Supplementary Materials: Training neural networks with end-to-end optical backpropagation

James Spall,[1,2] Xianxin Guo,[1,2,*] and A. I. Lvovsky[1,2,†]

[1] *Clarendon Laboratory, University of Oxford, Parks Road, Oxford, OX1 3PU, UK*

[2] *Lumai Ltd, Wood Centre for Innovation, Quarry Road, Headington, Oxford, OX3 8SB, UK*

(Dated: December 9, 2024)

CONTENTS

* xianxin.guo@lumai.co.uk

† alex.lvovsky@physics.ox.ac.uk

Supplementary Note 1. THEORY OF NN TRAINING BY BACKPROPAGATION

Here we reproduce the salient equations of the backpropagation training method for reference; a full explanation can be found in e.g. [11]. We index the neuron vector at each layer with i, j and k respectively. We encode an input x_i, weight matrix $W^{(1)}_{ji}$ and bias $b^{(1)}_j$ in the first layer, and weight matrix $W^{(2)}_{kj}$ in the second layer. MVM-1, hidden layer activation and MVM-2 are then described as follows:

$$z^{(1)}_j = \sum_{i=1}^{N} W^{(1)}_{ji} x_i + b^{(1)}_j; \tag{S1}$$

$$a^{(1)}_j = g\left(z^{(1)}_j\right); \tag{S2}$$

$$z^{(2)}_k = \sum_{j=1}^{M} W^{(2)}_{jk} a^{(1)}_j. \tag{S3}$$

Note that the bias is applied only in MVM-1 but not in MVM-2.

We consider two types of loss function: categorical cross-entropy (CCE)

$$\mathcal{L} = -\sum_k t_k \log\left(a^{(2)}_k\right) \tag{S4}$$

and mean-squared error (MSE)

$$\mathcal{L} = \frac{1}{2}\sum_k \left|a^{(2)}_k - t_k\right|^2 \tag{S5}$$

where t_k is the one-hot label $(0, 1)$ or $(1, 0)$. In the case of CCE, the second-layer activation $a^{(2)}_k$ is found by applying the softmax activation function to $z^{(2)}_k$:

$$a^{(2)}_k = \frac{e^{z^{(2)}_k}}{\sum_k e^{z^{(2)}_k}} \tag{S6}$$

For MSE, there is no second-layer activation: $a^{(2)}_k = z^{(2)}_k$.

In both cases, the gradients of the loss function can be calculated to be

$$\frac{\partial \mathcal{L}}{\partial W^{(2)}_{kj}} = \delta^{(2)}_k \cdot a^{(1)}_j, \tag{S7}$$

$$\frac{\partial \mathcal{L}}{\partial W^{(1)}_{ji}} = \delta^{(1)}_j \cdot x_i \tag{S8}$$

and

$$\frac{\partial \mathcal{L}}{\partial b^{(1)}_j} = \delta^{(1)}_j, \tag{S9}$$

where

$$\delta^{(2)}_k = \left(a^{(2)}_k - t_k\right) \tag{S10}$$

and

$$\delta^{(1)}_j = \rho^{(1)}_j \cdot g'\left(z^{(1)}_j\right), \tag{S11}$$

with

$$\rho_j^{(1)} = \left(\sum_{k=1}^{L} w_{kj}^{(2)} \delta_k^{(2)} \right). \tag{S12}$$

We can therefore directly calculate the second weight matrix update, but to calculate the updates to the first layer, we use our optical backpropagation scheme to optically perform the calculation (S11).

Supplementary Note 2. EXPERIMENTAL SETUP

Fig. S1 presents the full diagram of the experimental setup, whose simplified version can be found in Fig. 1 of the main text. Fig. S3 and Fig. S4 show the beam paths and field patterns for the forward and backward propagating beams respectively.

Each fan-in and fan-out process is performed by a set of three cylindrical lenses, equally spaced at $f = 150$ mm between the input and output planes. The central lens, of focal length $2f = 300$ mm, performs the Fourier transform of the field in the input plane into the output plane with respect to one of the transverse coordinates. The other two cylindrical lenses, oriented at 90° with respect to the central one, have the focal length $f = 150$ mm and implement $4f$ imaging of the input plane to the output plane with respect to the other transverse coordinate, preventing unwanted diffraction and maintaining the correct phase and amplitude profile. The sets CL-1 and CL-2a perform a Fourier transform of the horizontal dimension while imaging the vertical dimension. The set CL-2b does the opposite. Note that because the ONN input in both directions is supplied in the 'fanned-out' state, no Fourier transformation is required between the forward input and SLM-1 as well as between the backward input and SLM-2; the inputs are simply imaged onto the SLMs.

Our ONN uses a coherent beam to encode real-valued elements. Below we describe how we encode and detect real-valued numbers.

The forward ONN input contains the test set element coordinates, which are always positive and can hence be encoded by the DMD. On the other hand, the errors that constitute the backward ONN input can take negative values. We encode their absolute value by means of the DMD, and subsequently apply a π phase shift to encode negative numbers by reflecting the beam from a dedicated section of SLM-1. Fig. S2(b) depicts the patterns generated on the DMD and SLM that encode the input vector and weight matrix in the forward beam, and real-valued error vectors in the backward beam. Both DMD and SLMs in our setup are reflective, but for illustrative purpose we assume they are transmssive in our schematics.

The weight matrices are encoded by means of a phase grating on the phase-only LC-SLMs. By spatially varying the offset and height of the grating, we can control both the amplitude and phase of each weight matrix element. The matrix-vector dot products are then diffracted into the first diffraction order, which is isolated by a spatial filter after reflection from the LC-SLM [12, 29].

