NANOPHOTONICS

Experimentally realized in situ backpropagation for deep learning in photonic neural networks

Sunil Pai[1]*†, Zhanghao Sun[1], Tyler W. Hughes[2]‡, Taewon Park[1], Ben Bartlett[2]†, Ian A. D. Williamson[1]§, Momchil Minkov[1]‡, Maziyar Milanizadeh[3], Nathnael Abebe[1]#, Francesco Morichetti[3], Andrea Melloni[3], Shanhui Fan[1], Olav Solgaard[1], David A. B. Miller[1]

Integrated photonic neural networks provide a promising platform for energy-efficient, high-throughput machine learning with extensive scientific and commercial applications. Photonic neural networks efficiently transform optically encoded inputs using Mach-Zehnder interferometer mesh networks interleaved with nonlinearities. We experimentally trained a three-layer, four-port silicon photonic neural network with programmable phase shifters and optical power monitoring to solve classification tasks using "in situ backpropagation," a photonic analog of the most popular method to train conventional neural networks. We measured backpropagated gradients for phase-shifter voltages by interfering forward- and backward-propagating light and simulated in situ backpropagation for 64-port photonic neural networks trained on MNIST image recognition given errors. All experiments performed comparably to digital simulations (>94% test accuracy), and energy scaling analysis indicated a route to scalable machine learning.

Neural networks (NNs) are ubiquitous computing models loosely inspired by the structure of a biological brain. Such models are trained on input data to implement complex signal processing or "inference" (*1*, *2*), powering various modern technologies ranging from language translation to self-driving cars. The required energy for training and inference to power these technologies has recently been estimated to double every 5 to 6 months (*3*), and thus necessitates an energy-efficient hardware implementation for NNs.

To address this problem, programmable photonic neural networks (PNNs) have been proposed as a promising, scalable, and mass-manufacturable integrated photonic hardware solution (*4*). A popular implementation of PNNs consists of silicon photonic meshes, $N \times N$ networks of Mach-Zehnder interferometers (MZIs) and programmable phase shifters (*5*–*7*), which optically accelerate the most expensive operation in a PNN: unitary matrix-vector multiplication (MVM). The MVM $\mathbf{y} = U\mathbf{x}$ is implemented by simply sending an input mode vector $\mathbf{x}$ (optical phases and modes in N input waveguides) through the network implementing U to yield output modes $\mathbf{y}$ (*4*, *6*, *8*). This fundamental mathematical operation, based on optical scattering theory, additionally enables various analog signal processing applications beyond machine learning (*4*, *9*) such as telecommunications (*8*), quantum computing (*10*, *11*), and sensing (*12*).

Recently, "hybrid" PNNs, which interleave programmable photonic linear optical elements (e.g., meshes) and digital nonlinear activation functions (*9*, *13*), have proven to be a low-latency and energy-efficient solution for NN inference in circuit sizes of up to $N = 64$ (*14*). Compared to current fully analog PNNs with electro-optic (EO) nonlinear activations (*15*, *16*), hybrid PNNs get around the critical problem of photonic loss and offer more versatility than multilayer PNNs for between-layer logical operations that do not favor optics. Such features may be present in a number of state-of-the-art machine learning architectures such as recurrent neural networks (*17*) and transformers (*18*, *19*). When fully optimized, the energy efficiency of PNN inference has been estimated to be up to two orders of magnitude higher than that of state-of-the-art digital electronic application-specific integrated circuits (ASICs) in artificial intelligence (AI) (*20*). However, despite the success in PNN-based inference, efficient on-chip training of PNNs has not been demonstrated owing to substantially higher experimental complexity compared to the inference procedure.

In this study, we experimentally demonstrated a photonic implementation of backpropagation, the most widely used method of training NNs (*1*, *2*). [A minimal bulk optical demonstration has been previously explored (*21*).] Backpropagation is generally performed by propagating error signals backward through the NNs to determine programmable parameter gradients via the chain rule. In our multilayer PNN device, we performed in situ training on a foundry-manufactured silicon photonic integrated circuit by sending light-encoded errors backward through the PNN and measuring optical interference with the original forward-going "inference" signal (*22*). Once trained, our chip achieved an accuracy similar to that of digital simulations, adding new capabilities beyond existing inference or in silico learning demonstrations (*4*, *23*, *24*). We further designed and experimentally validated an analog (electro-optic) phase-shifter update protocol, a key improvement over past proposals requiring more energy-intensive "digital subtraction" (*22*). Finally, we systematically analyzed energy and latency advantages of in situ backpropagation and its scalability to larger (64×64) PNN systems. Our findings ultimately pave the way for energy-efficient optoelectronic training of neural networks and optical systems more broadly.

Photonic neural networks

We built a hybrid PNN by alternating sequences of analog programmable unitary MVM operations $U^{(\ell)}\left(\vec{\eta}^{(\ell)}\right)$ [implemented on a custom-designed silicon photonic triangular mesh (*6*)] and digital nonlinear transformations $f^{(\ell)}$ [implemented using autodifferentiation software (*25*–*27*)] where layer $\ell \leq L$ (total of L layers). The PNN was parameterized by programmable phase shifts $\vec{\eta} \in [0, 2\pi)^D$, where D represents number of PNN phase shifters. Mathematically, the following "inference" function sequence transformed input $\mathbf{x} = \mathbf{x}^{(1)}$, proceeding in a "feedforward" manner to the output $\hat{\mathbf{z}} := \mathbf{x}^{(L+1)}$ (Fig. 1, A to D):

$$\mathbf{y}^{(\ell)} = U^{(\ell)}\mathbf{x}^{(\ell)} \tag{1}$$

$$\mathbf{x}^{(\ell+1)} = f^{(\ell)}\left(\mathbf{y}^{(\ell)}\right) \tag{2}$$

The "cost function" is defined as $\mathcal{L}(\mathbf{x}, \mathbf{z}) = c\left(\hat{\mathbf{z}}(\mathbf{x}), \mathbf{z}\right)$, where c represents the error between $\hat{\mathbf{z}}$ and ground truth label $\mathbf{z}$. Backpropagation updates parameters $\vec{\eta}$ that are on D-dimensional gradient $\partial\mathcal{L}/\partial\vec{\eta}$ evaluated for "training example" $(\mathbf{x}, \mathbf{z})$ (or averaged over a batch of examples).

