nature photonics

Supplementary information

https://doi.org/10.1038/s41566-024-01567-z

Single-chip photonic deep neural network with forward-only training

In the format provided by the authors and unedited

CONTENTS

I. DEVICE CHARACTERIZATION

Transmitter

A Mach-Zehnder interferometer (MZI) performs the programmable 2×2 unitary operation:

$$U(\theta_1, \theta_2) = ie^{i\theta_1/2} \begin{bmatrix} e^{i\theta_2}\sin(\theta_1/2) & e^{i\theta_2}\cos(\theta_1/2) \\ \cos(\theta_1/2) & -\sin(\theta_1/2) \end{bmatrix} \tag{S1}$$

To characterize a single MZI, we first input light into one port of the device and measure the output transmission $T(\theta_1) = P_{\text{out}}/P_{\text{in}}$. For an ideal device, the output transmission at the bar port is

$$T_{\text{bar}}(\theta_1) = \sin^2\left(\frac{\theta_1}{2}\right) \tag{S2}$$

and at the cross port:

$$T_{\text{cross}}(\theta_1) = \cos^2\left(\frac{\theta_1}{2}\right) \tag{S3}$$

In a fabricated MZI, θ_1 is determined by the total dissipated power $I \times V(I)$, where I is the programmed current and $V(I)$ is the voltage dropped across the device. To characterize a device in the transmitter, where the cross port of each channel has a photodiode, we sweep I and measure the voltage $V(I)$ and output transmission $T(I)$. We fit the expressions:

$$V(I) = a_4 I^4 + a_3 I^3 + a_2 I^2 + a_1 I \tag{S4}$$

$$T_{\text{cross}} = A + B\cos\left(\frac{IV(I)}{P_\pi}\pi + p_0\right) \tag{S5}$$

where $a_4, a_3, a_2, a_1, A, B, P_\pi, p_0$ are fitting parameters. Here, P_π is the total dissipated power required to induce a π phase shift, p_0 is the static phase difference between the two interferometer arms, and $1/(A-B)$ is the interferometer extinction ratio. We found that a fourth-order polynomial was required to fit the voltage-current relationship of the heaters, which became non-Ohmic at high currents due to self-heating and velocity saturation of the carriers. If we measured $T(I)$ at the bar port, the latter expression would instead be:

$$T_{\text{bar}} = A - B\cos\left(\frac{IV(I)}{P_\pi}\pi + p_0\right) \tag{S6}$$

This yields a mapping between the current I and the programmed phase $\theta_1(I)$ of the form:

$$\theta_1(I) = p_4 I^4 + p_3 I^3 + p_2 I^2 + p_1 I + p_0 \tag{S7}$$

Coherent matrix multiplication unit (CMXU)

The procedure for characterizing the CMXU is sketched in Figure S1. We use the photodiodes at the output of each matrix processor; for the first two layers, we use the photodiode at each nonlinear optical function unit (NOFU), while the last layer is calibrated with the system's receiver.

In an uncalibrated mesh, light will scatter randomly through the circuit as p_0 is random for each device. To characterize the circuit, we first input light into the top input (input 1) of the mesh and measure the transmission at the bottom output (output 6). We then optimize the internal phase shifters along the main diagonal in a round robin fashion to maximize the signal at output 6. This procedure deterministically initializes the main diagonal to the cross state ($\theta_1 = 0$), as there is only one possible path between input 1 and output 6. Having initialized the diagonal, we can then calibrate each device along it by sweeping the phase shifter θ_1, measuring T at output 6, and fitting equation S5 to the data. The antidiagonal, connecting input 6 to output 1, is calibrated in the same way.

Having characterized the main diagonals, the remainder of the devices can be calibrated in a similar fashion. For instance, inputting light into mode 1 and setting the top left MZI in the circuit, which is already calibrated, to the bar state provides

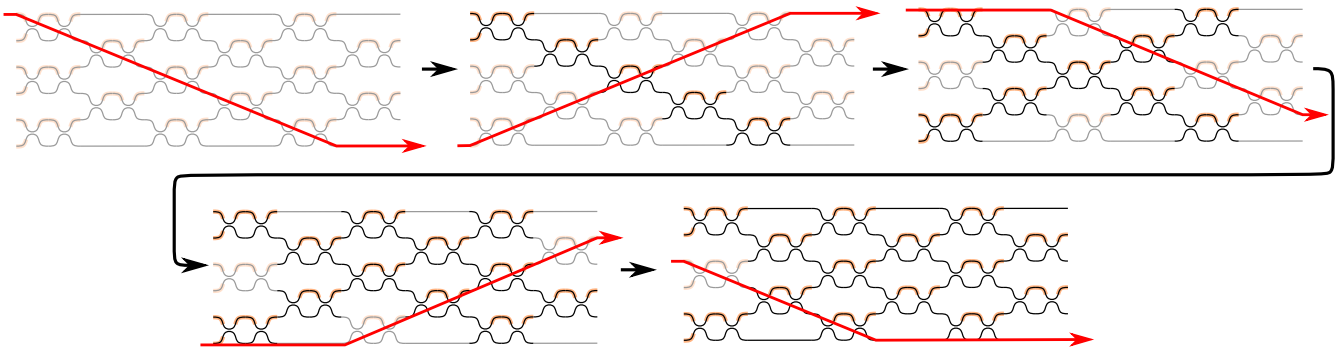

FIG. S1: Calibration procedure for internal phase shifters in the CMXU. The devices along the main diagonal and antidiagonal are calibrated first. Once these devices are characterized, the remainder of the phase shifters can be calibrated by programming devices along the main diagonal.

access to the first subdiagonal. The uncalibrated devices can then be characterized with the same procedure as was used for the main diagonal. We show the full calibration sequence in Figure S1.

This protocol calibrates all internal phase shifters θ_1 in a matrix processor. The external phase shifters θ_2 are calibrated using "meta-MZIs," as shown in Figure S2. A "meta-MZI" consists of two MZIs in columns $i-1, i+1$ that are programmed to implement a 50-50 beamsplitter ($\theta_1 = \pi/2$). This subcircuit now functions as an effective MZI, where the relative phase difference between two external phase shifters $\theta_{2,a}, \theta_{2,b}$ is equivalent to the setting of the internal phase shifter in a discrete device.

We fix one of the two external phase shifters to $I = 0$, sweep the current programmed into the other, and measure the output transmission T. Fitting the data to equations S5, S6, depending on the port T is measured out of, calibrates the static phase difference $\Delta\phi(I = 0) = \theta_{2,b}(I = 0) - \theta_{2,a}(I = 0)$. Repeating this procedure for all devices produces a linear system of equations that can be inverted to find the static phase p_0 for each external heater. More details on this procedure can be found in [S1].

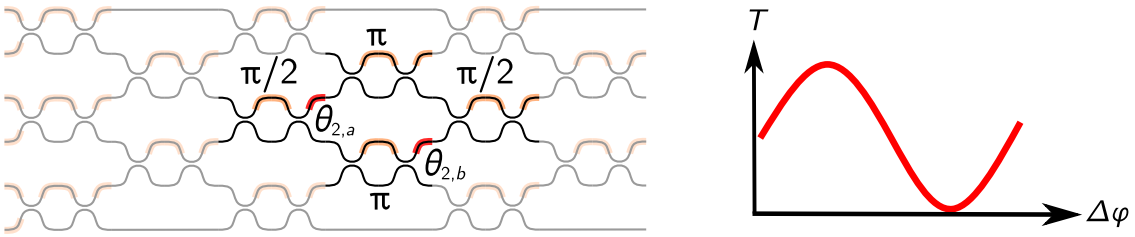

FIG. S2: "Meta-MZI" for calibrating external phase shifters. Two phase shifters in columns $i-1, i+1$ are set to implement a 50-50 beamsplitter. The output transmission of this meta-interferometer, which functions exactly like a discrete MZI, is dependent on the phase difference between the external phase shifters $\Delta\phi = \theta_{2,b} - \theta_{2,a}$.

II. NONLINEAR OPTICAL FUNCTION UNIT

Figure S3 shows the measured response of a typical nonlinear optical function unit. The resonator is designed to be overcoupled, as the injection of carriers increases the round-trip loss in the ring. Thus, as current is increased, the resonator transitions through the critical coupling regime to being undercoupled at large incident powers.

The phase response and round-trip attenuation a as a function of the photocurrent I are shown in Figure S3c and d, respectively. Assuming a photodiode responsivity of ~ 1 A/W, we find that about 75 μW is sufficient to detune the NOFU by a linewidth. As we bias the device to 0.8 V in our experiment, the power consumption during operation is therefore ~ 60 μW.

