a gate of the unitary Reck mesh moves light from one waveguide into the next. No extra gates are needed when $j = k+1$. As before, single-mode symplectic gates complete the inversion.

To see why this works, let M be the current response before nulling $M_{j,k}$. The modes before k are already decoupled (Appendix B.2), and the blocks below row j have been nulled, so only rows $k, \ldots, j$ of column k are nonzero. Since M is symplectic, the diagonal block of $M^\top J_{2N} M = J_{2N}$ for column k, together with Eq. (11), gives

$$\sum_{i=k}^{j} \det M_{i,k} = 1. \tag{19}$$

Intuitively, this equation says that input mode k carries one unit of symplectic area (Section 5.3), spread over output modes $k, \ldots, j$. The extra gates collect this area into modes $j-1$ and j. After them, the blocks $M_{i,k}$ with $k \le i \le j-2$ have a zero first column and hence zero determinant. The extra gates are symplectic and leave mode j unchanged, so

$$\Delta = \det M_{j-1,k} + \det M_{j,k} = 1, \tag{20}$$

and the gate on modes $(j-1, j)$ can null $M_{j,k}$ (Section 5.1). For $j = k+1$, the same identity gives $\Delta = 1$ without extra gates. The extra gates act only on modes $k, \ldots, j-1$, whose blocks in earlier columns are already zero, so they preserve every earlier zero. Appendix B.2 then completes the inversion.

In hardware, each extra gate can be set by feedback, minimizing the measured response of its first mode to the x_k probe, or computed in closed form from that response. In noiseless simulations with the minimum-squeezing gate, the extra gates also reduce the squeezing required by the two-mode gates, $\frac{1}{2} \operatorname{arccosh} \kappa_{\max}$. For 30 random targets with $N = 20$, the median $\kappa_{\max}$ drops from 7.4 to 1.04, corresponding to 11.7 dB and 1.3 dB of squeezing. As the target approaches a nulling failure in Fig. 2, the required squeezing remains near 5 dB with the extra gates, whereas without them it grows to tens of dB. Other constructions of universal architectures are also possible.

D Conditioning and minimum squeezing

The nulling criterion establishes when a gate exists. We now ask how much squeezing it requires and how strongly it amplifies measurement errors. We construct a minimum-squeezing gate, prove its optimality, count its parameters, and derive the nulling-step error bound of Section 5.3. We assume V has rank two and $\Delta \neq 0$, with κ as in Eq. (4).

D.1 The minimum-squeezing gate

We build the gate from a two-mode passive gate R followed by the two-mode squeezer

$$T_{\mathrm{sq}}(r) = \begin{pmatrix} \cosh r\, I_2 & \sinh r\, Z \\ \sinh r\, Z & \cosh r\, I_2 \end{pmatrix}, \qquad Z = \mathrm{diag}(1, -1). \tag{21}$$

This squeezer can itself be built from two single-mode squeezers between two passive interferometers [21]. For $T_{\mathrm{sq}}(r)$ to output $\binom{Q}{0}$, its input must be

$$W = T_{\mathrm{sq}}(r)^{-1} \begin{pmatrix} Q \\ 0 \end{pmatrix} = \begin{pmatrix} \cosh r\, Q \\ -\sinh r\, ZQ \end{pmatrix}, \tag{22}$$

so the passive gate must map V to W.

We first choose r and Q. A passive gate preserves both $V^\top V$ and $V^\top J_4 V = \Delta J_2$, so W must have the same two products. Using $Z^\top J_2 Z = -J_2$,

$$W^\top W = \cosh(2r)\, Q^\top Q, \qquad W^\top J_4 W = (\det Q)\, J_2. \tag{23}$$

We therefore need $Q^\top Q = V^\top V/\cosh(2r)$ and $\det Q = \Delta$. Taking determinants gives $\cosh(2r) = \kappa$, so

$$r = \tfrac{1}{2} \operatorname{arccosh} \kappa. \tag{24}$$

Let $Q_0 = (V^\top V/\kappa)^{1/2}$ be the symmetric positive square root, so $Q_0^\top Q_0 = V^\top V/\kappa$ and $\det Q_0 = |\Delta|$. We take $Q = Q_0$ if $\Delta > 0$ and $Q = ZQ_0$ if $\Delta < 0$, which leaves $Q^\top Q$ unchanged and gives $\det Q = \Delta$.