Each MZI was parametrized by thermo-optic phase shifters that locally heat the waveguides using current sourced from a separate control driver board (Fig. 2, A and B). Phase shifts were placed at the input (ϕ, voltage V_ϕ) and internal (θ, voltage V_θ) arms of all MZIs to control the propagation pattern of infrared C band (1530 to 1565 nm) light, enabling arbitrary unitary matrix multiplication. We embedded an arbitrary 4×4 unitary matrix multiply in a 6×6 triangular network of MZIs. This configuration incorporated two 1×5 photonic meshes on either end of the 4×4 "matrix unit" capable of sending any input vector $\mathbf{x}$ and measuring any output vector $\mathbf{y}$ from Eqs. 1 and 2. These "generator" and "analyzer" optical input/output (I/O) circuits (Figs. 1E and 2B and fig. S5) require calibrated voltage mappings $\theta(V_\theta), \phi(V_\phi)$ to control optical phase (*4*, *28*, *29*) (fig. S2).

Backpropagation demonstration

Our core result (Fig. 1E) was experimental realization of backpropagation on a photonic

[1]Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA. [2]Department of Applied Physics, Stanford University, Stanford, CA 94305, USA. [3]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy.
*Corresponding author. Email: spai@psiquantum.com
†Present address: PsiQuantum, Palo Alto, CA, USA.
‡Present address: Flexcompute Inc., Belmont, MA, USA.
§Present address: X Development LLC, Mountain View, CA USA.
#Present address: Google, Mountain View, CA USA.

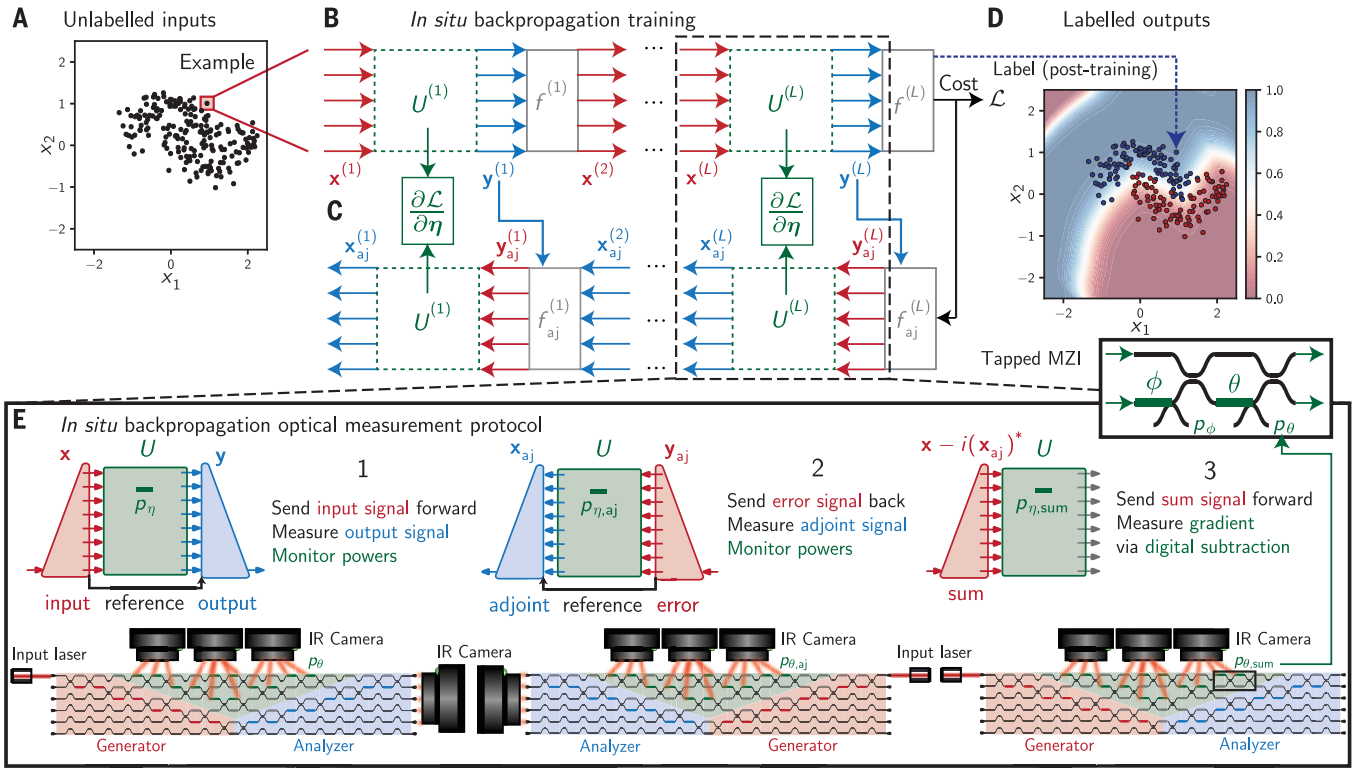

Fig. 1. In situ backpropagation concept. (**A**) Example machine learning problem: An unlabeled 2D set of points that are formatted to be input into a PNN. (**B**) In situ backpropagation training of an L-layer PNN for the forward direction and (**C**) the backward direction showing the dependence of gradient updates for phase shifts on backpropagated errors. (**D**) An inference task implemented on the actual chip resulted in good agreement between the chip-labeled points and the ideal implemented ring classification boundary (resulting from the ideal model) and a 90% classification accuracy. (**E**) Our proposed scheme performed the three steps of in situ (analog) backpropagation, using a 6 × 6 mesh implementing coherent 4 × 4 bidirectional unitary matrix-vector products using a reference arm. The (1) forward, (2) backward, and (3) sum steps of in situ backpropagation are shown. Arbitrary input setting and complete amplitude and phase output measurement were enabled in both directions using the reciprocity and symmetries of the triangular architecture. All powers throughout the mesh were monitored by an IR camera using the tapped MZI shown in the inset for each step, allowing for digital subtraction to compute the gradient (*22*). These power measurements performed at phase shifts are indicated by green horizontal bars.

triangular mesh MVM chip using a custom optical rig and silicon photonic chip (fig. S1) (*22*). Our backpropagation-enabled architecture differs in three ways from a typical PNN photonic mesh (*4*):

1) We enabled "bidirectional light propagation," the ability to send and measure light propagating left to right or right to left through the circuit (as depicted in Fig. 1E).