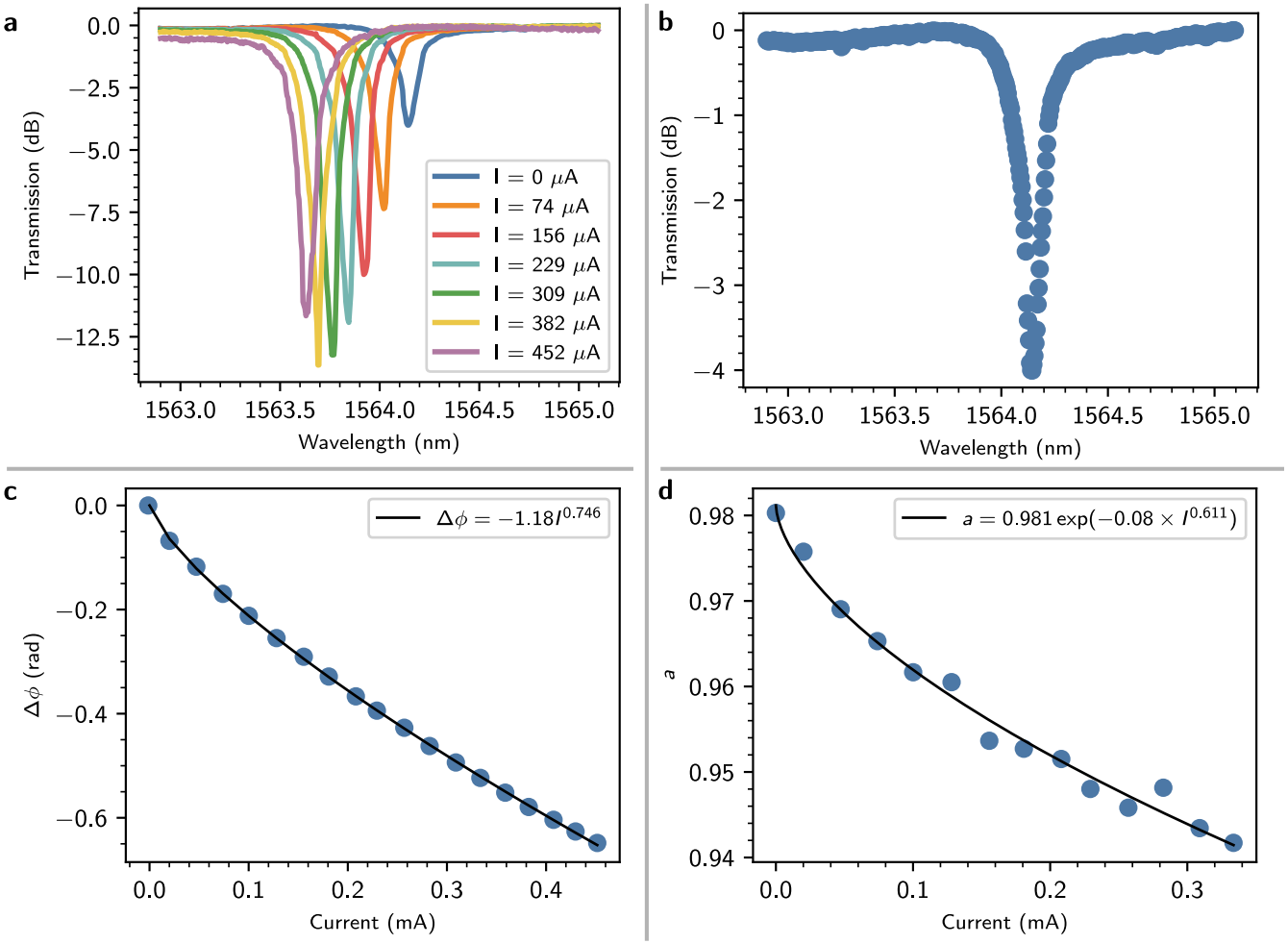

FIG. S3: a) NOFU response vs. injected photocurrent. We assume a photodiode responsivity of 1 A/W. The resonator is initially overcoupled, and transitions to undercoupling as the incident optical power is increased. b) NOFU resonance when no power is incident on the photodiode. c) Phase shift $\Delta\phi$ in cavity vs. incident photocurrent. d) Round-trip amplitude loss a as a function of incident photocurrent. As photocurrent increases more carriers are injected into the waveguide, increasing the loss of the optical signal inside the resonator.

III. CORRECTING HARDWARE ERRORS

Static component error, such as errors in beamsplitters or transmission losses, affect the accuracy of matrices programmed into the CMXU. Since we use thermal phase shifters for programming the matrix processors, thermal crosstalk between devices will also impact the performance of the system. In this section, we outline the hardware error correction procedures used to obtain high matrix accuracies in the FICONN.

Transmitter correction

For the transmitter, we corrected thermal crosstalk between components by directly measuring the 12×6 crosstalk matrix M, where M_{ij} denotes the crosstalk on channel i produced by an aggressor channel j. This quantity was measured by driving channel i and measuring the output transmission T at different current settings for channel j. For each measurement, we fit equation S5 to the data to extract the static phase p_0. Thermal crosstalk will cause p_0 to vary as a function of the settings in channel j due to parasitic heating; we fit a linear expression to this data to extract the crosstalk coefficient M_{ij}.

Figure S4a shows an example of this procedure, where we extracted the crosstalk on channel 1 produced by channel 2.

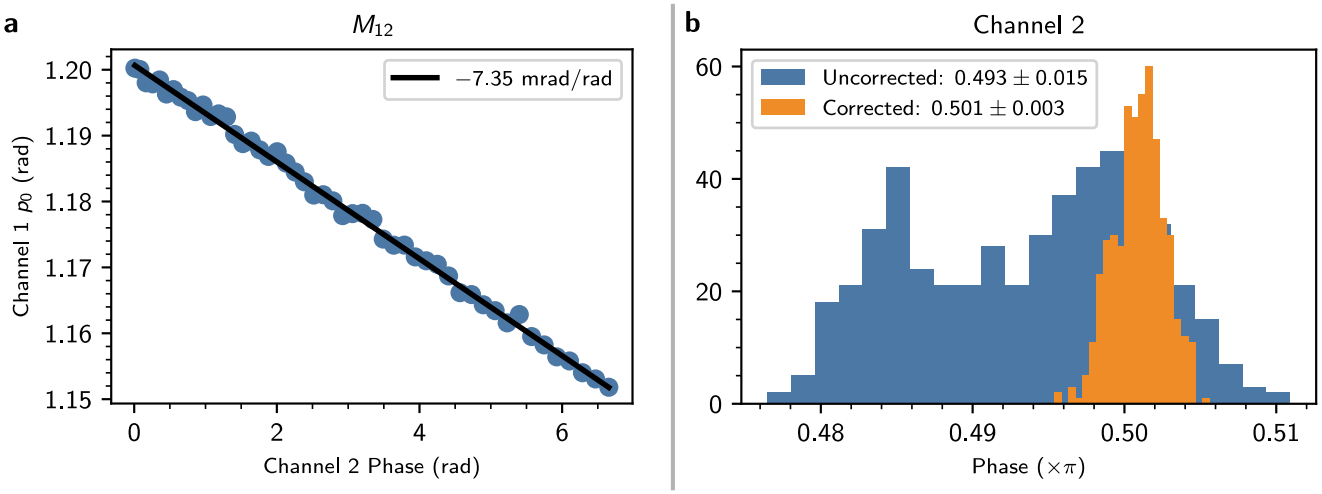

FIG. S4: a) To determine the elements of the thermal crosstalk matrix M, we drive an aggressor channel j while characterizing the static phase p_0 of channel i. As an example, here we characterize M_{12} by plotting the static phase of channel 1 as a function of the phase setting of channel 2. We fit a linear function to this data to find a crosstalk coefficient of $M_{12} = -0.00735$. b) We benchmark the effectiveness of thermal crosstalk correction by repeatedly trying to program a channel to $\theta_1 = \pi/2$, while setting all other channels to random values. We then determine the actual phase implemented by measuring the output transmission T and computing $2\arccos\sqrt{T}$. As an example, here we show the results for channel 2, where over 500 random experiments thermal crosstalk correction greatly improves the repeatability of programming a channel to a desired phase.

Having obtained M, we can now obtain the phase settings $\mathbf{\Phi}$ for a desired programming $\mathbf{\Phi}'$ by computing:

$$\mathbf{\Phi} = M^{-1}(\mathbf{\Phi}' - \mathbf{\Phi}_0) + \mathbf{\Phi}_0 \tag{S8}$$

where $\mathbf{\Phi}_0$ is the static phase for each channel. We neglected crosstalk on the external phase shifters of the transmitter, which program the phase of the input $\mathbf{a}^{(1)}$, as we did not have coherent detection directly at the transmitter output.

In order to benchmark the correction protocol for each transmitter channel we repeatedly attempted to program $\theta_1 = \pi/2$ while setting all other channels to a random phase setting. Shown in Figure S4b is the phase setting actually implemented by the transmitter channel for 500 such experiments, which we extracted by measuring the transmission T and computing $2\arccos\sqrt{T}$. Thermal crosstalk correction greatly improves both the accuracy and repeatability of each channel; for example, the measured phase on channel 2 improves from 0.493 ± 0.015 to 0.501 ± 0.003 following correction.

CMXU correction

Correcting for thermal crosstalk in the CMXU is more challenging. As the CMXU is a mesh of interferometers, changing the programming of aggressor channels can introduce phases and redirect light through the circuit in unexpected ways. These effects are challenging to disentangle from pure thermal crosstalk when the circuit also has other component errors, such as beamsplitter imperfections and device loss, making it difficult to directly measure M.

To address this, we instead developed a "digital twin" of the hardware, which modeled in software the response of a device with known beamsplitter errors, waveguide losses, and thermal crosstalk. As the effects of all of these imperfections are known *a priori* for Mach-Zehnder interferometer meshes [S2, S3], we can fit a software model, where these imperfections are initially unknown model parameters, to data taken on the real device. If the software model can accurately reproduce measurements from the hardware, the parameters found to describe the device imperfections can be used to deterministically correct errors on the real hardware. We note here that our approach is not a "black-box" or neural network model of the device. Our model is based on the physics of how Mach-Zehnder interferometer meshes behave, and thus the parameters we find are realistic and correspond to true physical attributes of the device, such as the error for a particular directional coupler.

We fit the model to a dataset obtained by programming 300 random unitary matrices into the chip and measuring the response to 100 randomly selected input vectors. Our software model, which is written in `JAX` for auto-differentiability, is fit to the measured data using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm. We found that our software model was able to predict hardware outputs with an average fidelity $F = \mathrm{Tr}[U^\dagger_{\text{measured}} U_{\text{software}}]/N$ of 0.969 ±

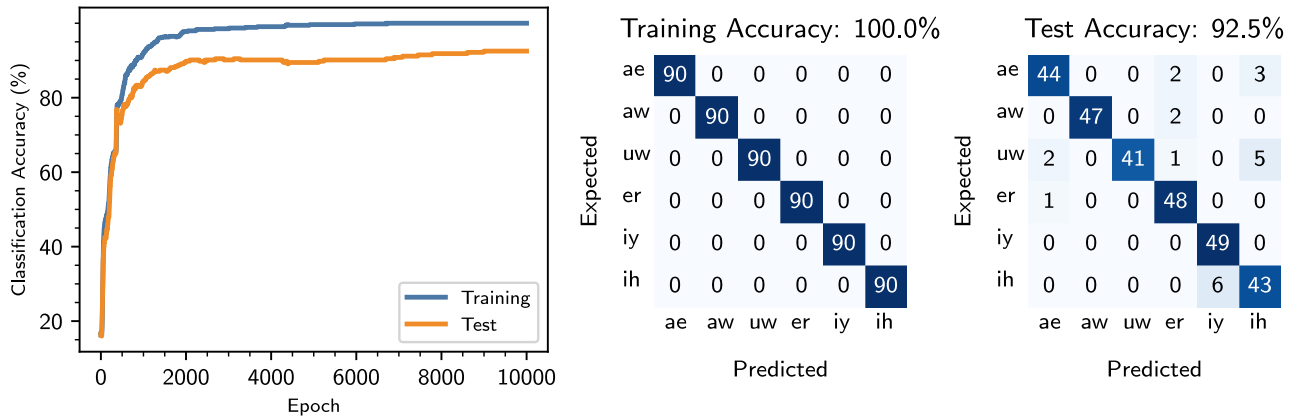

FIG. S5: Performance of the digital model on the vowel classification task. The model overfits the training set, achieving 100% accuracy, but performance on the test set is comparable to the accuracy achieved by our system (92.5% on the digital model vs. 92.5% on the FICONN).

0.023. The data shown in Figure 2d of the main text was taken by first implementing a "direct" programming, where the phase shifter settings were obtained using the Clements decomposition, and then by using the settings obtained from the software model.