This requires $\kappa \geq 1$, which always holds. To see this, define the complex 2×2 matrix A_V with entries $(A_V)_{mc} = V_{x_m,c} + iV_{p_m,c}$, so that each row gives the complex amplitude $x_m + ip_m$ of output mode m in response to input quadrature c. A direct calculation gives

$$A_V^\dagger A_V = V^\top V + i\, V^\top J_4 V = V^\top V + i\Delta J_2. \tag{25}$$

Since $A_V^\dagger A_V$ is positive semidefinite, its determinant $\det(V^\top V) - \Delta^2$ is nonnegative, so $\kappa \geq 1$.

With r and Q fixed, it remains to find a passive gate R that maps V to W. Define A_W from W in the same way. Since W has the same two products as V, Eq. (25) gives $A_W^\dagger A_W = A_V^\dagger A_V$. If $\kappa > 1$, then $|\det A_V|^2 = \det(V^\top V) - \Delta^2 > 0$, so A_V is invertible and $U = A_W A_V^{-1} \in \mathrm{U}(2)$ satisfies $UA_V = A_W$. We take $R = O(U)$, the two-mode passive gate that applies U to the complex amplitudes $x_m + ip_m$ of the two modes (Eq. (6)). Any element of $\mathrm{U}(2)$ can be realized with a beam splitter and phase shifters, as in Eq. (5) (Appendix A). Then R maps A_V to $UA_V = A_W$, so $RV = W$. If $\kappa = 1$, then $r = 0$ and the columns of A_V are proportional, so a beam splitter with phase shifters moves both into the upper mode. In either case, $G = T_{\mathrm{sq}}(r)R$ nulls E.

D.2 No nulling gate uses less squeezing

The construction above achieves squeezing r. To show that this is minimal, we compare it with an arbitrary nulling gate. By Appendix B.3, any other nulling gate has the form $\widetilde{G} = (L_1 \oplus L_2) T_{\mathrm{sq}}(r) R$ with $L_1, L_2 \in \mathrm{Sp}(2, \mathbb{R})$. Since R is orthogonal,

$$\|\widetilde{G}\|_F^2 = \cosh(2r)\big(\|L_1\|_F^2 + \|L_2\|_F^2\big) \geq 4\cosh(2r), \tag{26}$$

because each L_i has singular values σ, σ^{-1}, so $\|L_i\|_F^2 = \sigma^2 + \sigma^{-2} \geq 2$. If $\widetilde{G}$ has Bloch–Messiah squeezing parameters $s_1, s_2 \geq 0$ (Eq. (7)) and $s = \max(s_1, s_2)$, then

$$\|\widetilde{G}\|_F^2 = 2\cosh 2s_1 + 2\cosh 2s_2 \leq 4\cosh 2s. \tag{27}$$

Together these give $s \geq r$. Our gate squeezes by exactly r, so it minimizes the largest squeezing parameter.

D.3 Four parameters per gate

We next count the independent controls needed to implement this gate. Naively, each minimum-squeezing gate has five parameters, one squeezing parameter r and four in the passive part R, but one is redundant. Let $D_\phi = \mathcal{R}(\phi) \oplus \mathcal{R}(-\phi)$. Since $\mathcal{R}(\phi)Z = Z\mathcal{R}(-\phi)$, D_ϕ commutes with $T_{\mathrm{sq}}(r)$, so

$$T_{\mathrm{sq}}(r) D_\phi R V = D_\phi \begin{pmatrix} Q \\ 0 \end{pmatrix} = \begin{pmatrix} \mathcal{R}(\phi) Q \\ 0 \end{pmatrix}. \tag{28}$$

Replacing R by $D_\phi R$ therefore still nulls E, so this phase can be fixed. With four parameters per two-mode gate and three per single-mode gate,

$$4 \cdot \frac{N(N-1)}{2} + 3N = N(2N+1) = \dim \mathrm{Sp}(2N, \mathbb{R}), \tag{29}$$

matching the dimension of the symplectic group [85].