2) We implemented "global monitoring" to measure optical power p_η propagating through any phase shift η in the circuit using 3% grating taps (shown in the inset of Fig. 1E and Fig. 2, A and B). In our proof-of-concept setup, we used an infrared (IR) camera mounted on an automated stage to image these taps throughout the chip (fig. S1E).

3) We implemented both amplitude and phase detection [improving on past approaches (*30*)] using a self-configuring programmable matrix unit layer (*28*) on both generator and analyzer subcircuits (Figs. 1E and 2B and fig. S5), which by symmetry worked for sending and measuring light that propagated forward or backward through the mesh.

These improvements on an already versatile hardware platform enabled backpropagation entirely using physical optical power measurements to obtain cost gradients (*22*). As shown in Fig. 1E, backpropagation required global optical monitoring, and bidirectional optical I/O was required to switch between forward- and backward-propagating signals to experimentally realize in situ backpropagation. Equipped with these additional elements, our protocol can be implemented on any feedforward photonic circuit (*31*) with the requisite analyzer and generator circuitry (Fig. 1 and fig. S5).

Here we give a brief summary of the procedure (further explained in the supplementary text). The "forward inference" signal $\mathbf{x}^{(\ell)}$ and "backward adjoint" signal $\mathbf{x}_{\mathrm{adj}}^{(\ell)}$ are sent forward and backward, respectively, through the mesh that implements $U^{(\ell)}$. The "sum" vector $\mathbf{x}^{(\ell)} - i(\mathbf{x}_{\mathrm{adj}}^{(\ell)})^*$ is sent forward, and subtracting the forward and backward measurements from it digitally yields the gradient (*22*), a reverse-mode differentiation process that we call an "optical vector-Jacobian product (VJP)."

Analog update

Going beyond an experimental implementation of a past theoretical proposal (*22*), we additionally explored a more energy-efficient fully analog gradient measurement update for the final step, avoiding a digital subtraction update. Instead of global monitoring optical power in the first two steps and the final "sum" step, we toggled an adjoint phase $\zeta(t)$, a square wave modulation with period T that periodically toggles between "sum" and "difference" settings $\zeta = 0$ and π corresponding to signal inputs $\mathbf{x}_{\pm}^{(\ell)} = \mathbf{x}^{(\ell)} \mp i(\mathbf{x}_{\mathrm{adj}}^{(\ell)})^*$. The gradient is $\partial\mathcal{L}/\partial\eta = \left(p_{\eta,+} - p_{\eta,-}\right)/4$, or half the "signed amplitude" of the AC (mean-subtracted) signal (supplementary text 2.6 and fig. S6). The sum and difference inputs $\mathbf{x}_{\pm}^{(\ell)}$ were computed digitally (off-chip), requiring $\mathcal{O}(N)$ operations to compute per input. The sum and difference inputs were directly programmed at the generator to compute phase gradients, and corresponding sum and difference signal power measurements at each phase shifter subtracted in the analog domain to update phase-shift voltages. One option to efficiently achieve a periodic