We measure the sign of the fields in the two layers by means of homodyne detection, similar to our previous work [12]. Dedicated regions on the DMD create uniform reference beams, labeled R1, R2 and R3, which pass through the system along with the forward and backward MVM signals. The relative size and position of these reference encoding regions are indicated in Fig. S2(b).

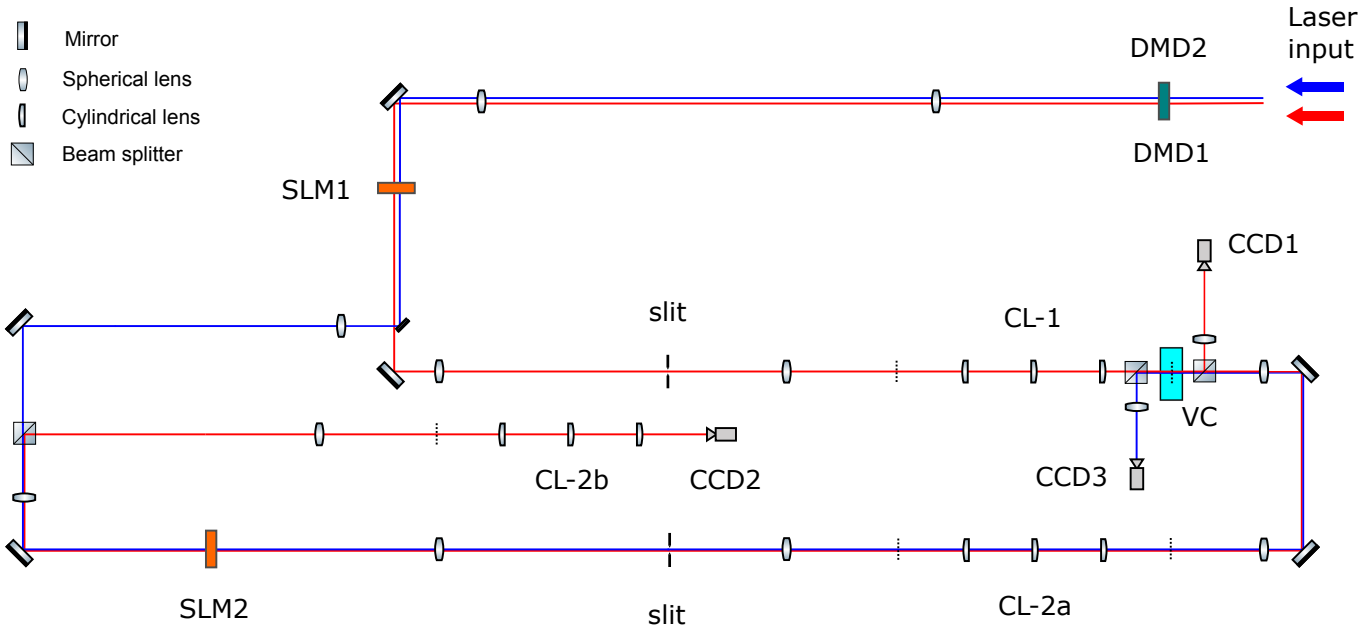

FIG. S1. **Experiment schematic.** The forward beam is shown in red, and backward beam in blue. Two pixel regions of the same DMD are used for the input of both the forward and backward paths. CL-1, CL-2a and CL-2b are the three cylindrical lenses discussed in the main text. VC: vapor cell, CCD: digital camera.