D.4 Nulling-step error amplification

The same gate construction also gives a direct bound on the residual left by measurement errors. Suppose the gate $\widehat{G}$ is computed from a noisy estimate $\widehat{V} = V + \delta V$ with nonzero estimated pivot, and let $\widehat{N}$ be the bottom two rows of $\widehat{G}$. By construction $\widehat{N}\widehat{V} = 0$, so on the true response

$$\widehat{N}V = \widehat{N}(\widehat{V} - \delta V) = -\widehat{N}\,\delta V. \tag{30}$$

The bottom two rows of $T_{\mathrm{sq}}(\widehat{r})$ are orthogonal with squared norm $\cosh^2\widehat{r} + \sinh^2\widehat{r} = \cosh 2\widehat{r} = \widehat{\kappa}$, and right multiplication by the orthogonal $\widehat{R}$ preserves their inner products. Hence $\widehat{N}\widehat{N}^\top = \widehat{\kappa} I_2$, and

$$\|\widehat{N}V\|_F \le \sqrt{\widehat{\kappa}}\,\|\delta V\|_F, \tag{31}$$

with equality in the worst case, since both singular values of $\widehat{N}$ equal $\sqrt{\widehat{\kappa}}$.

E Training cost

We now count the measurements and computation required to train the full mesh. In our experiments, we choose a fixed measurement budget for each nulling step, with q shots per homodyne setting and, for feedback training, n_{eval} loss evaluations per gate (Appendix G.3). Another option could have been to train each gate until its measured loss falls below a chosen threshold, spending fewer evaluations on easy steps. The mesh has $N(N-1)/2$ two-mode gates and N single-mode gates, each with a constant number of parameters, so it has $O(N^2)$ parameters. Closed-form training uses 16 homodyne settings per two-mode gate and 8 per single-mode gate, or $16 \cdot N(N-1)/2 + 8N = 8N^2$ settings in total, which is $8qN^2$ shots. Feedback training uses 8 settings per loss evaluation, so it uses $O(n_{\mathrm{eval}} qN^2)$ shots. Each closed-form update is a calculation on two 2×2 blocks, so the classical computation is $O(N^2)$, and the training time also scales as $O(N^2)$. For comparison, tomography followed by offline compilation also needs $O(qN^2)$ shots when blocks are measured one at a time, which is the matched budget of Fig. 3. If all outputs are measured in parallel, tomography recovers a whole block column per probe, reducing this to $O(qN)$, whereas in-situ training remains $O(qN^2)$ because each update depends on the previous one. Tomography then needs $O(N^3)$ classical computation to invert and decompose the estimate. In practice, training time is likely to be limited by measurement and hardware reconfiguration rather than by classical computation.

E.1 Conditional sample-complexity guarantee

The cost above is expressed in terms of the shots per setting, q. We now bound how large q must be for closed-form training to reach a residual $\mathcal{E} \le \epsilon$ with probability at least $1 - \delta$, by combining the nulling-step bound of Section 5.3 over the full nulling sequence. We assume exactly implemented gates, so the only errors come from finite-shot measurements, together with a set of stability conditions stated below. Let $M = S_\theta S_{\mathrm{unk}}$ be the final response and write $M - I = L + U + D$, where L, U and D contain the blocks below, above and on the block diagonal. Let R_t be the residual block left immediately after nulling step t, and let H_k be the combined action of all later gates on modes $k+1, \dots, N$.

We first show how these residuals accumulate. Once column k is processed, its lower part is a stack of that column's residual blocks, which later gates multiply by H_k. If $\|H_k\|_2 \le \Gamma$ for every k, then $\|L\|_F^2 \le \Gamma^2 \sum_t \|R_t\|_F^2$. Symplecticity gives $M^{-1} - I = -J(M-I)^\top J$, so the upper part of $M^{-1} - I$ has norm $\|L\|_F$ and $\|M^{-1} - I\|_F = \|M - I\|_F$. Taking the upper part of the identity $(M-I) + (M^{-1} - I) = -(M-I)(M^{-1} - I)$ gives $\|U\|_F \le \|L\|_F + \rho\|M - I\|_F$, where $\rho = \|M - I\|_2$. For $\rho < 1$,

$$\mathcal{E}^2 \le \frac{4\Gamma^2 \sum_t \|R_t\|_F^2 + \|D\|_F^2}{N(1-\rho)^2}. \tag{32}$$