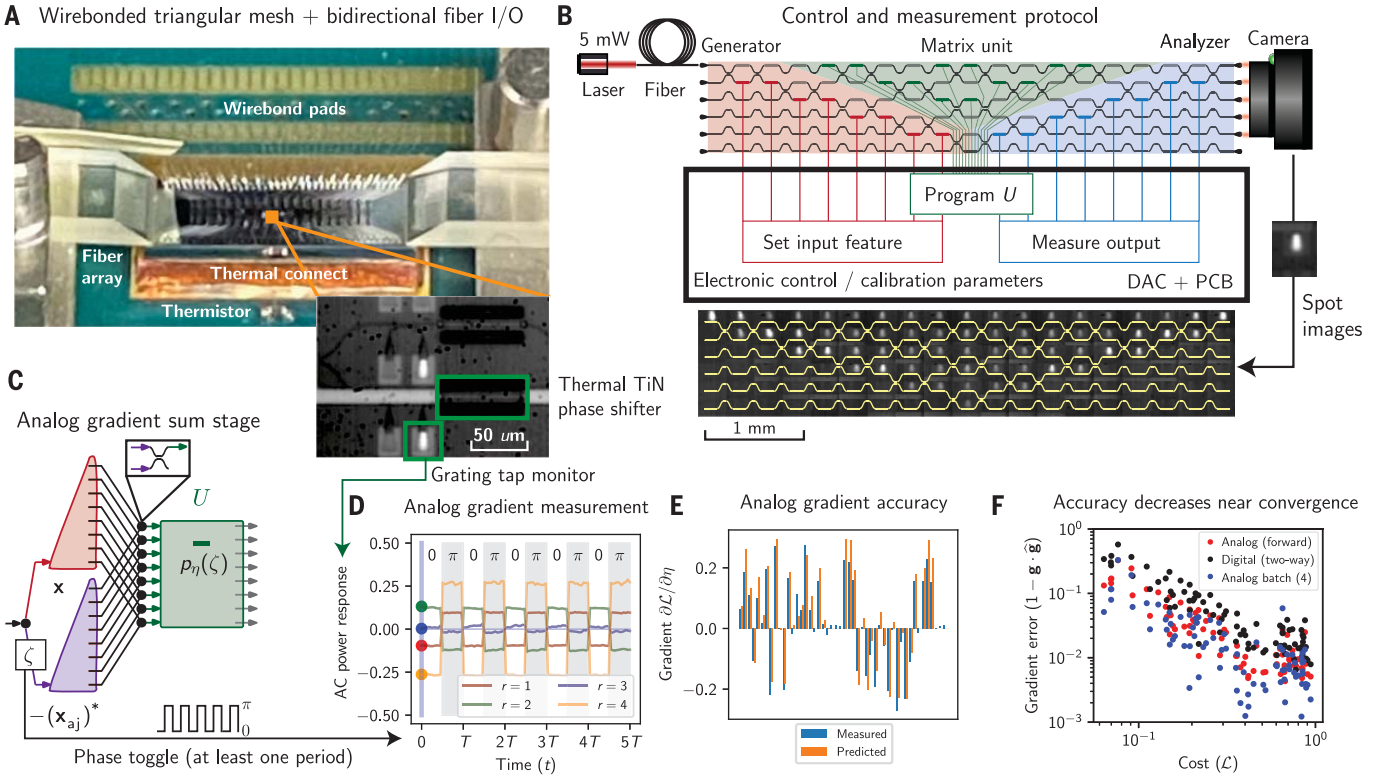

Fig. 2. Analog gradient experiment and simulation. (**A**) The photonic mesh chip was thermally controlled and wirebonded to a custom printed circuit board (PCB) with fiber array for laser input/output and a camera overhead for imaging the chip. Zooming in (IR camera image) reveals the core control-and-measurement unit of the chip, enabling power measurement using 3% grating tap monitors and a thermal TiN phase shifter nearby. (**B**) A 5-mW 1560-nm laser and a calibrated control unit was used for input generation and output detection. The IR camera over the chip imaged all grating tap monitors necessary for backpropagation. (**C**) Analog gradient update might optionally be implemented by introducing a summing interference circuit [not implemented on the chip in (B)] between the input and adjoint fields. (**D**) The adjoint phase was toggled between $\zeta = 0$ and π to evaluate the analog gradient measurement $\partial\mathcal{L}_i/\partial\eta$ for $i = 1$ to 4. (**E**) Gradients measured using the toggle scheme yielded approximately correct gradients when the implemented mesh was perturbed from the optimal (target) unitary given 1 rad phase error standard deviation. (**F**) Measured normalized gradient error decreased with cost function [distance between implemented $\hat{U}(\vec{\eta})$ and optimal $U = \mathrm{DFT}(4)$], and analog batch and single-example gradients outperformed digital gradients.

ζ toggle is to use the summing architecture in Fig. 2C, which sums $\mathbf{x}^{(\ell)}$ and $i(\mathbf{x}_{\mathrm{adj}}^{(\ell)})^*$ interferometrically with a fast modulator that implements ζ. In an optimized scheme, we would physically measure the gradient and update the phase-shift voltage in the analog domain using a photodiode, differential amplifier (implementing an analog subtraction), and a "sample-and-hold" update circuit using only a single toggle (fig. S6, B and C). This scheme, extended to energy-efficient "batch updates" incorporating data from multiple training examples, was tested on a single phase shifter to demonstrate the logic of this electronic feedback scheme (materials and methods, supplementary text 2.6, and fig. S7). Our demonstration avoided a costly digital-analog and analog-digital conversion; when fully integrated, our approach avoids additional digital memory complexity required to program N^2 elements, enabling a truly analog backpropagation scheme.

The local feedback just described updates each phase shifter η using the measured gradient:

$$\frac{\partial\mathcal{L}}{\partial\eta} = \mathcal{I}\left(x_\eta x_{\eta,\mathrm{adj}}\right) = \frac{|x_{\eta,+}|^2 - |x_\eta|^2 - |x_{\eta,\mathrm{adj}}|^2}{2} = \frac{p_{\eta,+} - p_\eta - p_{\eta,\mathrm{adj}}}{2} = \frac{p_{\eta,+} - p_{\eta,-}}{4} \quad (3)$$

where the sum field $x_{\eta,+} = x_\eta - ix^*_{\eta,\mathrm{adj}}$ and the last equality of Eq. 3 indicate the mathematical equivalence of "digital subtraction" (Fig. 1E) and our proposed "analog subtraction" scheme (Fig. 2, C and D, and figs. S6 and S7). Pseudocode and the complete backpropagation protocol are provided in supplementary text 2.5. Digital and analog gradient update steps can both be implemented in parallel across all PNN layers once the measurements from forward and backward steps are determined.

We experimentally estimated the accuracy of the analog gradient measurement for a matrix optimization problem (7) by digital processing of the optical power measurements (Fig. 2D). We programmed a sequence of inputs into the generator unit of our chip and recorded the square-wave response oscillating between $p_{\eta,+}$ and $p_{\eta,-}$ and separately subtracted the two measurements to find the gradient with respect to η.

We implemented in situ backpropagation in a single photonic mesh layer, optimizing the cost function defined for output port i via $\mathcal{L}_r = 1 - |\hat{\mathbf{u}}_r^T\mathbf{u}_r^*|^2$ or a "batch" cost function $\mathcal{L} = \sum_{r=1}^{4}\mathcal{L}_r/4$ averaged over four inputs ("batch size" $M = 4$). Here, $\mathbf{u}_r$ is row r of U, a target matrix that we chose to be the four-point discrete Fourier transform [DFT(4)], and $\hat{\mathbf{u}}_r$ is row r of $\hat{U}$, the implemented matrix on the device. For our gradient measurement step, we sent in the derivative $\mathbf{y}_{\mathrm{adj}} = \partial\mathcal{L}_r/\partial\mathbf{y} = -2(\hat{\mathbf{u}}_r^T\mathbf{u}_r^*)^*\mathbf{e}_r$ to measure an adjoint field $\mathbf{x}_{\mathrm{adj}}$, where $\mathbf{e}_r$ is the rth standard basis vector (1 at position m, 0 everywhere else).

We evaluated gradient direction error as $1 - \mathbf{g}\cdot\hat{\mathbf{g}}$ comparing normalized measured ($\hat{\mathbf{g}}$) and predicted gradients $\mathbf{g} = \partial\mathcal{L}/\partial\vec{\eta}\cdot\|\partial\mathcal{L}/\partial\vec{\eta}\|^{-1}$. Both digital and analog gradients were less accurate near convergence, with