IV. DIGITAL MODEL FOR VOWEL CLASSIFICATION

We trained a digital model on the vowel classification task to benchmark the performance of *in situ* training on our system. The two models had the same number of neurons ($3 \times 6^2 = 108$), but the weights of the digital model, unlike those of our system, were unconstrained and could be arbitrary real matrices. We trained the system with a tanh nonlinearity, as we obtained very poor performance on the test set using a ReLU function. When training, we normalized the output with a softmax function and used the categorical cross-entropy loss function.

The performance of the digital model is shown in Figure S5. We find that both digital and photonic systems realize comparable classification accuracies on the test set.

V. COMPARISON TO LINEAR CLASSIFIER

To confirm the computational efficacy of the NOFU, we trained a single-layer, linear classifier on the same vowel task used in our experiments. This linear classifier models the FICONN without NOFUs, as cascading multiple unitary layers without nonlinearity is computationally equivalent to a single unitary layer. Here, we train only the last CMXU in the system, while setting the first two meshes to implement identity operations that route inputs to the last layer. The nonlinear units for this experiment are turned off by setting the tap fraction to the photodiode to 0.

The results are shown in Figure S6. We obtain 81.6% accuracy on the test set, which is an accuracy penalty of nearly 11% relative to the full system and confirms that the FICONN is implementing a nonlinear optical transformation for the vowel classification task.

As the CMXU implements unitary weighting, we also compared the performance to a real-valued digital linear classifier, which can realize a greater variety of linear transformations. To ensure the best possible performance of the real-valued classifier, we trained it using backpropagation on a digital system and compared it to our experimental DNN accuracies obtained with *in situ* training. Despite the relative simplicity of the vowel task, this classifier still achieves lower accuracies on the test set, confirming that the FICONN, which implements both linear and nonlinear functions in the optical domain, outperforms a digital classifier that only performs linear transformations.

We also observed the FICONN, incorporating both linear and nonlinear transformations, trained much more efficiently than the linear classifier both on chip and in software. This also suggests that the multi-layer DNN we implement on chip spans a larger parameter space, enabling faster convergence to the solution.

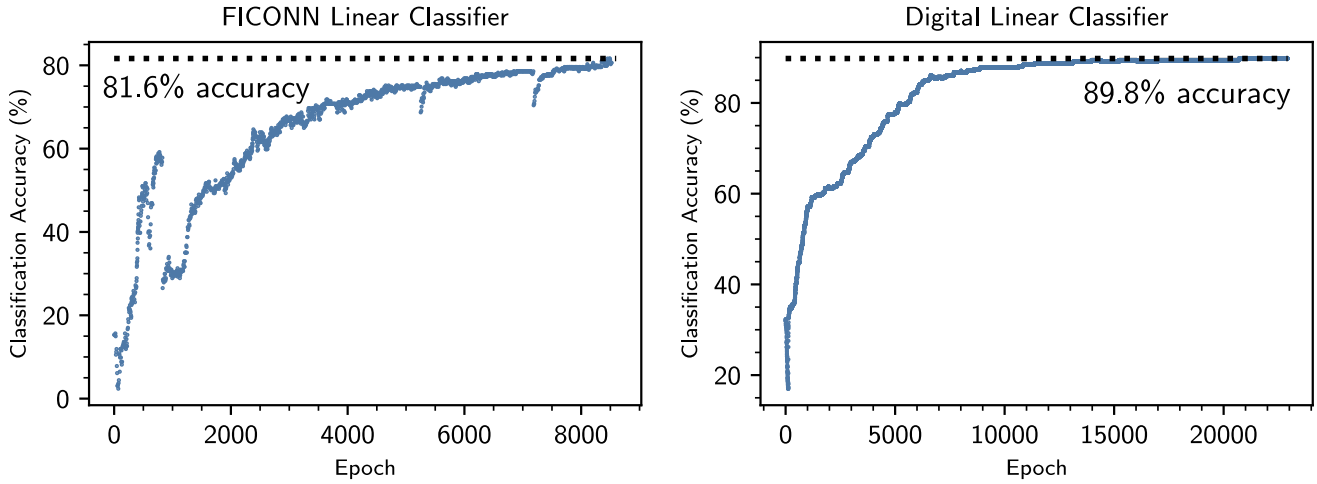

FIG. S6: Performance of a linear classifier trained on the hardware (left) and a digital, real-valued linear classifier (right). The linear classifier on the hardware implements unitary weighting, which accounts for the lower accuracy relative to the real-valued digital classifier. However, both classifiers exhibit lower accuracy relative to the multi-layer, nonlinear DNN implemented in our experiments.

VI. STOCHASTIC OPTIMIZATION PERFORMS GRADIENT DESCENT ON AVERAGE

Unlike other derivative-free optimizers, the advantage of the stochastic optimization approach we use is that it performs gradient descent on average. Here we illustrate this by adapting the proof provided in [S4].

Suppose we are optimizing a DNN with model parameters $\mathbf{\Theta}$ and error functional $\mathcal{L}(\mathbf{\Theta})$. Gradient descent iteratively optimizes the parameters with the update rule:

$$\Delta\mathbf{\Theta} = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{\Theta}} \tag{S9}$$

Assuming $\eta > 0$ and is sufficiently small, this update rule will converge to a local minimum of $\mathcal{L}(\mathbf{\Theta})$. Finite difference methods for analog hardware attempt to compute $\partial\mathcal{L}/\partial\mathbf{\Theta}$ by perturbing one parameter at a time in the system. Each epoch therefore requires $2N$ evaluations of the model on the training set for N model parameters.

Alternatively, one could perturb all model parameters at once by a random vector $\mathbf{\Pi} = [\pi_1, \pi_2, ..., \pi_N]$, where the elements of $\mathbf{\Pi}$ are randomly and independently chosen from an N-dimensional hypercube. The update rule here is:

$$\Delta\mathbf{\Theta} = -\mu\frac{\mathcal{L}(\mathbf{\Theta}+\mathbf{\Pi}) - \mathcal{L}(\mathbf{\Theta}-\mathbf{\Pi})}{2||\mathbf{\Pi}||}\mathbf{\Pi} \tag{S10}$$

$$= -\frac{\mu}{2|\pi|\sqrt{N}}[\mathcal{L}(\mathbf{\Theta}+\mathbf{\Pi}) - \mathcal{L}(\mathbf{\Theta}-\mathbf{\Pi})]\mathbf{\Pi} \tag{S11}$$

where μ is the learning rate. Assuming that the elements of $\mathbf{\Pi}$ are independently drawn from a Bernoulli distribution as $\pm\pi$, we can substitute $||\mathbf{\Pi}||$ as $|\pi|\sqrt{N}$. We note here that $\mu \neq \eta$, and in practice μ can be much larger than η while preserving stable convergence.

We Taylor expand the expression $[\mathcal{L}(\mathbf{\Theta}+\mathbf{\Pi}) - \mathcal{L}(\mathbf{\Theta}-\mathbf{\Pi})]$ as:

$$2\sum_i \frac{\partial\mathcal{L}}{\partial\theta_i}\pi_i \tag{S12}$$

Substituting this into the update rule, we get:

$$\Delta\mathbf{\Theta} = -\frac{\mu}{|\pi|\sqrt{N}}\left(\sum_i \frac{\partial\mathcal{L}}{\partial\theta_i}\pi_i\right)\mathbf{\Pi} \tag{S13}$$

$$= -\frac{\mu}{|\pi|\sqrt{N}}\left(\sum_i \frac{\partial\mathcal{L}}{\partial\theta_i}\pi_i\right)[\pi_1, \pi_2, ..., \pi_N] \tag{S14}$$

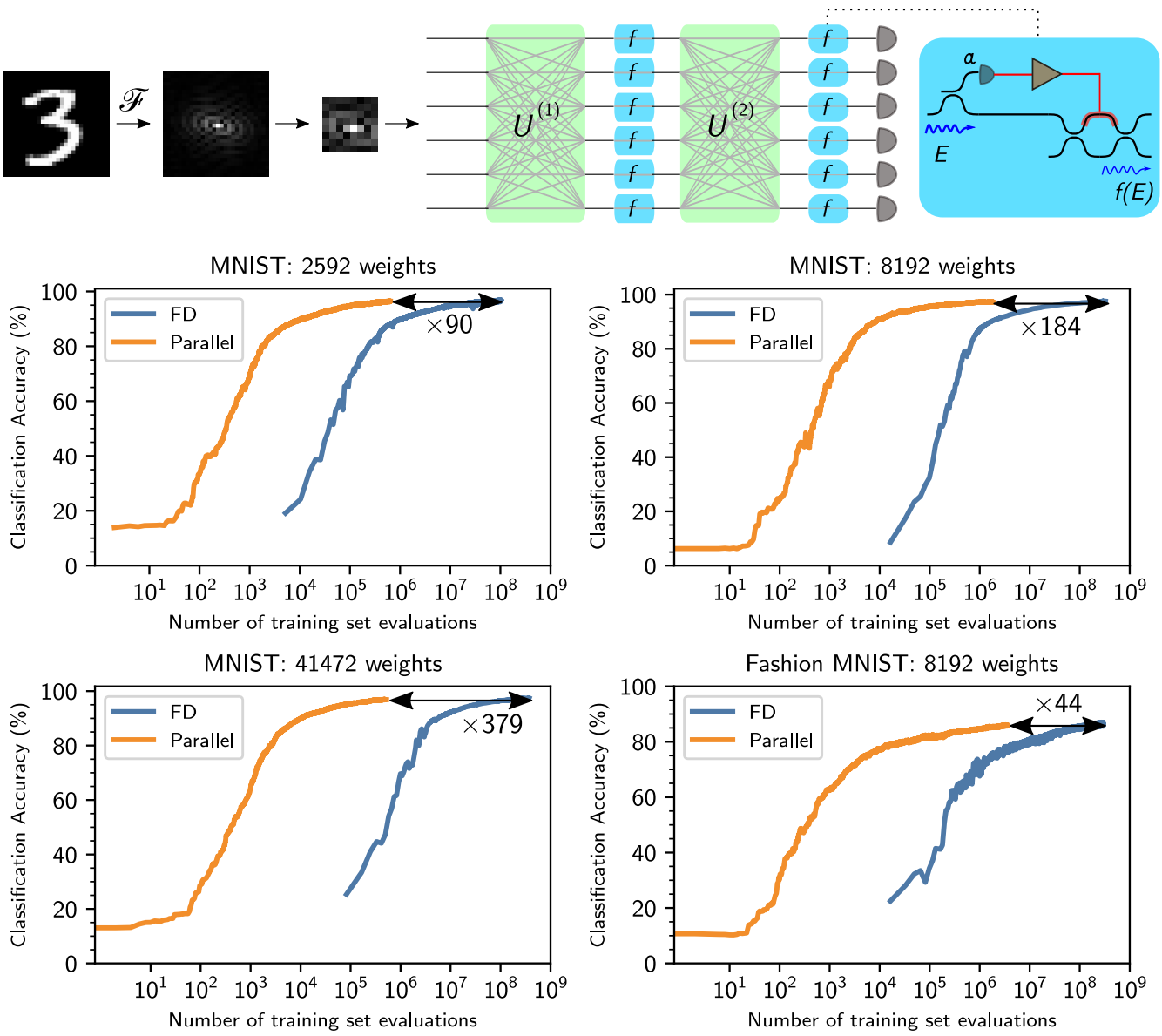

FIG. S7: Training efficiency of parallel stochastic optimization vs. finite difference (FD) on the MNIST and Fashion MNIST classification tasks. Input images are pre-processed with a Fourier transform and spatially filtered to fit onto the input layer of the DNN. We model fully-connected layers of the DNN with unitary meshes, while the nonlinearity is modeled as a photodetector driving a Mach-Zehnder modulator, as in refs. [S2, S5]. Our results suggest that parallel perturbation out-performs finite difference by at least 40× for the Fashion MNIST task and by more than 90× for the MNIST task.