To turn this accumulation bound into a shot requirement, we impose the following stability conditions. Suppose that, with probability at least $1 - \delta/2$, every measured condition number satisfies $\widehat{\kappa}_t \le K$, every $\|H_k\|_2 \le \Gamma$ and $\|M - I\|_2 \le \rho_0 < 1$, where these constants are fixed in advance. These conditions are most natural for few-mode meshes, since the worst condition number and the

amplification by later gates tend to grow with N for random targets (Fig. 2), and $\|M-I\|_2 \le \rho_0 < 1$ restricts the guarantee to the regime near convergence. Suppose also that, on these runs, the mean of each $\|R_t\|_F^2$ and of $\|D\|_F^2/(2N)$ is at most c/q. Summing Eq. (32) over the $N(N-1)/2$ nulling steps and applying Markov's inequality gives the sufficient condition

$$q \ge \frac{4c\,[\Gamma^2(N-1)+1]}{(1-\rho_0)^2\epsilon^2\delta} \quad \Longrightarrow \quad \Pr(\mathcal{E} \le \epsilon) \ge 1-\delta. \tag{33}$$

To relate c to the measurement noise, we use the nulling-step bound of Eq. (31), $\|R_t\|_F^2 \le \widehat{\kappa}_t \|\delta V_t\|_F^2$. If each measured quadrature has variance at most $a/2$, each of the eight entries of $\widehat{V}_t$ has variance at most $a/(4\eta^2 q)$, so one may take $c = \max\{4aK/\eta^2, c_{\mathrm{diag}}\}$, where c_{diag}/q bounds the diagonal error. With $8N^2$ settings, the total number of shots is $O\big(c\,(\Gamma^2 N^3 + N^2)/[(1-\rho_0)^2\epsilon^2\delta]\big)$. When c, Γ and $(1-\rho_0)^{-1}$ are bounded independently of N, this becomes

$$O\left(\frac{N^3}{\epsilon^2\delta}\right), \tag{34}$$

and it remains polynomial in N whenever they grow at most polynomially. This gives a conditional PAC guarantee for the full nulling sequence.

F Effect of loss

Theorem 1 assumes lossless operations. With loss, the response is no longer symplectic and cannot in general be inverted, so we instead test how well the trained mesh separates the modes of the first-moment response. We measure the remaining mode mixing by defining $C_{\mathrm{off}} = \|M - \mathrm{blockdiag}(M)\|_F / \|M\|_F$, where M is the response after training with Algorithm 1 and $\mathrm{blockdiag}(M)$ keeps only its 2×2 diagonal blocks. This quantity vanishes when the modes are decoupled, regardless of attenuation.

We use 30 targets with $N = 10$ from the distribution described in Section 6, with exact measurements, and compute each nulling gate from the current lossy response. We test two loss models. To model mode-independent loss, we attenuate every mode by the same fraction of its power at the output. Mode separation then holds to numerical precision for every power loss we simulate (where we simulate up to 40% loss). To model mode-dependent loss, we attenuate both output modes of every two-mode gate by the same fraction of their power. Because different modes pass through different numbers of gates, they accumulate unequal loss, and some mixing remains. The mean C_{off} is about 2.0% at 1% power loss per gate and 10.6% at 5% power loss per gate. Thus, the residual mixing comes from unequal loss across modes, not from the total amount of loss. A potential way to tackle these issues from mode-dependent loss is to add dummy components so that every mode experiences the same loss, similar to methods in Ref. [38]. We find in simulation that this can reduce the residual mode mixing to zero, thereby allowing full mode separation. However, the trade-off is lower overall transmission, which falls more steeply as the loss per gate grows. Finally, even perfect separation of the first moments cannot recover quantum information lost to the environment or remove the noise it adds [20, 21].

G Experimental details

We give the target distributions, noise models and training settings used in the numerical experiments below.

Targets and statistics. Unless stated otherwise, each data point uses 30 independent targets from the distribution of Section 6, with the squeezing parameters sorted in decreasing order. Tables and error bars report medians and 25th–75th percentiles across trials. For Table 1, these are taken over the pooled local losses of all gates rather than over trials.

G.1 Four-mode example (Fig. 1b)

Fig. 1(b) uses $N = 4$, and $r_i \sim \mathrm{Uniform}[0.3, 0.4]$. Gates are trained sequentially with automatic-differentiation gradients and noiseless responses.

G.2 Conditioning experiment (Fig. 2)

For Fig. 2, we perturb each block used by the closed-form gate as

$$\widehat{P} = P + \epsilon Z_P, \qquad \widehat{E} = E + \epsilon Z_E, \qquad \epsilon = 10^{-8}, \tag{35}$$

where Z_P and Z_E are independent Gaussian matrices normalized to unit Frobenius norm. The gate is computed from the perturbed blocks and applied to the true response, and $\kappa_{\max}$ is computed from the true blocks before each perturbation. The final single-mode gates use exact diagonal blocks, isolating errors from the two-mode steps.