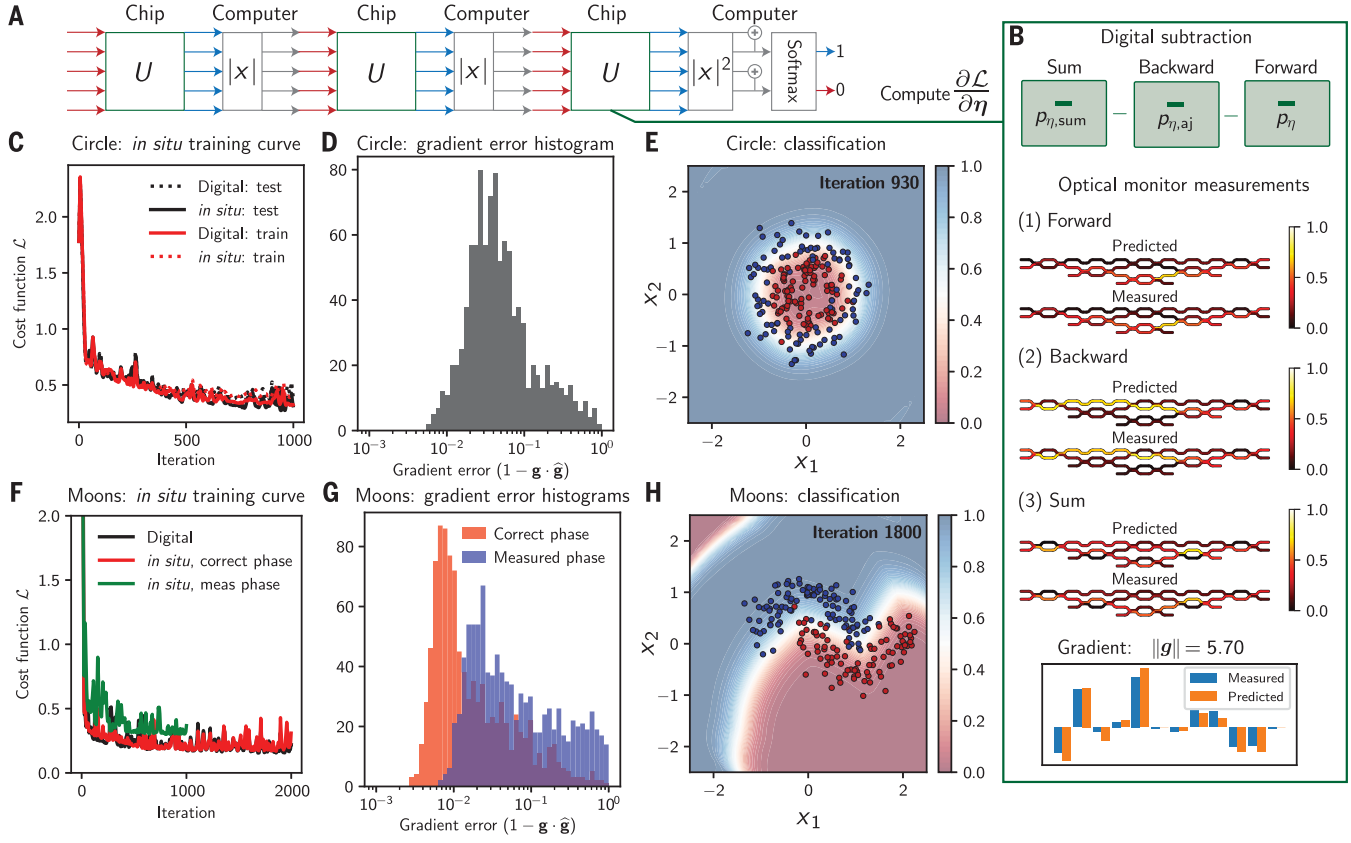

Fig. 3. In situ backpropagation experiment. In situ backpropagation training (*34*) was performed for two classification tasks solvable by (**A**) a three-layer hybrid PNN consisting of absolute-value nonlinearities and a softmax (effectively sigmoid) decision layer. (**B**) Three-step digital subtraction gradient update given monitored waveguide powers and the measured gradient output. (**C**) For the circle dataset, the digital and in situ backpropagation training curves show excellent agreement resulting in (**D**) model accuracy of 96% test and 93% train (depicted here for iteration 930, showing the true labels and the learned classification model outcomes) and (**E**) histogram of low gradient error. (**F**) For the moons dataset, our phase measurements were sufficiently inaccurate owing to hardware error affecting training, leading to a lower model model accuracy of 94% test and 87% train (green). Using ground truth phase (red), the device achieved (**G**) sufficiently high model accuracy of 98% test and 95% train. (**H**) The histogram of gradient errors improved considerably by roughly an order of magnitude using the correct phase measurement.

the errors empirically decreasing quadratically with cost $\mathcal{L}$ (Fig. 2F). The analog batch gradient (trained by averaging all four gradients to give $\partial\mathcal{L}/\partial\eta$) validated the photonic portion of the batch scheme (figs. S6B and S7). All gradient errors, regardless of implementation, scaled similarly with convergence distance; uncalibrated thermal cross-talk likely resulted in gradient measurement errors that were comparable to systematic power errors at the taps. Digital subtraction encountered different losses and coupling efficiencies in bidirectional tap gratings, whereas analog gradient measurements involved subtraction of only forward-going fields at forward gratings, likely resulting in superior performance (Fig. 2F). Finally, error in the full analog subtraction scheme was independent of batch size for the gradient calculation, and no significant deviation due to timing jitter or signal distortion was observed (fig. S7).

Photonic neural network training

To test overall on-chip training, we assessed the accuracy of in situ backpropagation to train multilayer PNNs using a digital subtraction protocol (*22*) (Fig. 3A and fig. S3) automated with Python software (*32*). We trained our chip to implement $L = 3$ layers with $N = 4$ ports to assign labeled noisy synthetic data, generated using Scikit-Learn (*33*), in 2D space to a 0 or 1 label based on the data points' spatial location (Figs. 1A; 3, E and H; and fig. S4, I and J). We performed an 80%:20% train–test split (200 train points, 50 test points) and trained on only train points to avoid overfitting.

To implement classification, our PNN assigned a probability to each point being assigned a 0 or 1 on the basis of the following model:

$$\hat{\mathbf{z}}(\mathbf{x}) = \text{softmax2}(\left|U^{(3)}\right|U^{(2)}\left|U^{(1)}\mathbf{x}\right|) \quad (4)$$

where softmax2 is the standard softmax (normalized sigmoid) function applied to two quantities: the total power in outputs 1 and 2 and total power in ports 3 and 4. The input data $\mathbf{x}$ was engineered such that any 2D point had the same total input power as a four-port vector (materials and methods). Each point was classified red or blue (0 or 1, respectively) on the basis of whether the output of Eq. 4 obeyed the condition $z_0 > z_1$ for each input (Fig. 3), which we optimized using a binary cross-entropy cost function (materials and methods).