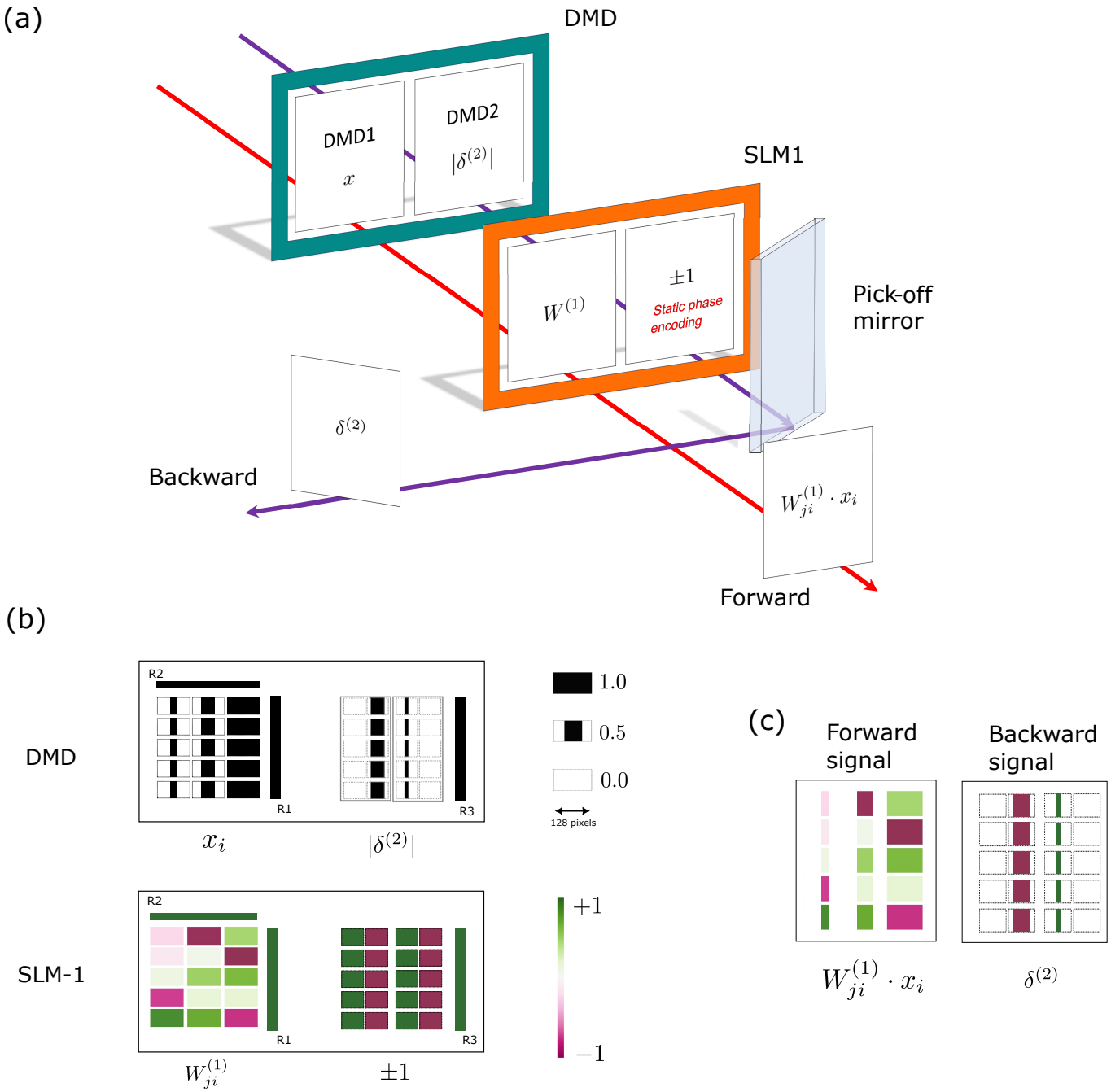

FIG. S2. **Experimental setup for encoding both forward and backward propagating beams.** **(a)** Illustration of using two halves of DMD and SLM1 to simultaneously encode both the forward and backward propagating beams. **(b)** Example DMD and SLM-1 patterns. Left side encodes forward signal and right side encodes backward signal. **(c)** Illustration of example field amplitudes generated in forward and backward signals after the modulation by DMD and SLM-1.

The reference beam for the detection of the MVM-2a output, R2, is displaced vertically with respect to the forward signal beam. In this way, the R2 reference beam does not interfere with the signal until the very last cylindrical lens set, which acts to perform the MVM summation as well as mix the signal and R2 reference beam. For the second layer, the R2 reference is always brighter than the signal, and because the two beams follow the same path they are phase-stable.

A weak reference beam for the sign measurement of the MVM-1 output, R1, is displaced horizontally with respect to the forward signal beam. The R1 reference beam then overlaps with the signal at the conjugate planes of first-slit and vapor cell, due to the action of CL-1. After the cell, the overlapping beams are tapped off via a beam splitter for