Since the π_i are independently chosen, $\mathrm{E}[\pi_i \pi_j] = 0$ if $i \neq j$. Therefore, the expected parameter update $\mathrm{E}[\Delta\boldsymbol{\Theta}]$ is:

$$\mathrm{E}[\Delta\boldsymbol{\Theta}] = -\frac{\mu}{|\pi|\sqrt{N}} \left(\sum_i \frac{\partial \mathcal{L}}{\partial \theta_i} \mathrm{E}[\pi_i^2] \hat{x}_i \right) \tag{S15}$$

$$= -\frac{\mu}{|\pi|\sqrt{N}} \left(\sum_i \frac{\partial \mathcal{L}}{\partial \theta_i} |\pi|^2 \hat{x}_i \right) \tag{S16}$$

$$= -\frac{\mu|\pi|}{\sqrt{N}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}} \tag{S17}$$

We therefore find that, on average, this procedure performs gradient descent with an effective learning rate $\eta = \mu|\pi|/\sqrt{N}$.

In order to evaluate the performance of this approximate gradient against using the true gradient, as would be computed in a finite difference algorithm, we compared in simulation the relative training time of a two-layer deep neural network on the MNIST and Fashion MNIST classification tasks. Here, we use the same DNN architecture as reported in [S2, S5]; each weight layer is implemented with a rectangular unitary mesh, while the nonlinear activation function is modeled as a photodiode driving a Mach-Zehnder modulator. The input images are pre-processed with a Fourier transform and spatially low-pass filtered to fit onto the input layer of the DNN.

As the final weight layer is square and the MNIST task has only ten output classifications, we truncate the output optical powers to the first ten outputs. This reduces the number of effective weights in the final layer by $\approx 1/2$, as only phase shifters in the upper left triangle of the mesh will affect the result. This can be understood by considering that light input into the bottom mode can travel to the first output and influence the classification result, but light input into the bottom mode and traveling to the bottom output will not. In the results presented in Figure S7 the total number of weights listed does not account for this, as the training algorithm does not know *a priori* that some of the phase shifter values do not impact the cost function.

The finite difference simulations are run using backpropagation, under the assumption that the true gradient can be evaluated on hardware with $2N$ evaluations of the training set, while the stochastic optimization algorithm, which requires 2 evaluations of the training set per epoch, is implemented in the same way as it is in our experiment. In order to ensure a fair comparison in training time, we varied the learning rate for both algorithms to obtain the highest possible rate with stable training and compared the time required for both methods to reach within 1% of the final test set accuracy obtained with gradient descent. For a clear algorithmic comparison we did not batch the training set, although we still obtained over 96% (87%) accuracy on the MNIST (Fashion MNIST) test set, indicating that the model generalizes well.

In Figure S7, we show the total number of evaluations of the training set (which, assuming a fixed inference clock rate, corresponds to the total training time) required to reach a specified classification accuracy on the test set; we find that for a 36×36 size two-layer DNN, corresponding to $N = 2592$ weights, the stochastic optimization algorithm is nearly two orders of magnitude more efficient. The improvement is even larger for 64×64 and 144×144 networks ($N = 8192$ and $N = 41472$, respectively), where the improvement in the number of training set evaluations is more than two orders of magnitude. Empirically, we observed roughly a $\sqrt{N}$ speedup on MNIST relative to finite difference, suggesting that as model sizes grow it is increasingly more efficient to train using stochastic optimization. Other implementations of stochastic optimization, such as updating only some of the weight parameters per epoch, i.e. batching parameter updates, or other types of perturbations such as sinusoidal or "code" perturbations [S6], could alternatively be employed to realize speedups over finite difference. We also trained a 64×64 DNN for the Fashion MNIST task, which is more difficult; here, we find that stochastic optimization is still 40 times more efficient than finite difference.

VII. LATENCY AND ENERGY EFFICIENCY

Latency

The latency of the FICONN is the time-of-flight through the photonic circuit. To confirm this, we experimentally measured the computational latency of the NOFU, which is dependent on both the electrical and optical response times of the system, and the end-to-end latency of the entire system.

Computational latency of the NOFU–injection mode

The NOFU can be operated either in an injection mode, where charge is directly injected into the waveguide, or in a depletion mode, where charge is added or removed from the junction capacitance. The injection mode bandwidth is limited by the carrier recombination time in silicon ($\sim$1 ns); however, the recombination time impacts only the reset time of the device, while the charge injection time can be extremely fast, resulting in short computational latencies on the order of tens of picoseconds.

To confirm this, we experimentally characterized the time dynamics of the NOFU in injection mode through a "pump-probe" measurement as depicted in Figure S8a. In the experimental setup, we multiplexed a continuous-wave (CW) "probe" laser with a pulsed (16 ps) "pump" laser to drive the device. The pulsed laser, which is tuned to be significantly off-resonant with the cavity, applies short impulses to the driving photodiode. Each pulse of the laser delivers a fixed amount of charge to the photodiode in a time window much faster than the cavity dynamics, allowing characterization of the system time response without confounding effects from the rise time of the pump pulse. The CW probe laser was parked at resonance

such that each driving pulse, which injects carriers into the waveguide, detunes the resonance and produces an increase in optical transmission.

Both laser inputs were multiplexed onto a single fiber with a 2x1 splitter and coupled into a test structure of the NOFU. The pump light was filtered out of the output signal with a wavelength-division multiplexer, and the remaining probe signal was coupled into a high-speed photodiode whose output was characterized with a 40 GSa/s oscilloscope. We confirmed that the response is mediated by the injection modulator, and not any interaction of the pulse with the cavity, by shutting off the voltage bias to the NOFU. When we did so, we found no response on the optical signal. Note that while we applied a higher bias to the device here (3 V) than in our classification experiments, the power consumption of the NOFU is only marginally affected as the injection modulator still drops only $\sim$0.7 V across the device and the rest is dropped across the photodiode.

Figure S8c shows the time-domain response of the NOFU. We find that the response time of the device, i.e. the time required for charge to be injected into the device and for the cavity to detune, is $<$100 ps. The reset time, as expected, is on the order of several nanoseconds due to the carrier lifetime in the waveguide, but could potentially be reduced through the use of pre-emphasis techniques [S7].

Real-time DNN applications, such as autonomous navigation, require ultra-fast latency, i.e. the "time to solution" following an input. The NOFU is ideal for such an application, as it realizes ultra-short computational latencies.

Processing large batches of inputs, such as in a datacenter, would depend on the AC bandwidth of the NOFU. The AC bandwidth of our device, which is determined both by rise ($\sim$ 100 ps) and fall time ($\sim$ 5 ns), would be $\sim$ 150 MHz. However, some device optimization, such as reducing the modulator diameter, can improve injection modulator bandwidths to $\sim$ 1 GHz [S8], which is comparable to the clock speed of modern digital ASICs for machine learning such as the Google tensor processing unit (TPU). As our architecture is compatible with wavelength and polarization multiplexing, parallel processing of data could further push inference rates to $> 10^{10}$ inferences/s.

Reset times in depletion mode

Our classification experiments in the main text were performed in injection mode. However, we also characterized the reset time of the nonlinear unit in depletion mode. Here, we modulated an input continuous-wave laser into the NOFU using an electro-optic modulator driven by a 25 GSa/s arbitrary waveform generator (AWG).

The experimental setup is shown in Figure S8b. A 1 GHz optical square wave was applied to the device and the output signal was resolved with a photodetector monitored by a high-speed oscilloscope.

The nonlinear dynamics of the device will be observed on the rising and falling edges of the optical signal, where the optical power incident on the device is significantly changing. Figure S9 shows the relative positions of the laser wavelength and the resonance. The laser is parked at a wavelength slightly red-shifted from the "passive" position of the resonance, such that at rising edges, where optical power into the device increases, the photodiode reverse biases the junction in the ring modulator and the resonance red-shifts, producing a dip in transmission that counteracts the overall increase in optical power. On falling edges, where optical power into the device drops, the resonance blue-shfts, producing an uptick in transmission.

The time response of the device is shown in Figure S9. The NOFU was optimized to operate in carrier injection, accounting for the relatively low modulation efficiency in depletion mode. However, we were still able to observe a nonlinear response of the device at the falling edge of the optical signal, where a 50 ps response time is observed that is limited by our test equipment, which was 10 GHz bandwidth. This fall time is significantly shorter than the fall time we observed in injection mode, suggesting that optimizing future NOFU devices for depletion mode may be a route to realizing faster bandwidths not limited by carrier lifetimes.

Time-of-flight through the NOFU

Finally, we characterized the optical time-of-flight through the NOFU also using the setup depicted in Figure S8b. Here, we use a lithium niobate modulator driven by an arbitrary waveform generator to send optical pulses into the device. The time-of-flight of the system is measured using an oscilloscope and the delay through the test setup was de-embedded by repeating the same procedure for a 450 μm long waveguide on chip.