Our chip performed data input, output, and matrix operations for all PNN layers. At each layer output, we digitally performed a square-root operation on output power to implement absolute-value nonlinearities [off-chip via JAX and Haiku (*26*, *27*)] and recorded output phases for the backward pass of in situ backpropagation. Ideally, PNNs are controlled by separate photonic meshes of MZIs for each linear layer to achieve low power consumption. However, to save on carbon footprint, we reprogrammed the same chip to perform successive linear layers because basic operating principles remain the same. We used the Adam gradient update (*34*) with a learning rate of 0.01 and performed digital simulations at each step to fully compare measured and predicted performance. Before on-chip training experiments, we calibrated all phase shifters on the chip (materials and methods and fig. S2) and performed forward inference with digitally pretrained neural network weights to verify

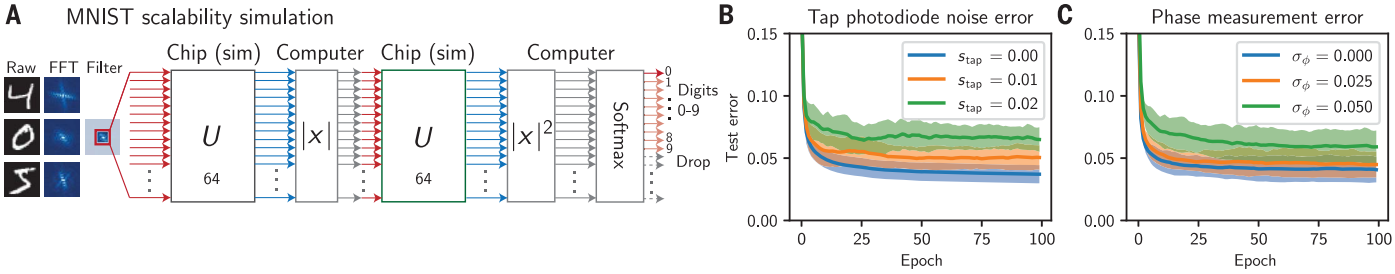

Fig. 4. In situ backpropagation simulation. (**A**) A two-layer PNN was simulated on MNIST data using a previously explored PNN benchmark incorporating rectangular photonic meshes (*31*). (**B** and **C**) Marginal training curve statistics (shaded regions indicate standard deviation error range about the mean) were computed over a grid search of 72 tap noise, loss, and I/O amplitude and phase errors (materials and methods). The dominant contributers were (B) tap noise factor s_{tap} (2.7% increase for $s_{tap} = 0.02$ from 3.7±0.7% average error) and (C) phase measurement error σ_ϕ (1.9% increase for $\sigma_\phi = 0.05$ from 4±1% average error).

accurate calibration. We achieved 90% and 98% device test set accuracy for ring and moons datasets, respectively (fig. S4, I and J). Because our photonic and digital implementation agreed closely in inference accuracy, we performed network training on-chip while conducting evaluations off-chip for convenience.

During training of the circle dataset, predicted and measured powers for grating tap-to-camera monitor measurements showed excellent agreement across all waveguide segments required for accurate gradient computation (Fig. 3B, fig. S3, and movie S1). The training curves in Fig. 3C indicate that stochastic gradient descent was a highly noisy training process for both predicted and measured curves owing to the noisy synthetic dataset about the boundary and our choice of single-example training as opposed to batch training. These large swings appeared roughly correlated between the simulated and measured training curves (Fig. 3E), and we successfully achieved 93% train and 96% test model accuracy (Fig. 3D and fig. S4, A to C). We then trained the moons dataset, applying the same procedure to achieve 87% train and 94% test model accuracy (Fig. 3F, green versus red). When using the predicted phase and measured amplitudes, we reduced gradient error by roughly an order of magnitude on average, resulting in 95% train and 98% test model accuracy (fig. S4, D to F), which agreed with digital training (Fig. 3, F to H, and movie S2). This improvement underscores the importance of accurate phase measurement for improved training efficiency. Further monitoring errors could be reduced by increasing signal-to-noise ratio using integrated avalanche photodiodes (*35*), noninvasive light monitoring (*36*), or phase shifter–based power monitoring (*37*).

Simulations and scalability

Given that our experimental results for $N = 4$ PNNs showed evidence of hardware error affecting training, we assessed the scalability for $N = 64$ PNNs on the MNIST handwritten digit dataset (*38*) in the presence of error to better understand the relative contributions at scale. We implemented a PNN simulation framework in Simphox (*25*) using JAX and Haiku (*26*, *27*) to simulate an in situ backpropagation training given a grid search of systematic and noise errors (materials and methods). After 100 epochs using $M = 600$ batch size, we achieved a maximum test accuracy of roughly 97.2% in the ideal case and a performance degradation to roughly 95% on average (Fig. 4, B and C). Phase and amplitude errors arising from photodetector noise and phase-shift quantization and calibration errors affected convergence in error the most. Overall, our MNIST simulation results suggest that in situ backpropagation is relatively robust at scale to noise and hardware errors, which are difficult to eliminate completely in current analog computing systems.

We also considered the energy and latency trade-off with accuracy for the optimized analog gradient update scheme assuming current state-of-the-art electronics cointegrated with active photonic components (supplementary text 2.7). Collectively, our simulation results (Fig. 4) and energy calculation contours (fig. S8, supported by tables S1 to S6) indicated minimal performance degradation for MNIST training simultaneously with threefold improvement in backpropagation energy efficiency. This assumed 100-fJ floating point operations for equivalent digital models (*39*) and tap noise factor of $s_{tap} < 0.01$ in the regime where optical power begins to dominate the energy consumption. Errors may be further reduced by improving avalanche photodiode sensitivity, reducing optical component loss, or increasing overall input optical power, a key factor in the energy-error trade-off (tables S1 to S6). Trade-off of input power and photodiode noise generally enforces a hard limit on scalability of photonic meshes (i.e., number of MZI layers N) because all photonic components have loss (*16*, *40*).