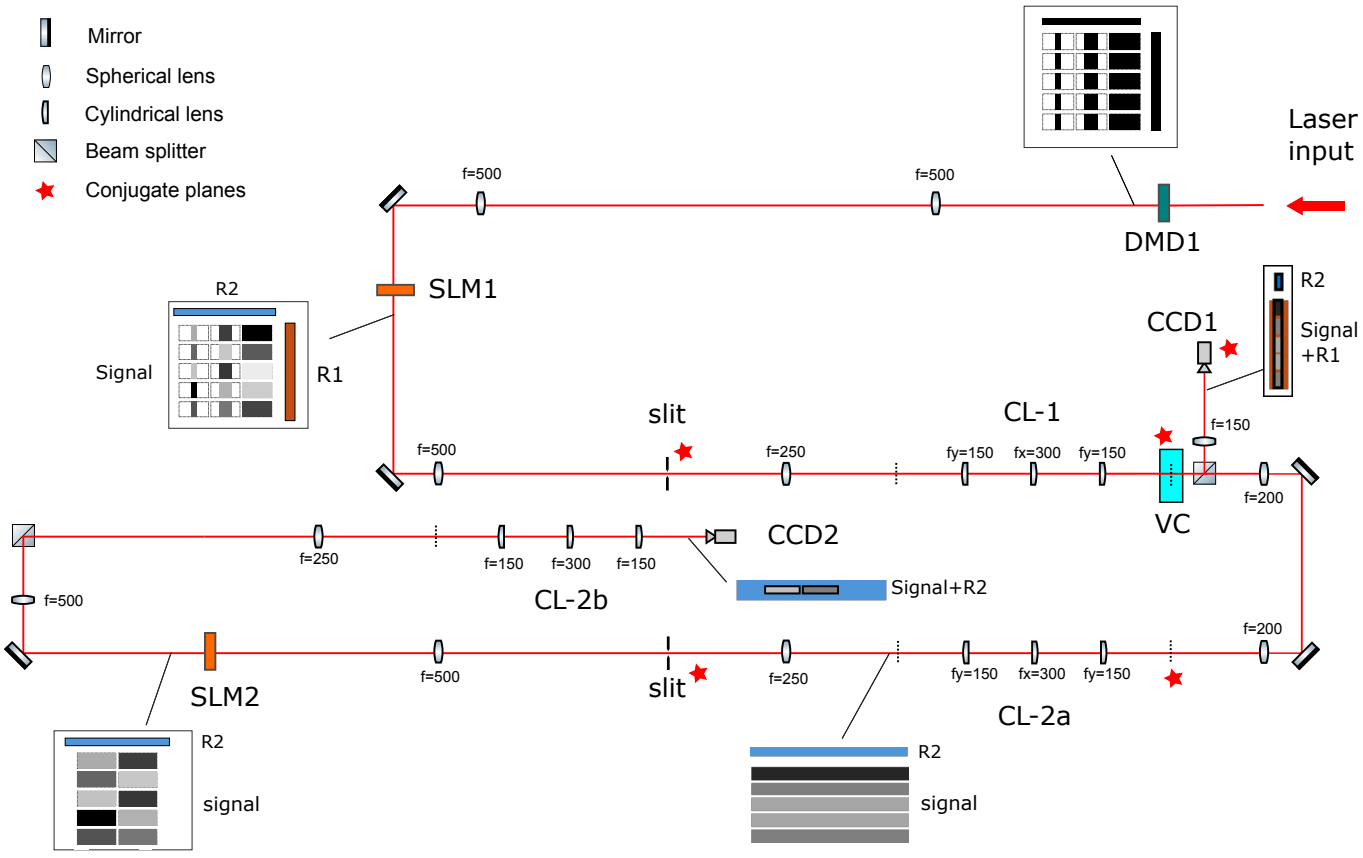

FIG. S3. **Optical path of the forward beam.** The images are representative optical field amplitude patterns as the beam propagate throughout the system.

measurement by CCD1.

The reference beam for the detection of the (backward) MVM-2b output, R3, is displaced horizontally with respect to the backward signal beam. The beam then overlaps with the signal at the conjugate planes of second-slit and vapor cell, due to the action of CL-2. After the cell in the reverse direction, the overlapping beams are tapped off via a beam splitter for measurement by CCD3.

Multiple sequential intensity measurements are made with CCD1, CCD2 and CCD3 in order to measure the three desired vectors: $a^{(1)}$, $z^{(2)}$ and $\delta^{(1)}$. These measurements are summarized in Tab. S1

In the first step, the forward signal and R2 reference are turned on, and all other signals and references turned off. Intensity measurement of the interfered beams at the output by CCD2 gives $I_2 = |E_{z^{(2)}} + E_{\mathrm{R2}}|^2$ and the MVM result can be simply read out as $E_{z^{(2)}} = \sqrt{I_2} - E_{\mathrm{R2}}$. The R2 reference beam passes through the activation vapor cell spatially separated from the signal in the vertical direction, so experiences some attenuation but remains uniform, and does not interfere with the pump-probe process. Two other intensity measurements, $I_{1a} = |E_{a^{(1)}}|^2$ and $I_{3a} = I_{\mathrm{pump}} = |E_{\mathrm{pump}}|^2$ are simultaneously taken with CCD1 and CCD3 respectively. The latter term is leakage of the forward signal into CCD3 due to fluorescence of the beam in the vapor cell, and is later used to make corrections to the backward signal measurement.

In the second step, both the forward signal and R1 reference beam are turned on, and other beams are turned off. The signal and R1 reference combine and interfere at the output of MVM-1, and intensity measurement with CCD1 yields $I_{1b} = |E_{a^{(1)}} + E_{R1}|^2$, allowing us to determine whether $E_{a^{(1)}}$ is positive or negative by comparing the magnitudes of I_{1b} and I_{1a}. The two measurements are necessary because the combination of the signal and reference beams is affected by the SA cell in a nonlinear fashion. The final value can be calculated as $E_{a^{(1)}} = \mathrm{sign}\,(I_{1b} - I_{1a}) \cdot \sqrt{I_{1a}}$.

Then the backward signal requires three intensity measurements (steps 3-5) for subtraction of all background terms. All three measurements are recorded with CCD3 after tap-off of the beam after the vapor cell. Recall we denote the result of MVM-2b as $\rho^{(1)}$, so we denote the field $E_{\rho^{(1)}}$ when we have only backward signal and no forward signal. When forward signal is present, due to the pump-probe interaction in the vapor cell the field is modulated to approximate the result $\delta^{(1)} = \rho^{(1)} \cdot g'(z^{(1)})$, and we denote the field $E_{\delta^{(1)}}$.

In the third step, only the backward signal is turned on, and the term $I_{3b} = |E_{\rho^{(1)}}|^2$ is measured by CCD3. This