We found the delay between the two signals to be about 125 ps, corresponding to the point at which each signal reaches half of its maximum value. As the waveguide routing on chip to the NOFU test structure is about 1.4 mm longer compared to the 450 μm long waveguide, this will account for an additional 20 ps of delay assuming a group index $n_g = 4.2$. The

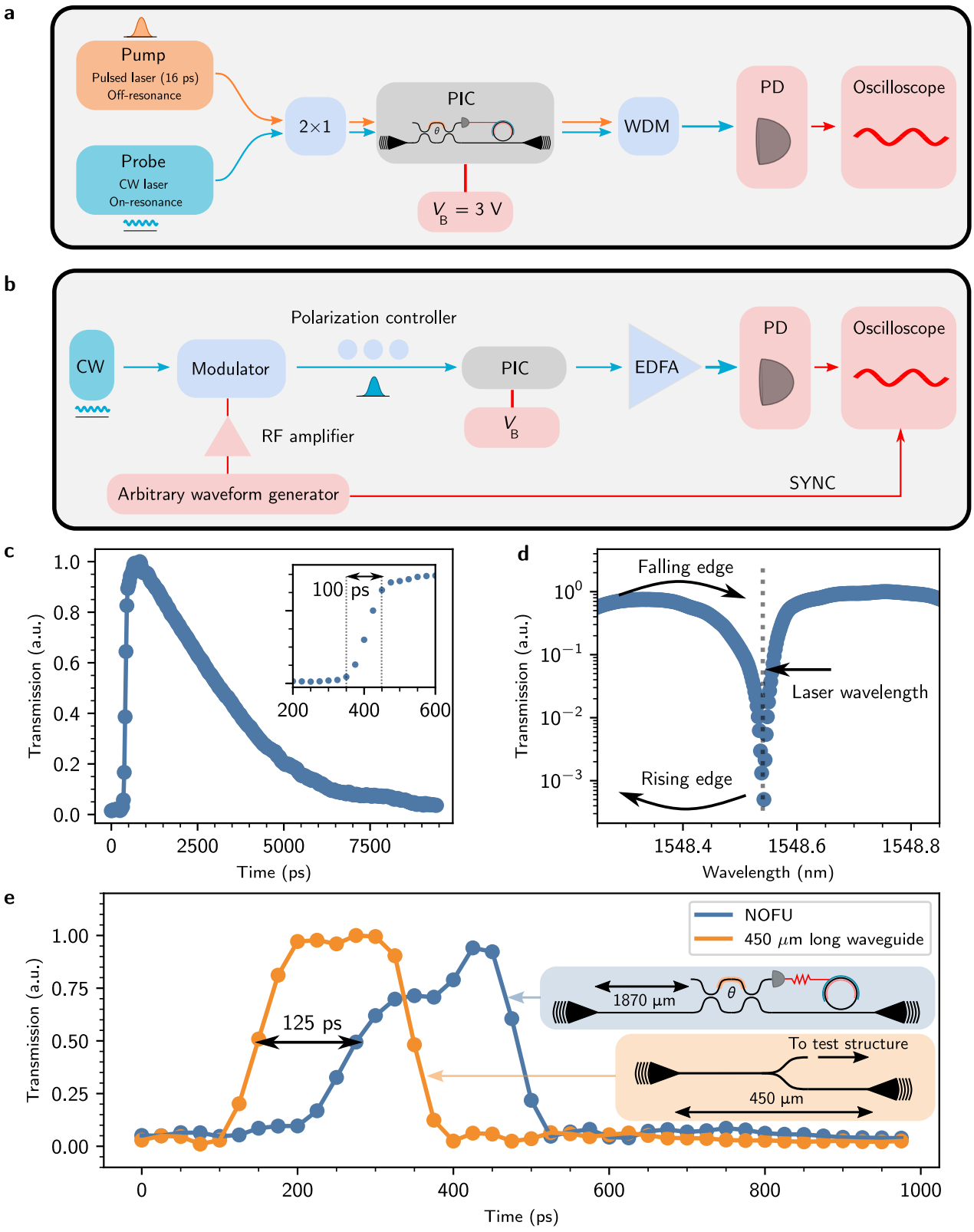

FIG. S8: a) Experimental setup for "pump-probe" measurement to characterize response time of NOFU in injection mode. b) Experimental setup for measuring the NOFU's time response in depletion mode and the time-of-flight through various system components. c) The response time of the NOFU in injection mode, corresponding to a computational latency of 100 ps. d) Relative positions of the resonance and CW laser for the data taken in part c). As pump light is injected into the system, the resonance blue-shifts, producing an increase in transmission. e) Delay of an optical pulse through a 450 μm long waveguide and the NOFU. We measured a total delay of 125 ps between the two; accounting for differences in waveguide routing between the test structures, this corresponds to a 105 ps time-of-flight through the NOFU.

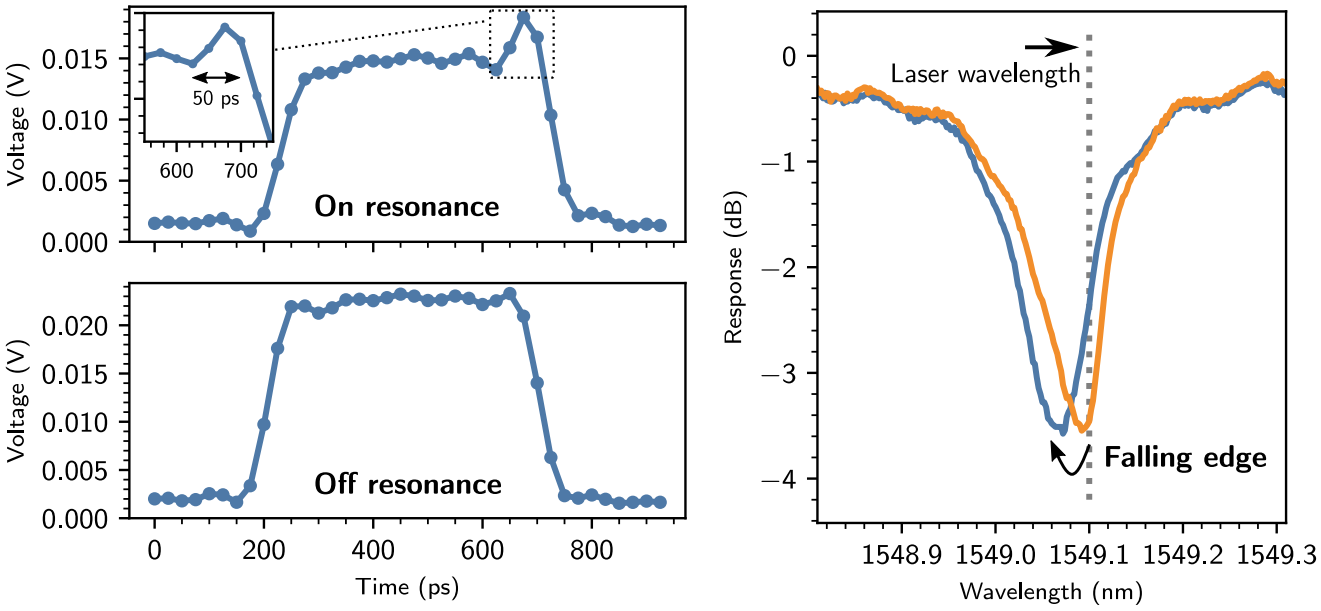

FIG. S9: Experimental measurement of NOFU response in depletion mode. To characterize the time response of the device, we input optical pulses into the device at a 1 GHz repetition rate. The laser was parked at a wavelength red-shifted from the passive resonance position; on falling edges of the optical signal, the resonance blue-shifted and an increase in transmission was observed that counteracted the envelope of the pulse. We characterized the response time of this nonlinearity to be 50 ps, which corresponds to a fall time much faster than those produced by carrier injection effects, and confirmed that this response is produced by the NOFU by detuning the input laser from the resonator, which resulted in no nonlinear response being observed.

optical time-of-flight through the device is therefore ∼105 ps, which is comparable to the electrical response time. Thus, the computational latency of the NOFU is limited by the optical time-of-flight through the device.

Classification latency

Our experimental measurements confirm that the electrical response time of the NOFU is comparable to the optical time-of-flight through the device. Thus, classification in our system is fundamentally limited by optical time-of-flight, which is an advantage of a coherent optical architecture such as the FICONN.

We experimentally characterized the time-of-flight limited latency of our system using a setup similar to that depicted in Figure S8b; the one difference for this measurement was that we used the photodiodes in the coherent receiver on chip, rather than an external photodiode.

As discussed in the main text, the circuit has two paths that effectively form the two arms of an interferometer–a "signal" arm, which performs classification, and a "local oscillator" arm that is split off at the input of the chip and routed to the coherent receiver. The local oscillator arm is 1 cm long and was mostly routed with a rib (partially-etched) waveguide, which corresponds to about $\tau_{\text{LO}} = 130$ ps propagation time assuming a group index $n_g = 3.8$. The signal arm, on the other hand, is much longer. We can thus estimate the latency by measuring the delay between the two arms $\tau_{\text{latency}} - \tau_{\text{LO}}$.

This delay was characterized using a "pump-probe" measurement in which we input two 40 ps duration optical pulses into the chip with a programmable delay $\tau_{\text{pulse delay}}$. When a single pulse enters the chip, it splits offs into the signal and local oscillator paths and will arrive at different times at the coherent receiver due to the time delay $\tau_{\text{latency}} - \tau_{\text{LO}}$ between paths. Thus, modulating the phase difference between the local oscillator and signal paths will not produce any interference at the coherent receiver.

However, when inputting a second pulse into the chip, the delay can be tuned such that the first pulse that traveled down the (longer) signal path arrives simultaneously at the receiver as the second pulse that traveled down the (shorter) local oscillator path. In this case, the two pulses will interfere at the receiver and modulation will be observed. This interference will only be observed, however, if the delay between the first and second pulses corresponds to the delay between the two arms of the chip, i.e. if $\tau_{\text{pulse delay}} = \tau_{\text{latency}} - \tau_{\text{LO}}$.