Discussion and outlook

In this study, we have demonstrated practically useful photonic machine learning hardware by physically measuring gradients calculated through interferometric measurements of in situ backpropagation (Fig. 1). We concluded that gradient accuracy played an important role in reaching optimal results during training and decreases near convergence (Fig. 2). As a core application, we trained multilayer PNNs using our gradient measurements and found good agreement with digital training simulations despite optical I/O calibration errors and camera noise at the global monitoring taps (Fig. 3). Correcting for phase measurement error yielded training curves highly correlated to digital predictions, so optical I/O calibration accuracy is vital. Even though individual updates were ideally faster to compute, higher error resulted in effectively longer training times that mitigated this benefit. To better understand this trade-off, we explored an optimized regime of our system, which considered cointegration of complementary metal-oxide semiconductor (CMOS) electronics with photonics (fig. S8 and tables S1 to S6), and found that in the regime of photonic advantage (e.g., $N = 64$ at sufficiently large batch sizes), we could successfully train MNIST close to digital equivalents (Fig. 4).

Our demonstration (Fig. 3) and energy calculations (fig. S8) suggest that in situ backpropagation, a technique widely used in machine learning for its efficiency, also efficiently trains hybrid PNNs. Our hybrid approach optically accelerated the most computationally intensive $\mathcal{O}(N^2)$ operations, whereas nonlinearities and their derivatives, which are $\mathcal{O}(N)$ computations, were implemented digitally. This is reasonable because $\mathcal{O}(N)$ time is required to modulate and measure optical inputs and outputs for the overall network, regardless of hybrid or all-analog operation. Because optics is ideal for low-latency and low-energy signal communication, our in situ backpropagation scheme could improve energy efficiency in data center machine learning and neural network accelerators (e.g., graphics processing units) with optical interconnects, in which data are already optically encoded. Such schemes may be compatible with mixed-signal schemes for accelerators that already aim to reduce the current communication energy bottleneck (*39*, *41*) in the race to address the energy-doubling AI problem (*3*).

Population-based methods (*42*), direct feedback alignment (*43*, *44*), and perturbative approaches (*16*) have some advantages but are ultimately less efficient for training neural networks compared to backpropagation, especially for hybrid PNNs. Unlike "receiverless" fully analog PNNs (*16*), hybrid PNNs require optoelectronic (i.e., digital-analog and analog-digital) conversions for each layer, which can slow down perturbative training. In contrast to perturbative approaches, in situ backpropagation calculates gradients in a modular framework compatible with larger-scale AI applications.

Although this work primarily dealt with hybrid PNNs, our backpropagation scheme could be compatible with all-analog or receiverless implementations implementing EO nonlinearities on-chip (*15*, *16*, *45*). Previous all-analog PNN implementations have suffered from exponential loss scaling because the same optical modes propagated through all L layers (*16*). We propose to reduce this scaling from exponential to linear by instead splitting input light equally across the layers and modulating each layer input by EO activations that depend on other layer output powers, which acts to "connect" the layers without an explicit optical connection (fig. S9, A and H). After incorporating electronic and optical switches, this "distributed nonlinearity" architecture can operate as a hybrid PNN platform for training or an all-analog platform for inference with full visibility of EO nonlinearity response to aid backpropagation training (fig. S9, B to G). The scaling and errors of these schemes, given the need to accurately model nonlinear activations for backpropagation, are left to a future work.

Ultimately, these all-analog schemes suffer from limited versatility to manipulate or transform data. Depending on the problem or architecture, "hybridizing" the all-optical PNN with digital platforms can add some flexibility when convenient at the expense of optoelectronic conversion energy. For instance, flexibility of large-scale hybrid PNN models has been demonstrated via high ResNet-50 image classification accuracy using commercially viable photonic meshes (*14*). Our experimental demonstration indicates a route to train such models on backpropagation-enabled devices that few other training methods can efficiently produce. In situ backpropagation can also train "optical transformers" that leverage hybrid PNNs for natural language processing and computer vision applications (*19*). The periodic application of digital activations, currently infeasible in optics [e.g., layer normalization (*19*)], enables one-to-one correspondence of hybrid PNNs and state-of-the-art large-scale NN models.

Our demonstration is an experimental analog of "inverse design" of photonic devices. Inverse design implements reverse-mode autodifferentiation with respect to material relative permittivity by interfering adjoint and forward fields. This forms the basis of the original proof of in situ backpropagation (*22*) because phases are trivially related to material relative permittivity changes. This suggests an even broader application domain for our technique to optimizing arbitrary programmable linear optical devices with no obvious calibration scheme, including robust designs (e.g., using multiport directional couplers) and recirculating designs (*46*, *47*). The analog gradient update experiment in Fig. 2 is relevant to calibration (*6*) because minimizing the cost function $\mathcal{L}$ maximizes device fidelity.

Our results ultimately have wide-ranging implications for bridging the fields of photonics and machine learning. Backpropagation is the most efficient and widely used neural network training algorithm for machine learning, and our demonstration of this popular echnique as a physical implementation presents promising capabilities of hybrid PNNs to reduce carbon footprint and counter the exponentially increasing costs of AI computation.