The mode-number sweep uses $N \in \{2, 5, 8, 11, 14, 17, 20, 25, 30\}$ with 60 trials per point. The squeezing sweep uses $N = 4$ and $r_i \sim \mathrm{Uniform}[0, r_{\max}]$, with $s_{\max} = 20 r_{\max}/\ln 10 \in \{0, 2, 4, 6, 8, 10\}$ dB.

For the targets approaching a nulling failure, we use $N = 3$ and

$$S(\alpha) = e^{\alpha L} S_{\mathrm{iso}}, \qquad S_{\mathrm{iso}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & -1 & 0 \end{pmatrix}. \tag{36}$$

At $\alpha = 0$, the first nulling step has rank two and $\Delta = 0$. To build the generator L, we draw S_{rand} from the default distribution and take L_0 to be the real part of the matrix logarithm of $S_{\mathrm{rand}} S_{\mathrm{iso}}^{-1}$. We then project it as $L = \frac{1}{2}(L_0 + J_6 L_0^\top J_6)$, which ensures $L^\top J_6 + J_6 L = 0$, so $e^{\alpha L}$, and hence the whole path, is symplectic. The matrix logarithm can be complex, so after taking the real part, $S(1)$ need not equal S_{rand}. This does not affect the sweep, since S_{rand} only sets a random direction away from S_{iso}. We use $\alpha \in \{10^{-8}, 10^{-7}, \ldots, 10^{-2}\}$, holding each trial's generator fixed across the sweep.

G.3 Finite-shot training (Tables 1 and 2, Fig. 5)

For a probe displacement u and current response $M = S_\theta S_{\mathrm{unk}}$, each measured output mean is simulated as

$$\widehat{\mu}_\ell(u) = (Mu)_\ell + \sqrt{\frac{(MM^\top)_{\ell\ell}}{2q}}\, \xi_{\ell,u}, \qquad \xi_{\ell,u} \sim \mathcal{N}(0, 1), \tag{37}$$

where q is the number of shots per homodyne setting. This propagates the vacuum covariance $I/2$ through M. We use $\eta = 1$ and independent noise for every setting. The device displacement cancels under opposite probes, so we set it to zero.

Before the closed-form method inverts a measured diagonal block, we project it onto $\mathrm{Sp}(2, \mathbb{R})$. We take its SVD, floor the singular values at 10^{-10}, reverse the direction of the smallest singular value if the determinant is negative, and normalize the determinant to one. The same single-mode correction is used in Appendix G.4.

Table 1 uses $N = 3$ and $q \in \{10^2, 10^3, 10^4, 10^5, \infty\}$. Table 2 and Fig. 5 use $q = 10^3$. Target, optimizer and measurement-noise seeds are independent across trials. One feedback loss evaluation uses eight homodyne settings, a closed-form two-mode gate uses sixteen, and a closed-form single-mode gate uses eight. The feedback optimizers use approximately 2000 loss evaluations per gate. Finite differences take 100 updates of 20 evaluations per two-mode gate and 334 updates of 6 evaluations per single-mode gate, and the genetic algorithm runs 100 generations of population 20. Dithering uses 100 updates of 20 evaluations per two-mode gate and 250 updates of 8 evaluations per single-mode gate. Matching the $G = N(N+1)/2$ trained gates to the closed-form budget gives

$$16 q_{\mathrm{cf}} \frac{N(N-1)}{2} + 8 q_{\mathrm{cf}} N = 2000 \cdot 8 q G \quad \Longrightarrow \quad q_{\mathrm{cf}} = 1000 q \frac{N+1}{N}. \tag{38}$$

Budgets are matched up to rounding of the update counts and do not represent minimum shot requirements.

Finite differences use central differences with step 0.08, Adam learning rate 0.02, gradient clipping at 10, and parameter bounds $[-5, 5]$. Dithering uses the same Adam settings, with simultaneous sinusoidal modulation of all controls at amplitude $A = 0.05$. For a window of T samples, $g_i = 2(AT)^{-1} \sum_t L(t) \chi_i(t)$, where the references χ_i are sine and cosine pairs at odd harmonics.