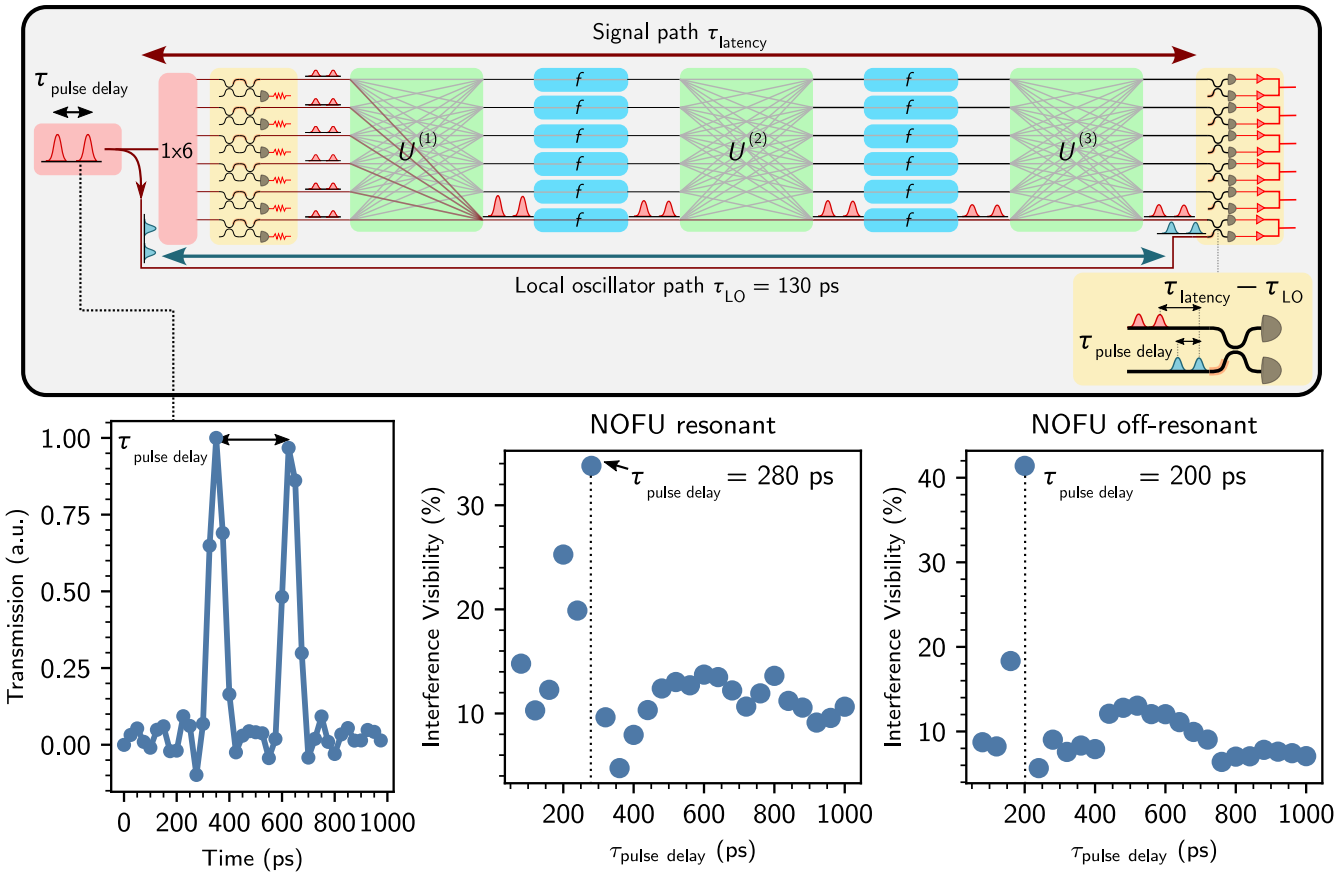

FIG. S10: Top: Characterization of the optical propagation delay. We use a "pump-probe" measurement that characterizes interference at the coherent receiver between two time-delayed pulses. Each pulse is split between the (shorter) local oscillator path and the (longer) signal path, which performs classification. Interference will only be observed when the first pulse, traveling down the longer signal path, overlaps in time at the receiver with the second pulse traveling down the shorter local oscillator path. We input optical pulses into the chip and program their relative delay using the setup shown in Figure S8b. Left: Time trace of two optical pulses input into the chip. Middle: Interference visibility as a function of pulse delay. We observe an increase in visibility at a 280 ps delay, corresponding to the time delay between the two arms of the chip. Some residual interference is observed at shorter pulse delays due to distortion of the pulse by the NOFU. Right: Interference visibility when the NOFU is off-resonant. The latency decreases by 80 ps, as there is no interaction with the cavity, and we no longer observe interference at shorter pulse delays.

Specifically, when we input an optical pulse $E(t) + E(t - \tau_{\text{pulse delay}})$ into the PIC and split them between the two paths, the optical field at the coherent receiver is:

$$E_{\text{RX}} = \frac{1}{\sqrt{2}}\left[\gamma_{\text{LO}}\alpha e^{i\phi}[E(t-\tau_{\text{LO}}) + E(t-(\tau_{\text{LO}}+\tau_{\text{pulse delay}}))] + \gamma_{\text{signal}}\sqrt{1-\alpha}[E(t-\tau_{\text{latency}}) + E(t-(\tau_{\text{latency}}+\tau_{\text{pulse delay}}))]\right] \tag{S18}$$

where ϕ is the programmable phase shift on the LO path, α is the (amplitude) fraction of light split to the LO path, γ_{LO} is the amplitude transmission of the LO path, and γ_{signal} is the amplitude transmission of the signal (classification) path. As the LO splitter is a programmable MZI, we can tune α such that $\gamma_{\text{LO}}\alpha = \gamma_{\text{signal}}\sqrt{1-\alpha}$. In that case:

$$E_{\text{RX}} \propto e^{i\phi}[E(t-\tau_{\text{LO}}) + E(t-(\tau_{\text{LO}}+\tau_{\text{pulse delay}}))] + E(t-\tau_{\text{latency}}) + E(t-(\tau_{\text{latency}}+\tau_{\text{pulse delay}})) \tag{S19}$$

We assume that: 1) our pulse is Gaussian, i.e. $E(t) = \exp(-t^2/(2\sigma^2))$; 2) the pulse width σ is short relative to the circuit delays, i.e. $\sigma \ll \tau_{\text{latency}}, \tau_{\text{LO}}$; and 3) power is integrated at the receiver for a time window t_{int} much longer than the circuit delay, i.e. $t_{\text{int}} \gg \tau_{\text{latency}}$. The total power integrated at the receiver as a function of ϕ will be:

$$\int_{t_{\text{int}}} |E_{\text{RX}}|^2 \, dt = 2\sigma\sqrt{\pi}\left(2 + \exp\left[-\frac{(\tau_{\text{pulse delay}} - (\tau_{\text{latency}} - \tau_{\text{LO}}))^2}{4\sigma^2}\right]\cos\phi\right) \tag{S20}$$

Parameter	Value	Reference
Digital-to-analog conversion (transmitter, 1 GHz)	6.1 mW	[S9]
Digital-to-analog conversion (transmitter, 14 GHz)	50 mW	[S10]
Digital-to-analog conversion (weights)	5 μW	[S11]
Resonant modulator (transmitter)	0.9 fJ/bit at 25 Gb/s $\rightarrow$ 22.5 μW	[S12]
Phase shifter (weights, thermal)	18.75 mW	Measured
Phase shifter (weights, MEMS)	75 μW	[S13]
NOFU (injection mode)	60 μW	Measured
Transimpedance amplifier (receiver, 1.6 GHz)	2.85 mW	[S14]
Transimpedance amplifier (receiver, 20 GHz)	57 mW	[S15]
Analog-to-digital conversion (receiver, 1 GHz)	2.55 mW	[S16]
Analog-to-digital conversion (receiver, 8.8 GHz)	35 mW	[S17]

TABLE S1: Parameters used for energy efficiency calculation.

If we vary ϕ, the integrated power will vary with a modulation amplitude $2\sigma\sqrt{\pi}\exp\left[-\frac{(\tau_{\text{pulse delay}}-(\tau_{\text{latency}}-\tau_{\text{LO}}))^2}{4\sigma^2}\right]$. Thus, by tuning $\tau_{\text{pulse delay}}$ to maximize the interference visibility, which occurs when $\tau_{\text{pulse delay}} = \tau_{\text{latency}} - \tau_{\text{LO}}$, we can characterize the on-chip propagation delay.

Our measurement setup is shown in the top of Figure S10. We generated the optical pulses and programmed $\tau_{\text{pulse delay}}$ using a lithium niobate modulator that we drive with a 25 GSa/s arbitrary waveform generator (AWG). Our measurement has a resolution limit of 40 ps, which is limited by the sampling rate of the AWG. In the bottom of Figure S10, we plot the interference visibility $V = (P_{\text{max}} - P_{\text{min}})/(P_{\text{max}} + P_{\text{min}})$ observed as a function of $\tau_{\text{pulse delay}}$.

We observe a peak in interference visibility between the local oscillator and signal arms at a time delay of 280 ps, corresponding to the time delay in time-of-flight between the two paths. Moreover, at delays longer than this, interference is no longer observed as the second pulse traveling down the (short) local oscillator path arrives after the first pulse traveling down the (long) signal path. We also observed some residual interference visibility of 5-10% at even very long pulse delays. This is not true interference, but is a result of amplitude modulation produced by the thermal phase shifter. To confirm this, we measured the visibility of "interference" when light is distributed to only the local oscillator path and also observed about 10% visibility.

We also measured interference occurring at shorter time delays than 280 ps, which occurs due to distortion of the pulse by the ring resonator in the NOFU. To confirm this, we detuned the NOFU resonance and characterized the latency once again. We found that this decreased the propagation delay to 200 ps and that the visibility at shorter time delays was no longer present.

We therefore find a total latency of 410 ps, accounting for the fact that the local oscillator arm has a delay of 130 ps. Due to chip area constraints, we connected different subsystems with U-turns, as shown in Figure 2a of the main text. These routing waveguides account for an additional 7.5 mm ($\sim$100 ps, assuming a group index $n_g = 4.2$) of propagation. Thus, an optimized layout for our system could achieve even shorter latencies.

Energy efficiency

The FICONN architecture with N modes and M layers performs $2MN^2 + 2(M-1)N$ operations per inference, where the first term accounts for linear matrix operations and the second term refers to the nonlinear activation function. For large N the first term dominates and we approximate the total number of operations as $2MN^2$.

The total energy consumption per operation of the system for a single inference can therefore be approximated as $\tau_{\text{latency}}P_{\text{total}}/(2MN^2)$, where P_{total} is the total power consumption of the photonics, drivers, and readout electronics and τ_{latency} is the time required for a single inference. The system requires MN^2 phase shifters, N transmitters, N receivers, and MN nonlinear optical function units, making the total power consumption $P_{\text{total}} = MN^2P_{\text{PS}} + MNP_{\text{NOFU}} + N(P_{\text{TX}} + P_{\text{ICR}})$. Substituting this into the expression for energy consumption per operation produces equation 2 in the main text.

Our device performs $2MN^2 + 2(M-1)N = 240$ operations per inference, where $M = 3$ and $N = 6$. The phase shifters require about 25 mW per π phase shift; as the internal phase shifters only require up to π phase shift, while the external phase shifters require up to 2π, we assume the average power consumption per phase shifter is 18.75 mW. The phase shifter contribution to the energy per operation is therefore $144 \times (18.75\text{ mW}) \times (410\text{ ps})/(240\text{ OPs}) = 4.6$ pJ/OP, where we include phase shifters for both model parameters and the transmitter. This energy requirement would reduce substantially with the use of undercut thermal phase shifters [S18], which reduce power dissipation by an order of magnitude, or MEMS-actuated devices [S13, S19], both of which are available in silicon photonic foundries. The nonlinear optical function unit consumes 60 μW of power, which contributes about $12 \times (60\ \mu\text{W}) \times (410\text{ ps})/(240\text{ OPs}) = 1.2$ fJ/OP to this total. The total on-chip energy consumption is therefore 2.7 W. For our experiments, we used a 20 mW tunable telecom laser; assuming a 20% wall plug efficiency, this consumes an additional 0.1 W. The chip temperature is stabilized using a Peltier unit that consumes approximately 0.5 W.

We used a benchtop voltage source (Qontrol Systems Q8iv), which consumes about 40 W, in our demonstration to control the devices on chip. This energy consumption could be significantly reduced by using discrete package DACs for the phase shifters encoding the weights, such as the Analog Devices AD8802 [S11], which consumes about 5 μW per channel. In that case, the electronics for the weight values would consume less than 1 mW and our total transmitter and readout electronics (which include DACs/TIAs/ADCs) would consume 2.1 W, producing a total power consumption of our experiment of 5.4 W.