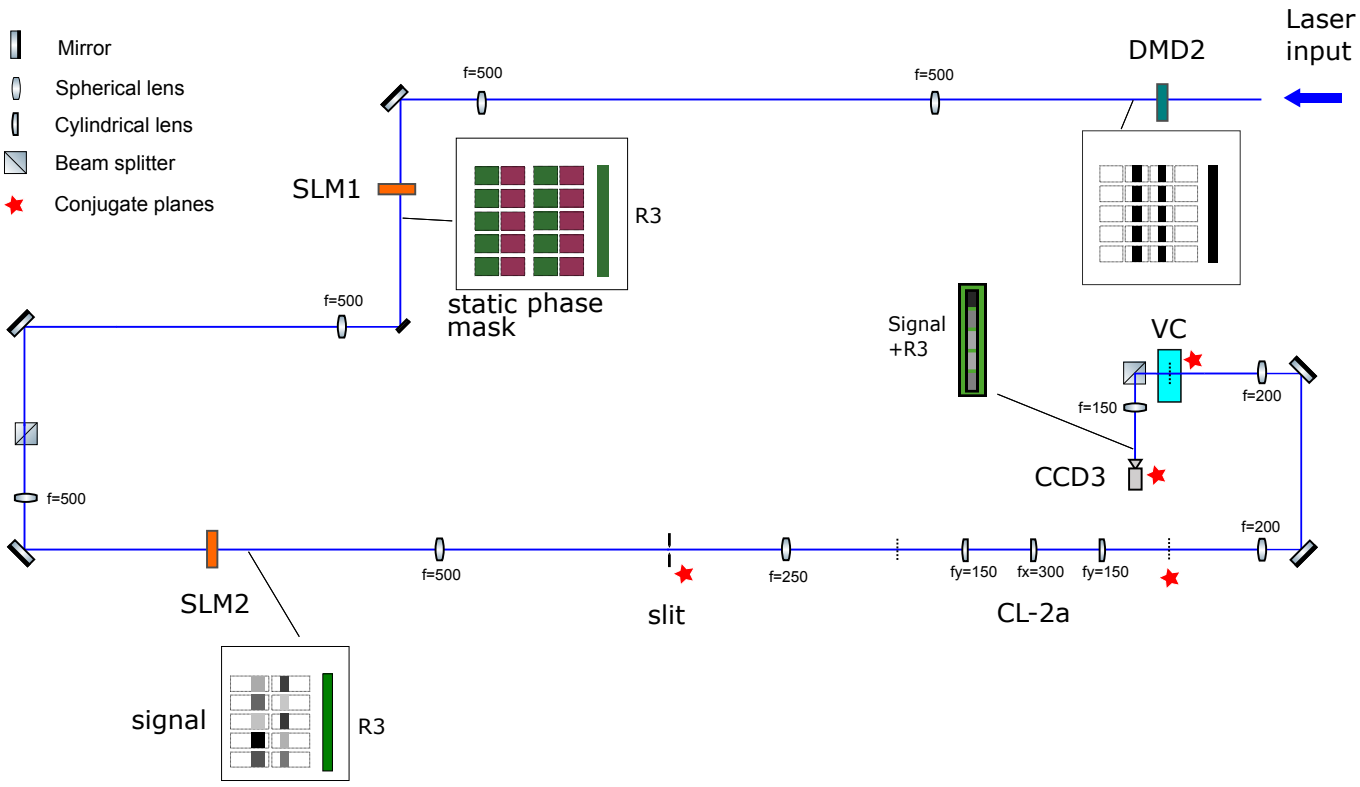

FIG. S4. **Optical path of the backward beam.** The image next to SLM1 is the static phase mask, and other images are representative optical field amplitude patterns.

TABLE S1. **Sequence of optical measurement for optical training.** CCD1 is used to measure forward beam at the hidden layer, CCD2 is used to measure forward beam at the output layer, and CCD3 is used to measure backpropagated error beam at the hidden layer.

Step	**Camera**	**Measurement**	**FW signal**	**BW signal**	**R1**	**R2**	**R3**	**Calculation**
1	CCD1 CCD2 CCD3	$I_{1a} = \lvert E_{a^{(1)}} \rvert^2$ $I_2 = \lvert E_{z^{(2)}} + E_{R2} \rvert^2$ $I_{3a} = I_{pump} = \lvert E_{pump} \rvert^2$	ON	OFF	OFF	ON	OFF	$z^{(2)} = \sqrt{I_2} - R2$ Pump fluorescence background
2	CCD1	$I_{1b} = \lvert E_{a^{(1)}} + E_{R1} \rvert^2$	ON	OFF	ON	OFF	OFF	$a^{(1)} = \text{sign}(I_{1b} - I_{1a}) \cdot \sqrt{I_{1a}}$
3	CCD3	$I_{3b} = \lvert E_{\rho^{(1)}} \rvert^2$	OFF	ON	OFF	OFF	OFF	Unabsorbed probe background
4	CCD3	$I_{3c} = \lvert E_{\rho^{(1)}} + E_{R3} \rvert^2$	OFF	ON	OFF	OFF	ON	
5	CCD3	$I_{3d} = \lvert E_{\delta^{(1)}} + E_{pump} + E_{\rho^{(1)}} \rvert^2$	ON	ON	OFF	OFF	OFF	$\delta^{(1)} = \text{sign}(I_{3c} - I_{3b}) \cdot \left(\sqrt{I_{3d}} - \sqrt{I_{3a}} - \sqrt{I_{3b}}\right)$

gives the background term due to the unabsorbed probe signal. In the fourth step, both the backward signal and R3 reference are turned on, and all the other beams are off, and the term $I_{3c} = |E_{\rho^{(1)}} + E_{R3}|^2$ is measured, allowing us to determine the sign of $\rho^{(1)}$. This will equal the sign of $\delta^{(1)}$ as the pump-probe process has no affect on phase. Finally, in the fifth step, both forward and backward signal are turned on, and all the reference beams are off. The two beams interact within the vapor cell under the pump-probe mechanism. The intensity of the backward signal is measured as $I_{3d} = |E_{\delta^{(1)}} + E_{\text{pump}} + E_{\rho^{(1)}}|^2$, where there are two undesired background terms due to pump fluorescence and unabsorbed probe. The final result is measured as $E_{\delta^{(1)}} = \text{sign}\left(I_{3c} - I_{3b}\right) \cdot \left(\sqrt{I_{3d}} - \sqrt{I_{3a}} - \sqrt{I_{3b}}\right)$.

Supplementary Note 3. ADDITIONAL DATA ON OPTICAL TRAINING

Figure S5 shows the learning curves and the decision boundary evolution for the two training sets, akin to Fig. 3(a) in the main text.