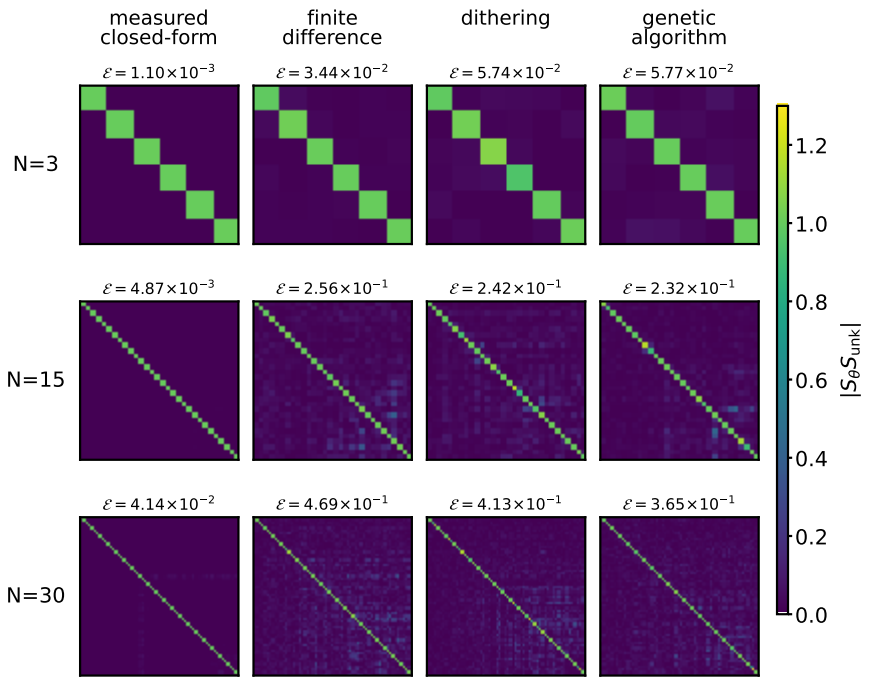


Figure 5: Response $|S_\theta S_{\text{unk}}|$ after all gates in the mesh have been trained, for the runs in Table 2 with $N = 3$, 15 and 30 (rows) and all four methods (columns). A perfect result is the identity. Each panel shows the trial whose residual is closest to that method's median in Table 2, so different columns may show different targets. The residual is printed above each panel, and all panels share a color scale.

Two-mode gates use harmonics $1, 3, 5, 7, 9$ with $T = 20$, and single-mode gates use the first three references with $T = 8$. Each sample uses fresh homodyne noise. The amplitude was selected on held-out targets starting at seed 500, with the finite-difference step checked on the same targets. These settings remain fixed in the reported sweep. The genetic algorithm uses elite fraction 0.3, parent fraction 0.5, crossover by random per-parameter averaging of two parents, mutation scale 0.04, mutation decay 0.995, and the same bounds. These hyperparameters are fixed across all runs.

G.4 Fixed calibration errors (Fig. 3)

The finite-shot experiments above assume exact gate implementation. Here we introduce fixed gate errors to compare in-situ training with tomography followed by offline compilation. Fig. 3 uses $N = 10$, 30 targets, and 13 error bounds $\epsilon_{\max} = 10^{-3+k/6}$, $k = 0, \ldots, 12$. Every gate G_g is implemented as

$$\widetilde{G}_g = E_g(\boldsymbol{\delta}_g) G_g, \qquad \delta_{g,a} \overset{\text{iid}}{\sim} \text{Uniform}[-\epsilon_{\max}, \epsilon_{\max}]. \tag{39}$$

For a two-mode gate, the ten parameters enter the Bloch–Messiah form

$$E_g = O(U_{g,1})\ \text{diag}(e^{\delta_{g,5}}, e^{-\delta_{g,5}}, e^{\delta_{g,6}}, e^{-\delta_{g,6}})\, O(U_{g,2}), \tag{40}$$

with $O(U)$ from Eq. (6), $U_{g,1} = U_{\text{err}}(\delta_{g,1}, \ldots, \delta_{g,4})$, $U_{g,2} = U_{\text{err}}(\delta_{g,7}, \ldots, \delta_{g,10})$, and

$$U_{\text{err}}(a, b, \theta, \phi) = \text{diag}(1, e^{i\phi}) \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \text{diag}(e^{ia}, e^{ib}). \tag{41}$$

For a single-mode gate, $E_g = \mathcal{R}(\delta_{g,1})\,\text{diag}(e^{\delta_{g,2}}, e^{-\delta_{g,2}})\mathcal{R}(\delta_{g,3})$. Angles are in radians and squeezing parameters are dimensionless, so a single bound applies to both. Note that $\epsilon_{\max}$ bounds each error parameter, not the size of the resulting matrix error. The errors are drawn once per trial and shared by all four curves. Across the sweep, the same draws from $\text{Uniform}[-1, 1]$ are rescaled by $\epsilon_{\max}$.