How would a system with fully-integrated electronics perform?

Ideally, the devices in our chip would be controlled by a custom CMOS integrated circuit and use high-speed transmitters at the input. To estimate the energy contribution of an electronic driver and readout circuitry in such a system, we use values reported in the literature for DACs, TIAs, and ADCs, which are shown in Table S1. These literature values were sourced from different process nodes (for example, ref. [S15] was fabricated in 0.25 μm CMOS, while ref. [S17] was fabricated in 32 nm CMOS), but could in principle be manufactured together in a sufficiently advanced node with high-resolution lithography.

The transmitter would require high-speed DACs for programming input vectors into the system, while the receiver would require high-speed TIAs and ADCs for readout. In order to ensure electrical-to-optical conversion time does not excessively increase latency, we assume the electronics operate at a bandwidth of ~ 10 GHz. The model parameters can be controlled by lower-speed electronics, as they are not anticipated to change frequently; even when training the system *in situ*, the parameters will only be updated after an entire batch is evaluated.

Using these values, we find the worst-case energy consumption of electronics for our device would be $[12 \times (50\text{ mW} + 57\text{ mW} + 35\text{ mW}) + 132 \times (5\ \mu\text{W})] \times (410\text{ ps})/(240\text{ OPs}) = 2.9$ pJ/OP. As the on-chip phase shifters contribute 4.2 pJ/OP (now excluding the thermal phase shifters for the transmitter, which would instead be high-speed electro-optical phase shifters [S12] consuming 22.5 μW per channel) and the laser and Peltier unit contribute $0.6\text{ W} \times (410\text{ ps})/(240\text{ OPs}) = 1.0$ pJ/OP, the total system performance for a single inference with low-latency electronics would be 8.1 pJ/OP.

If we include the electrical-to-optical conversion time (assuming a response time of $\approx 0.35/f_{\text{BW}}$), this would contribute an additional ~ 100 ps latency and the energy/OP will increase to 10.0 pJ/OP.

Scaling

The latency and energy efficiency of the FICONN would be further improved by performing inference on large batches of vectors, as they can be transmitted into the PIC at a faster rate than the end-to-end propagation delay. For W input vectors, the total latency of the system is $\tau_{\text{latency}} + (W-1)/f_{\text{BW}}$, where f_{BW} is the total bandwidth of the system. For large W, the latter term dominates the latency. The system then performs $2f_{\text{BW}}(MN^2 + (M-1)N)$ TOPS and the energy efficiency becomes:

$$E_{\text{OP}} \approx \frac{1}{2f_{\text{BW}}}\left[P_{\text{PS}} + \frac{P_{\text{NOFU}}}{N} + \frac{P_{\text{TX}} + P_{\text{ICR}}}{MN}\right] \tag{S21}$$

The ultimate speed and energy efficiency of our architecture is therefore determined by the rate at which input vectors can be transmitted into the DNN, rather than the end-to-end propagation time. Resonant, high-speed modulators in silicon photonics have previously been shown to consume less than 1 fJ/bit [S12]. In our implementation the NOFU is realized with a carrier injection device, which would usually be limited to about $\sim$1 ns response time due to the relatively long

N	Phase shifter	E_{OP}	$E_{total,projected}$	$\tau_{latency}$	TOPS
6	Undercut thermal [S18]	28.9 fJ/OP	317.2 fJ/OP	151 ps	1.9
6	MEMS [S13, S19]	5.7 fJ/OP	294 fJ/OP	151 ps	1.9
64	MEMS [S13, S19]	2.7 fJ/OP	33.2 fJ/OP	1.4 ns	199
128	MEMS [S13, S19]	2.5 fJ/OP	18.3 fJ/OP	2.8 ns	791
256	MEMS [S13, S19]	2.4 fJ/OP	10.8 fJ/OP	5.5 ns	3154

TABLE S2: Predicted performance metrics for a three-layer FICONN with N neurons. We list the on-chip energy consumption E_{OP}, as well as a projection of the total power dissipation $E_{total,est}$ including integrated driver electronics. The predicted metrics assume wavelength-multiplexed inference with 16 channels at a rate of 1 GHz/λ. For latency, we assume a device length of 500 μm and an optimized layout (i.e. no U-turns).

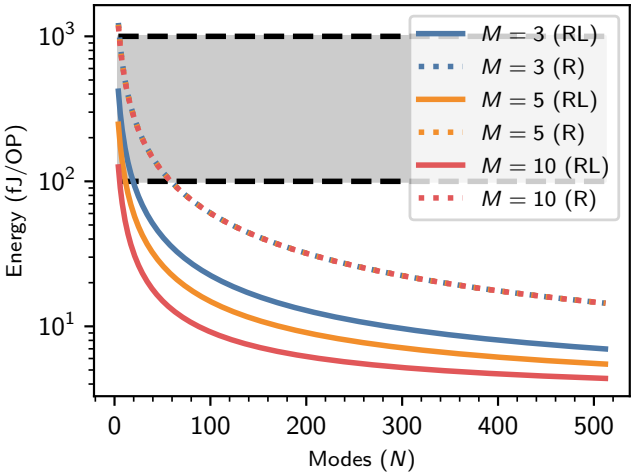

FIG. S11: The energy efficiency per operation of a photonic DNN as a function of number of modes (N), number of layers (M), and whether the architecture is receiverless (RL) or requires readout between each intermediate layer (R). The shaded region indicates the energy efficiencies of current-day electronic ASICs for DNNs ($\sim$0.1$-$1 pJ/OP). We assume a clock rate of 1 GHz/λ with 16 wavelengths.

recombination time in silicon. Optimized injection modulators could realize clock speeds of $>$ 1 GHz [S8], which could potentially enable inference clock rates of $>$ 10 GHz using wavelength multiplexing.

In Table S2, we report the predicted performance metrics of a three-layer FICONN with N neurons assuming an inference clock speed of 1 GHz with 16 wavelength channels (integrated silicon photonic transmitters with as many as 32 channels have been previously reported [S20]). This calculation assumes that while the CMXU can perform broadband matrix-vector multiplication across multiple wavelength channels, each wavelength channel will require a separate transmitter/receiver/NOFU. As the NOFU makes use of resonant modulators, which naturally multiplex and demultiplex wavelengths, a wavelength-multiplexed nonlinear optical function unit could be constructed by employing a ring resonator to tap off a fraction of the incident signal in a targeted wavelength channel to a photodiode and use it to drive a second ring that modulates only that wavelength. Thus, each wavelength channel would require only two rings and a photodiode; as microdisk modulators with 2.4 μm radius have been previously demonstrated [S12], this structure could be relatively compact on chip. For a wavelength multiplexed system with K channels, the energy scaling in equation S21 will be:

$$E_{OP} \approx \frac{1}{2Kf_{BW}}\left[P_{PS} + \frac{KP_{NOFU}}{N} + \frac{KP_{TX} + KP_{ICR}}{MN}\right] \tag{S22}$$

Performance improves for larger system sizes, as the energy cost of the electronics and nonlinearity is amortized by the number of modes N. Figure S11 shows the expected energy consumption of a coherent optical DNN, like our system, and one that reads out optical signals between layers as a function of N and M. We show also the current performance region of application-specific integrated circuits for DNNs, which consume approximately 0.1$-$1 pJ per operation [S21, S22]. For

a receiverless, three layer system such as ours, a circuit with $N = 19$ modes is sufficient to achieve energy efficiencies below 100 fJ/OP, while $N \sim 300$ attains efficiencies better than 10 fJ/OP. We assumed a clock rate of 1 GHz/λ with 16 wavelengths in Figure S11.

Direct processing of optical data would further lower energy consumption, as high-speed DACs [S23] are no longer required to encode data onto an optical carrier. High speed readout and digitization of the outputs [S24, S25], however, is still required. This accounts for the comparatively poorer scaling of energy efficiency for devices that read out between layers. A three layer system that performs intermediate readout and re-encoding between layers would need to have nearly three times as many modes as a receiverless circuit to achieve energy efficiencies better than 100 fJ/OP. Moreover, while a receiverless system with $M = 10, N = 90$ could attain efficiencies of 10 fJ/OP, a system with intermediate readout and re-encoding would require $N > 500$ to achieve the same efficiency.

VIII. PERFORMANCE SUMMARY

TABLE S3: Comparison table to state-of-the-art integrated photonic and electronic systems.

	This work	Ref. [S26]	Ref. [S27]	Ref. [S28]	Ref. [S29, S30] (Google Edge TPU)
Single chip?	Yes	No	No	No	Yes
Implements	End-to-end DNN	Linear algebra	Linear algebra	End-to-end DNN	Linear algebra
Inference platform	Optical	Optoelectronic	Optoelectronic	Optoelectronic	Electronic
Training platform	*In situ*	Digital	Digital	Digital	Digital
Power consumption	~45 W*	N/A	10 W**	3.75 W	2 W
Latency	410 ps	N/A	N/A	570 ps	$\sim$ 1 ms
End-to-end TOPS	0.59	N/A	N/A	0.27	N/A
End-to-end computation density	0.02 TOPS/mm^2	N/A	N/A	N/A	N/A

*Using benchtop current source. $\sim$ 5 W if using discrete DACs.
**Computed using the reported energy efficiency of 0.4 TOPS/W. Does not include power consumption of fully-connected layer implemented on GPU.

IX. EXPERIMENTAL SETUP

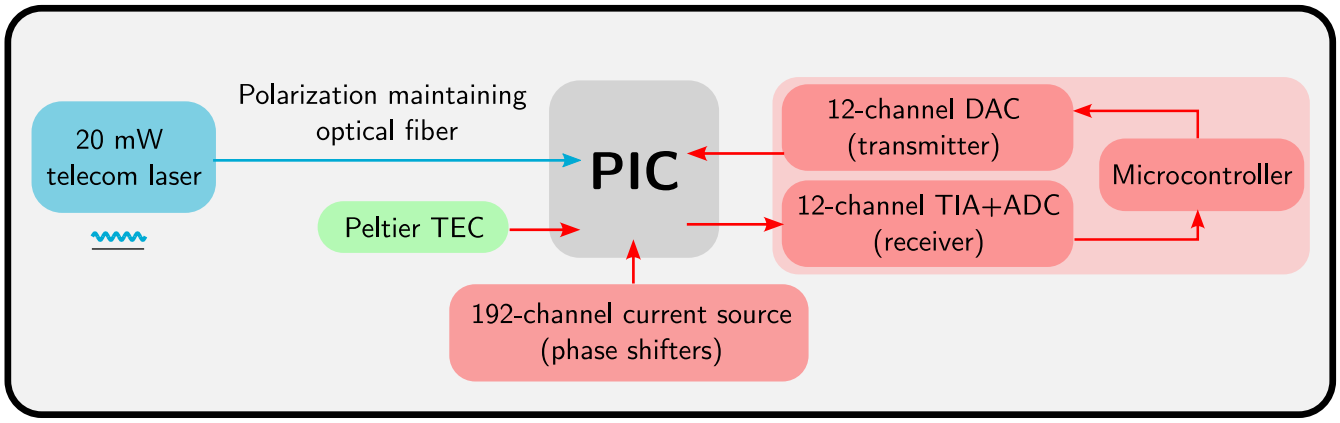

FIG. S12: Schematic of experimental setup. Light is coupled into the PIC from a 20 mW tunable laser (Santec TSL-710) using a polarization-maintaining fiber array. Weights on chip are programmed with a 192-channel current source from Qontrol Systems (Q8iv), while vector input and readout is performed with a custom-designed driver system that integrates 16-bit DACs, low-noise transimpedance amplifers, and 18-bit ADCs onto a single board operated by a microcontroller. The chip temperature is stabilized using a Peltier module.