Figure S6 compares the learning curves between optical training and digital training for the Rings dataset. The optical learning curves are the same as that shown in Fig. 3(b) in the main text. As we can see, optical training converges at a similar rate with digital training, despite the perturbation of experimental imperfections and noise. Digital training can achieve high validation accuracy, but this accuracy is not tested on the ONN. After applying the digitally trained weights to the ONN, the test accuracy is below that of optical training.

Figure S7 shows the evolution of optically estimated gradients and digitally calculated gradients during the training process for the three datasets. We can see that the optically estimated gradients match the digitally calculated gradients very well throughout the training process in spite of some small deviation. When digital gradients approach zero, so are optical gradients. When digital gradients fluctuate, optical gradients show the same pattern. The fluctuation is a nature of stochastic gradient descent. Besides, in the early training phase for the XOR dataset, it's evident that both gradients are highly correlated, which is another unambiguous evidence of optical gradient descent.

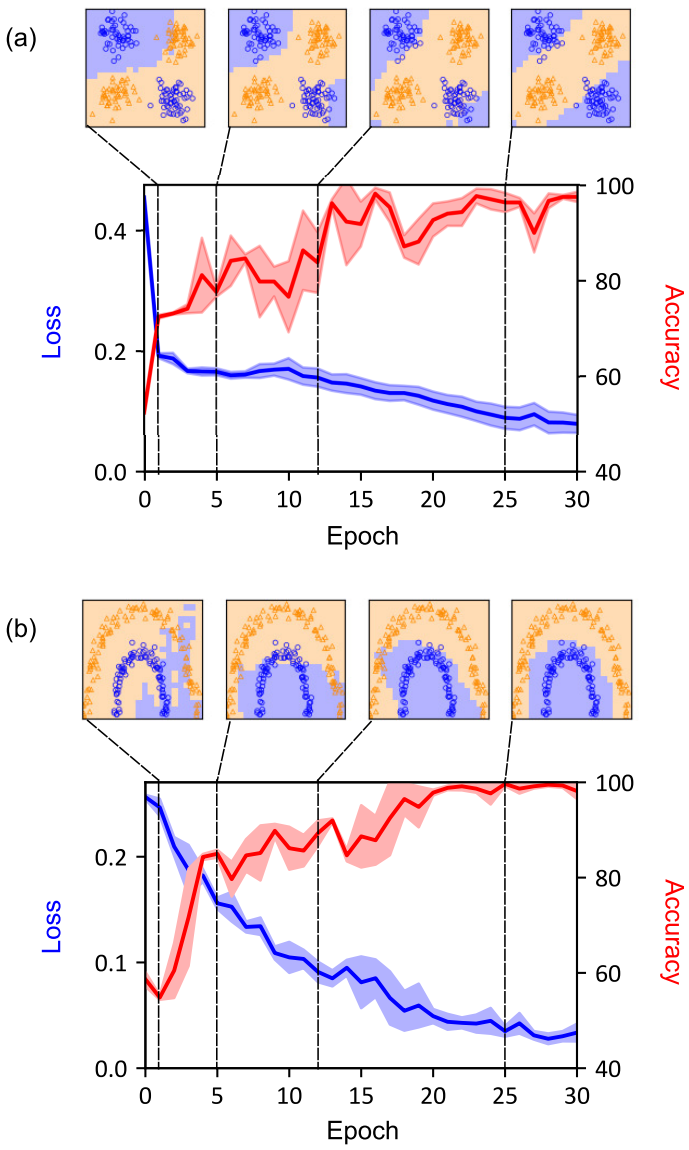

FIG. S5. **Optical training result for the 'XOR' and 'arches' datasets.(a)** Mean and standard deviation of validation loss and accuracy during optical training of the 'XOR' dataset. Shown above are example boundary plots of the test dataset after certain epochs. **(b)** As above, for the 'arches' dataset.