The tomography baseline estimates every entry of S_{unk} once with $\pm$ probes and projects the estimate $\widehat{S}$ onto $\mathrm{Sp}(2N, \mathbb{R})$,

$$S_{\mathrm{proj}} = \widehat{S}\left(J^{\top}\widehat{S}^{\top}J\widehat{S}\right)^{-1/2}, \tag{42}$$

using the real principal inverse square root. It then computes every gate offline from this model. After each gate, it updates the model assuming the gate was implemented exactly, so errors in earlier gates are not accounted for when later gates are computed. In-situ training instead measures the two blocks needed by each two-mode gate, and each diagonal block for the final layer, on the physical device after all earlier imperfect gates have acted.

Both methods use the minimum-squeezing gate and the single-mode correction of Appendix G.3. In-situ training uses $16N(N-1)/2 + 8N = 8N^2$ homodyne settings, the same as full tomography, and both use $q = 10^4$ shots per setting. The noiseless curves keep the same gate errors but use exact responses. All 30 trials are valid at every $\epsilon_{\max}$. The right panel reports quartiles of $\mathcal{E}_{\mathrm{tomog}} - \mathcal{E}_{\text{in-situ}}$ computed within each paired trial.

G.5 Computational resources

All simulations can run on a standard CPU. Closed-form training takes less than a second per target for every size we test. The feedback optimizers take longer, and the full set of finite-shot experiments took approximately two hours in total using 16 parallel CPU workers.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and the contribution list (Section 1) are supported by the theory (Sections 5.1–5.3, Appendices B and D), the architecture (Section 4) and the simulations (Section 6). We say explicitly that the guarantee applies to generic lossless targets and that all results come from simulation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph in Section 7 discusses optical loss (Appendix F), the strong squeezing needed near zero pivots, how errors and squeezing grow with N, and our assumption of sequential access. Appendices C, E.1 and E cover the universal variant, the conditions behind the sample-complexity result, and the training cost.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 states its genericity assumption. We sketch the proof in the main text and give the full proof in Appendix B. Proofs of the nulling criterion, the minimum-squeezing gate, the parameter count and the error bound are in Appendices B.1 and D. The sample-complexity result (Appendix E.1) is conditional, and we list its conditions there.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithm 1 and the gate constructions (Appendices A and D) define the method. Appendix G gives everything else needed to rerun the experiments: how targets are sampled, the noise and gate-error models, the matched budgets, and the optimizer hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is available at `https://github.com/janetzhong/symplectic-reck` under the MIT license, with a README, a script that runs all experiments, and unit tests. We use only synthetic targets, so no external data is needed.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We do not use data splits, since every trial samples new synthetic targets. All settings and hyperparameters are listed in Appendix G. We chose the dithering amplitude on a separate set of targets and then kept every hyperparameter fixed for the reported runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report medians with 25th–75th percentile ranges, computed directly from the samples, over 30 or 60 targets (90 pooled gates in Table 1). Each caption says which. The spread comes from random targets, shot noise, optimizer randomness and gate errors.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All simulations run on standard CPUs (Appendix G). The most expensive runs, the finite-shot experiments, took about two hours on 16 CPU workers. Preliminary runs did not use substantially more compute than this.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is entirely simulation-based, involves no human subjects or personal data, and follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is basic research on photonic circuits, done entirely in simulation, and we do not see a direct route to negative societal impact. We mention possible positive applications in Sections 7 and 8.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any models or datasets with a risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use PyTorch, NumPy and SciPy (BSD-3-Clause) and Matplotlib (BSD-compatible license) in line with their licenses. We use no external datasets or pretrained models.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The only new asset is our simulation code, released under the MIT license with a README and unit tests at `https://github.com/janetzhong/symplectic-reck`.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used to help with writing and editing, to assist with checking mathematical derivations, and to help write simulation code. The authors verified all derivations and checked the code and results, including with the released unit tests. LLMs are not a component of the proposed method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.