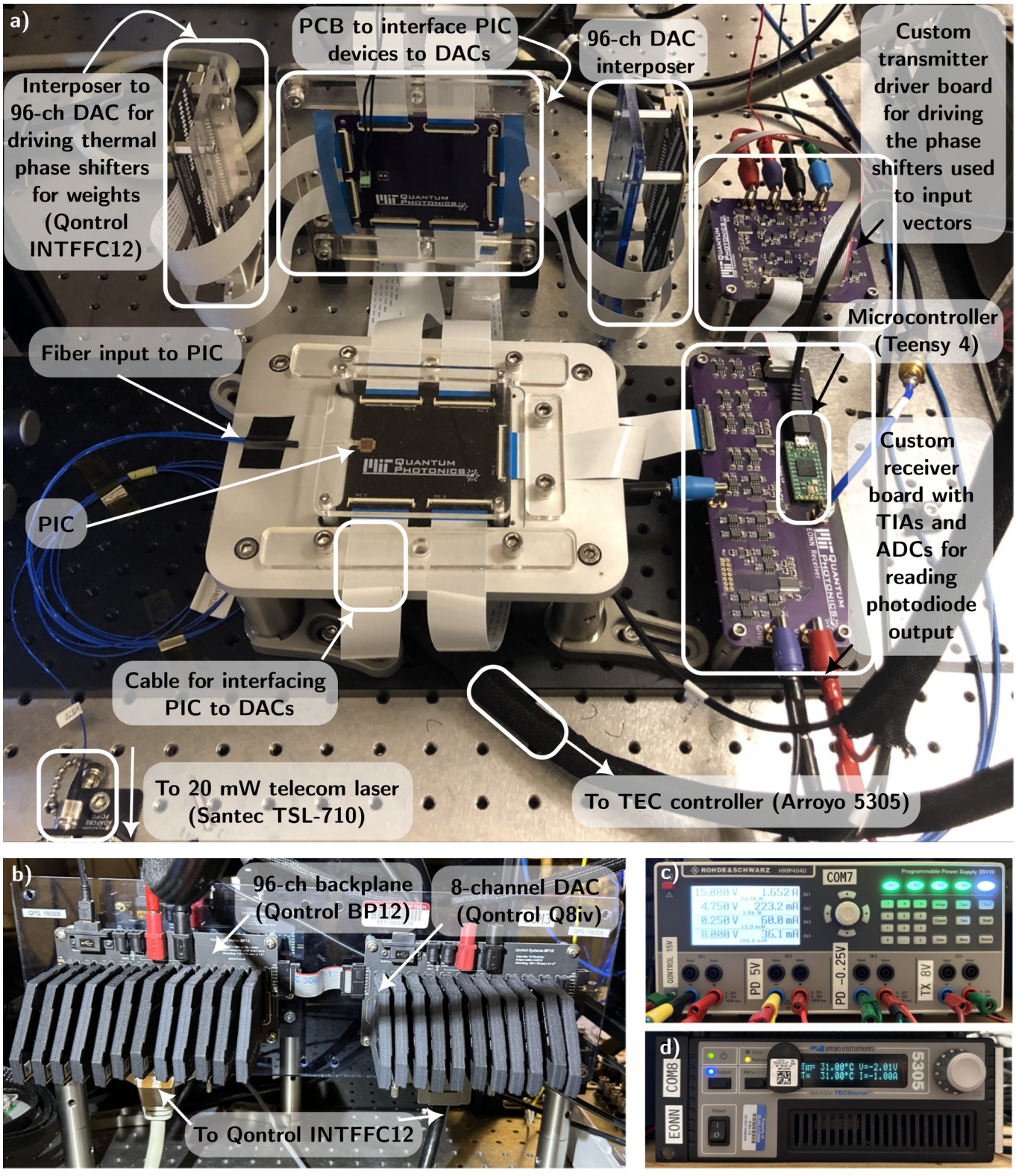

FIG. S13: Photograph of experimental setup. a) Chip testing setup. The PIC is wirebonded to a PCB for electrical control and is fiber attached for simplified interfacing to the laser. Electrical signals are controlled through a commercial DAC system for the weights and a custom transmitter board for the inputs. Outputs are read out with a custom receiver board. b) Commercial current-driver system (Qontrol Q8iv) for controlling PIC. c) Benchtop power source (HMP4040) used to supply power to the driver electronics. d) Commercial TEC controller (Arroyo 5305) used to stabilize PIC temperature.

[S1] Prabhu, M. *et al.* Accelerating recurrent Ising machines in photonic integrated circuits. *Optica* **7**, 551 (2020).

[S2] Bandyopadhyay, S., Hamerly, R. & Englund, D. Hardware error correction for programmable photonics. *Optica* **8**, 1247–1255 (2021).

[S3] Hamerly, R., Bandyopadhyay, S. & Englund, D. Stability of Self-Configuring Large Multiport Interferometers. *Phys. Rev. Applied* **18**, 024018 (2022).

[S4] Cauwenberghs, G. A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization. In Hanson, S., Cowan, J. & Giles, C. (eds.) *Advances in Neural Information Processing Systems*, vol. 5 (Morgan-Kaufmann, 1992).

[S5] Williamson, I. A. D. *et al.* Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks. *IEEE Journal of Selected Topics in Quantum Electronics* **26**, 1–12 (2020).

[S6] McCaughan, A. N. *et al.* Multiplexed gradient descent: Fast online training of modern datasets on hardware neural networks without backpropagation. *APL Machine Learning* **1**, 026118 (2023).

[S7] Xu, Q., Manipatruni, S., Schmidt, B., Shakya, J. & Lipson, M. 12.5 Gbit/s carrier-injection-based silicon micro-ring silicon modulators. *Optics Express* **15**, 430 (2007).

[S8] Wu, R. *et al.* Compact models for carrier-injection silicon microring modulators. *Optics Express* **23**, 15545 (2015).

[S9] Kim, J. *et al.* A 12-b, 1-GS/s 6.1 mW current-steering DAC in 14 nm FinFET with 80 dB SFDR for 2G/3G/4G cellular application. In *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 248–251 (2017).

[S10] Caragiulo, P., Mattia, O. E., Arbabian, A. & Murmann, B. A Compact 14 GS/s 8-Bit Switched-Capacitor DAC in 16 nm FinFET CMOS. In *2020 IEEE Symposium on VLSI Circuits*, 1–2 (2020).

[S11] AD8802 Datasheet (2022). URL `https://www.analog.com/media/en/technical-documentation/data-sheets/AD8802_8804.pdf`.

[S12] Timurdogan, E. *et al.* An ultralow power athermal silicon modulator. *Nature Communications* **5**, 4008 (2014).

[S13] Gyger, S. *et al.* Reconfigurable photonics with on-chip single-photon detectors. *Nature Communications* **12**, 1408 (2021).

[S14] Abd-elrahman, D., Atef, M., Abbas, M. & Abdelgawad, M. Low power transimpedance amplifier using current reuse with dual feedback. In *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 244–247 (2015).

[S15] Sedighi, B. & Scheytt, J. C. Low-Power SiGe BiCMOS Transimpedance Amplifier for 25-GBaud Optical Links. *IEEE Transactions on Circuits and Systems II: Express Briefs* **59**, 461–465 (2012).

[S16] Oh, D.-R. *et al.* An 8b 1GS/s 2.55mW SAR-Flash ADC with Complementary Dynamic Amplifiers. In *2020 IEEE Symposium on VLSI Circuits*, 1–2 (2020).

[S17] Kull, L. *et al.* A 35mW 8b 8.8 GS/s SAR ADC with low-power capacitive reference buffers in 32nm Digital SOI CMOS. In *2013 Symposium on VLSI Circuits*, C260–C261 (2013).

[S18] Dong, P. *et al.* Thermally tunable silicon racetrack resonators with ultralow tuning power. *Optics Express* **18**, 20298 (2010).

[S19] Baghdadi, R. *et al.* Dual slot-mode NOEM phase shifter. *Optics Express* **29**, 19113 (2021).

[S20] Rizzo, A. *et al.* Massively scalable Kerr comb-driven silicon photonic link. *Nature Photonics* **17**, 781–790 (2023).

[S21] Jouppi, N. P. *et al.* In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ISCA '17, 1–12 (Association for Computing Machinery, New York, NY, USA, 2017).

[S22] Shao, Y. S. *et al.* Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 14–27 (ACM, 2019).

[S23] Greshishchev, Y. *et al.* A 60 GS/s 8-b DAC with gt; 29.5dB SINAD up to Nyquist frequency in 7nm FinFET CMOS. In *2019 IEEE BiCMOS and Compound semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, 1–4 (2019).

[S24] Ahmed, M. G. *et al.* A 34Gbaud Linear Transimpedance Amplifier with Automatic Gain Control for 200Gb/s DP-16QAM Optical Coherent Receivers. In *2018 Optical Fiber Communications Conference and Exposition (OFC)* (2018).

[S25] Pfaff, D. *et al.* A 56Gb/s Long Reach Fully Adaptive Wireline PAM-4 Transceiver in 7nm FinFET. In *2019 Symposium on VLSI Circuits*, C270–C271 (2019).

[S26] Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11**, 441–446 (2017).

[S27] Feldmann, J. *et al.* Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58 (2021).

[S28] Ashtiani, F., Geers, A. J. & Aflatouni, F. An on-chip photonic deep neural network for image classification. *Nature* **606**, 501–506 (2022).

[S29] Seshadri, K., Akin, B., Laudon, J., Narayanaswami, R. & Yazdanbakhsh, A. An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks. In *2022 IEEE International Symposium on Workload Characterization (IISWC)*, 79–91 (2022).

[S30] Sun, Y. & Kist, A. M. Deep Learning on Edge TPUs. *arXiv:2108.13732* (2021).