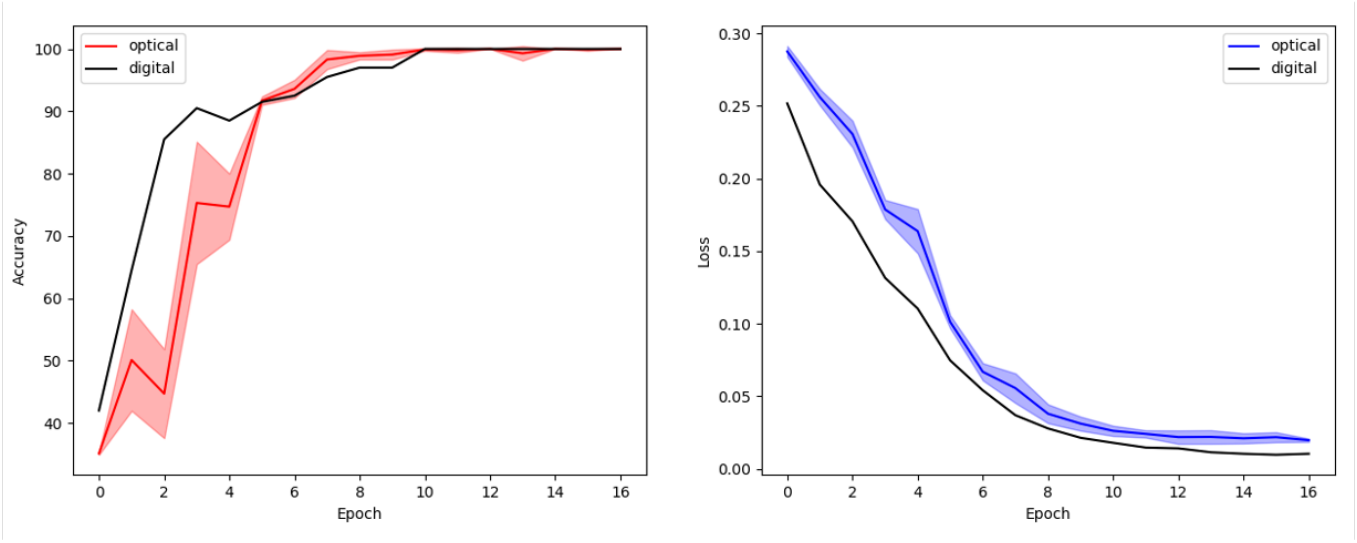

FIG. S6. **Comparison of the accuracy and loss curves between optical training and digital training for the Rings dataset.**

We note that these gradients are further scaled by the learning rate and the Adam optimiser before being applied to update the weights. Therefore, though the gradients may fluctuate around zero as training iteration increases, the actual weights have converged.

Figure S8 shows one time-lapse frame of Supplementary Video 1, which records a complete run of end-to-end optical training for the Rings dataset. As the weight and bias parameters are updated, the loss function and accuracy plots converge quickly, as reflected in the evolution of the decision boundary. The activation values and backpropagated error values were recorded by cameras and plotted against theoretical values that are calculated from MVM and SA function.

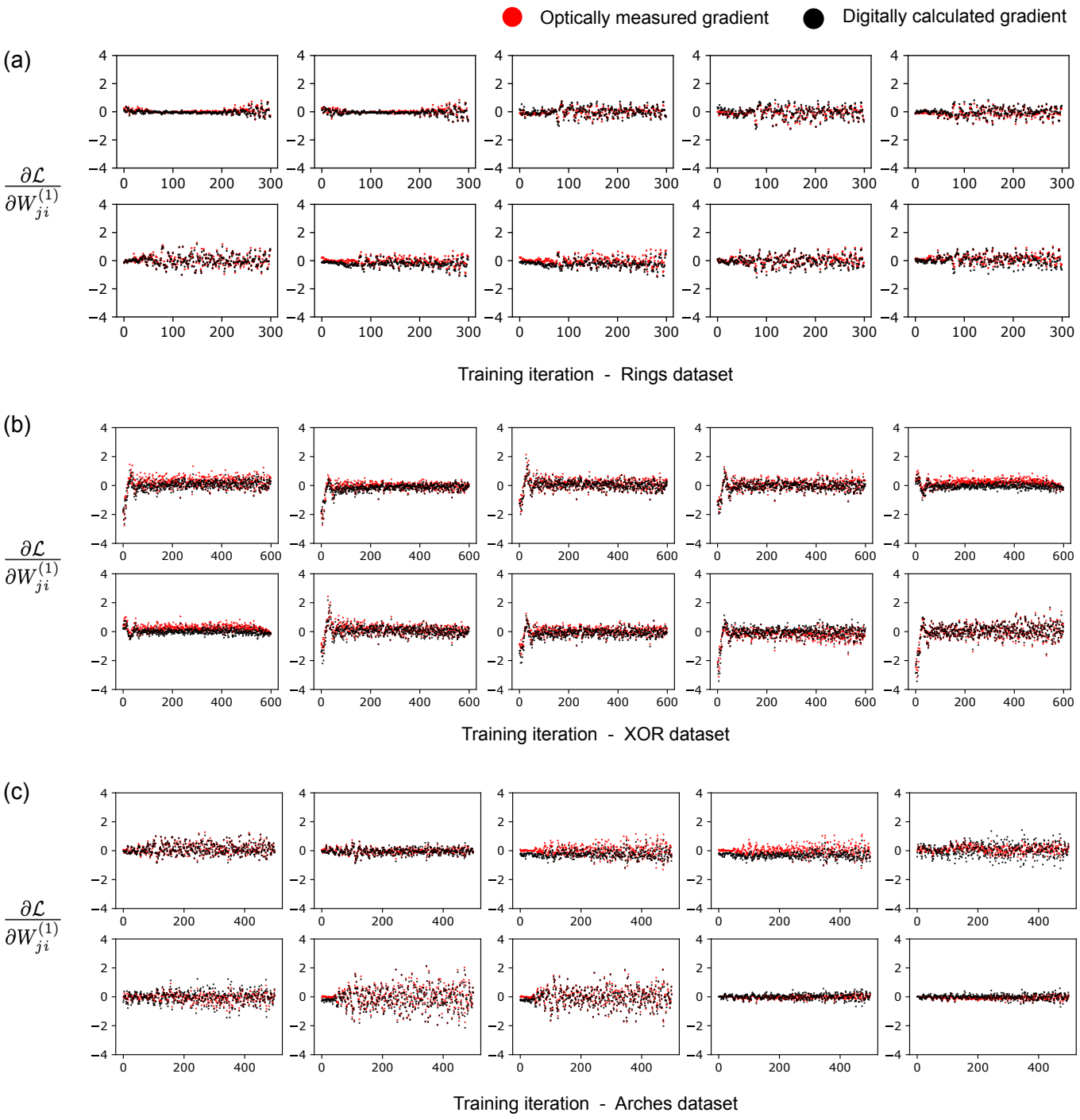

FIG. S7. **Evolution of optically estimated and digitally calculated gradients for (a) 'Rings' dataset, (b) 'XOR' dataset, and (c) 'Arches' dataset.** Each subfigure shows gradients for each of the ten weight matrix elements at the first layer. These gradients are subsequently scaled by the learning rate and Adam optimiser before being applied to update the weights.