REFERENCES AND NOTES

1. S. Linnainmaa, *BIT* **16**, 146–160 (1976).
2. D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Nature* **323**, 533–536 (1986).
3. J. Sevilla *et al*., *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy (2022), pp. 1–8.
4. Y. Shen *et al*., *Nat. Photonics* **11**, 441–446 (2017).
5. M. Reck, A. Zeilinger, H. J. Bernstein, P. Bertani, *Phys. Rev. Lett.* **73**, 58–61 (1994).
6. D. A. B. Miller, *Photon. Res.* **1**, 1 (2013) [Invited].
7. S. Pai, B. Bartlett, O. Solgaard, D. A. B. Miller, *Phys. Rev. Appl.* **11**, 064044 (2019).
8. A. Annoni *et al*., *Light Sci. Appl.* **6**, e17110 (2017).
9. W. Bogaerts *et al*., *Nature* **586**, 207–216 (2020).
10. J. Carolan *et al*., *Science* **349**, 711–716 (2015).
11. B. Bartlett, S. Fan, *Phys. Rev. A* **101**, 042319 (2020).
12. M. Milanizadeh *et al*., *Light Sci. Appl.* **11**, 197 (2022).
13. N. C. Harris *et al*., *Optica* **5**, 1623 (2018).
14. C. Ramey, "Silicon Photonics for Artificial Intelligence Acceleration (Lightmatter)" in *IEEE Hot Chips 32 Symposium (HCS)* (2020), pp. 1–26.
15. I. A. D. Williamson *et al*., *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–12 (2020).
16. S. Bandyopadhyay *et al*., arXiv:2208.01623 [cs.ET] (2022).
17. L. Jing *et al*., in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia (2017), vol. 70, pp. 1733–1741.
18. A. Vaswani *et al*., *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017).
19. M. G. Anderson, S.-Y. Ma, T. Wang, L. G. Wright, P. L. McMahon, arXiv:2302.10360 [cs.ET] (2023).
20. M. A. Nahmias *et al*., *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–18 (2020).
21. A. A. Cruz-Cabrera *et al*., *IEEE Trans. Neural Netw.* **11**, 1450–1457 (2000).
22. T. W. Hughes, M. Minkov, Y. Shi, S. Fan, *Optica* **5**, 864 (2018).
23. L. G. Wright *et al*., *Nature* **601**, 549–555 (2022).
24. J. Spall, X. Guo, A. I. Lvovsky, *Optica* **9**, 803–811 (2022).
25. S. Pai, simphox: Another inverse design library [Computer software]; https://github.com/fancompute/simphox (2022).
26. J. Bradbury *et al*., JAX: composable transformations of Python+NumPy programs [Computer software]; https://github.com/google/jax (2022).
27. T. Hennigan, T. Cai, T. Norman, I. Babuschkin, Haiku: Sonnet for JAX, [Computer software]; https://github.com/deepmind/dm-haiku (2020).
28. D. A. B. Miller, *Optica* **7**, 794 (2020).
29. M. Prabhu *et al*., *Optica* **7**, 551 (2020).
30. H. Zhang *et al*., *Nat. Commun.* **12**, 457 (2021).
31. S. Pai *et al*., *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–13 (2020).
32. S. Pai, solgaardlab/photonicbackprop: Adding some new analog gradient measurement data (0.0.3), Zenodo (2023); https://doi.org/10.5281/zenodo.6557413.
33. F. Pedregosa *et al*., *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
34. D. P. Kingma, J. L. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations, 7 to 9 May 2015, San Diego.
35. J. K. Perin, M. Sharif, J. M. Kahn, *J. Lightwave Technol.* **34**, 5542–5553 (2016).
36. F. Morichetti *et al*., *IEEE J. Sel. Top. Quantum Electron.* **20**, 292–301 (2014).
37. S. Pai *et al*., *Nanophotonics* **12**, 985–991 (2023).
38. L. Deng, *IEEE Signal Process. Mag.* **29**, 141–142 (2012).
39. D. A. Miller, *J. Lightwave Technol.* **35**, 346–396 (2017).
40. S. Pai *et al*., *Optica* 10.1364/OPTICA.476173 (2023).
41. B. Murmann, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **29**, 3–13 (2021).
42. H. Zhang *et al*., *ACS Photonics* **8**, 1662–1672 (2021).
43. A. Nøkland, in *Proceedings of the 30th Conference on Neural Information Processing Systems*, D. D. Lee *et al*., Eds. (Curran Associates, 2016), pp. 1045–1053.
44. M. J. Filipovich *et al*., *Optica* **9**, 1323–1332 (2022).
45. X. Guo, T. D. Barrett, Z. M. Wang, A. I. Lvovsky, *Photon. Res.* **9**, B71–B80 (2021).
46. D. Pérez *et al*., *Nat. Commun.* **8**, 636 (2017).
47. R. Tang, R. Tanomura, T. Tanemura, Y. Nakano, *ACS Photonics* **8**, 2074–2080 (2021).
48. S. Pai, Z. Sun, T. Park, phox: Base repository for simulation and control of photonic devices [Computer software], https://github.com/solgaardlab/phox/ (2022).
49. S. Pai, N. Abebe, dphox: photonic layout and device design [Computer software], https://github.com/solgaardlab/dphox (2022).

ACKNOWLEDGMENTS

We acknowledge Advanced MicroFoundries (AMF) in Singapore for help in fabricating and characterizing the photonic circuit for our demonstration and Silitronics for help in packaging our chip for our demonstration. Thanks also to P. Broaddus for helping with wafer dicing; S. Lorenzo for help in fiber splicing the fiber switch for bidirectional operation; J. Kahn for guidance on avalanche photodetector noise estimates; N. Pai for advice on electronics, scalability, and electrical and thermal control packaging; R. Quan for help in building our all-analog gradient measurement electronics; and C. Langrock and K. Urbanek for help in building our movable optical breadboard. **Funding:** We acknowledge funding from Air Force Office of Scientific Research (AFOSR) grants FA9550-17-1-0002 in collaboration with UT Austin and FA9550-18-1-0186 through which we share a close collaboration with UC Davis under B. Yoo. **Author contributions:** S.P. ran all experiments with input from Z.S., T.W.H., T.P., B.B., I.A.D.W., N.A., M. Minkov, O.S., S.F., and D.A.B.M. S.P., T.W.H., M. Minkov, and I.A.D.W. conceptualized the experimental protocol. S.P., N.A., F.M., M. Milanizadeh, and A.M. contributed to the design of the photonic mesh. S.P. and Z.S. wrote code to control the photonic integrated circuit active elements and camera detection and electronic circuit for analog gradient measurement. T.P. designed the custom PCB with input from S.P. S.P. wrote the manuscript with input from all coauthors. All coauthors contributed to discussions of the protocol and results. **Competing interests:** S.P., Z.S., T.W.H., I.A.D.W., M. Minkov, S.F., O.S., and D.A.B.M. have filed a patent for the analog backpropagation update protocol discussed in this work with provisional application no. 63/323743. D.M. holds two related patents on the SVD architecture: US Patent no. 10,877,287 and no. 10,534,189. The authors declare no other conflicts of interest. **Data and materials availability:** Materials and methods are available as supplementary materials. All other software and data for running the simulations and experiments are available through Zenodo (*32*) and Github through the Phox framework, including our experimental code via Phox (*48*), simulation code via Simphox (*25*), and circuit design code via Dphox (*49*).

SUPPLEMENTARY MATERIALS

science.org/doi/10.1126/science.ade8450
Materials and Methods
Supplementary Text
Figs. S1 to S9
Tables S1 to S6
References (*50*–*76*)
Movies S1 and S2

Submitted 27 September 2022; accepted 8 March 2023
10.1126/science.ade8450