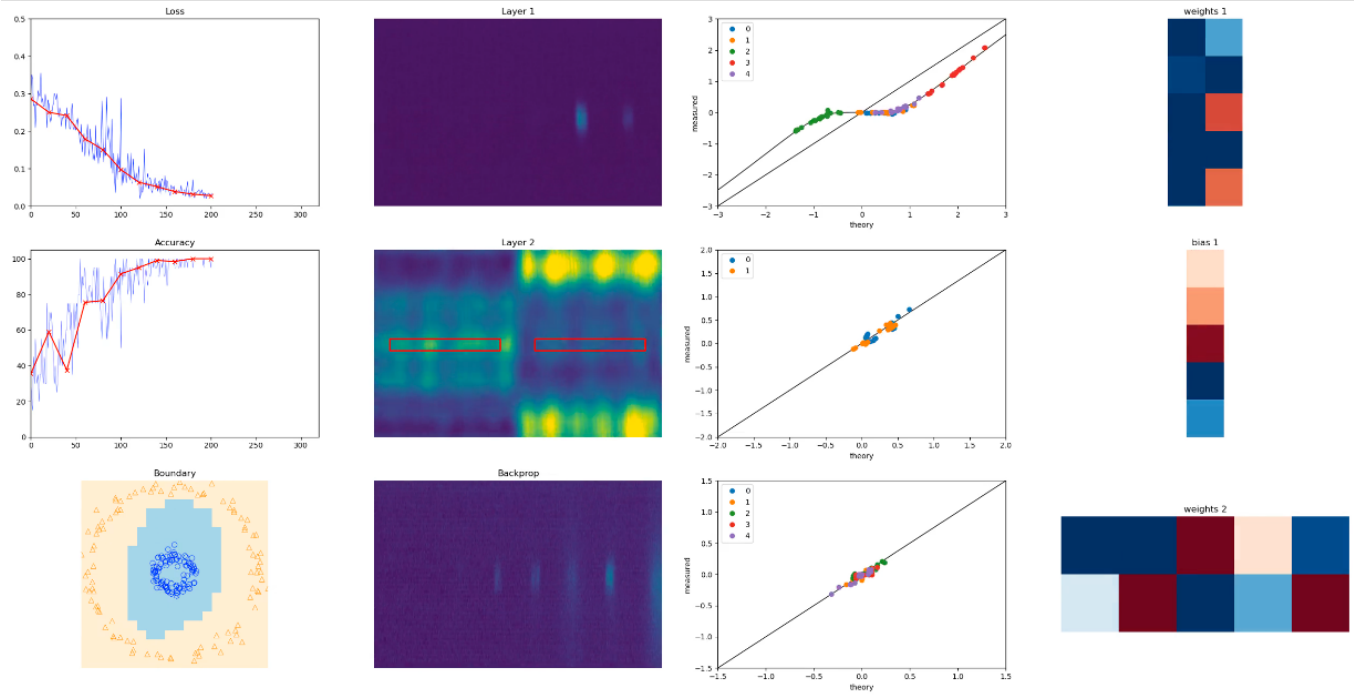

FIG. S8. **Time-lapse of one complete end-to-end optical training process for the Rings dataset (Video 1). First column**: evolution of network loss (top), validation accuracy (middle) and decision boundary (bottom) during optical training. **Middle columns**: recorded camera images of CCD1 ('Layer 1'), CCD2 ('Layer 2') and CCD3 ('Backprop'). Associated plots show the optically-calculated values ('measured') plotted against the digitally-calculated theoretical values ('theory'), for each of $a^{(1)}$ (top), $z^{(2)}$ (middle), and $\delta^{(1)}$ (bottom). Different colors represent values of different neurons. **Last column**: evolution of first-layer weight matrix $W^{(1)}$ ('weights 1') and bias vector $b^{(1)}$ ('bias 1') and evolution of second-layer weight matrix $W^{(2)}$ ('weights 2'). Values are normalized within $+1$ (dark blue) and -1 (dark red).

Supplementary Note 4. OPTICAL BACKPROPAGATION THROUGH THE LINEAR LAYER

To carry out optical backpropagation in an ONN, the linear layers need to be bidirectional such that light can propagate in both directions, and achieve a transpose of the weighting or transform matrix in the backward direction. In this section, we will explain how to achieve optical backpropagation through three types of linear layers: MVM, diffraction and convolution layer.

A. Photonic MVM

Our optical MVM is built with free-space optics using lenses and spatial light modulators following the 'Stanford multiplier' design first proposed by Goodman in 1978 [28]. There are of course different ways to achieve optical MVM. For example, on the photonic platform, one can also perform MVM using a crossbar array architecture, as illustrated in Fig. S9(a). In this setup, a micro-resonator is placed at each crossing of two waveguides. Forward propagating signal enters from the left side of the crossbar array and exits from the bottom. Each element of the weight matrix is controlled by a variable beamsplitter — for example, a micro-resonator, so the transmission and reflection of the signal at each crossing depends on the resonance condition, which can be electrically controlled. Forward signals from multiple rows are thus weighted and summed up to yield the MVM output. To perform optical backpropation, we can inject the error vector from the bottom. The microresonators will then reflect it towards the left side, such that the signal and error beams counter-propagate. A more convenient option is to send the error vector from the top side of the crossbar array and direct it towards the right output ports [30]. The error vector from different columns are weighted and summed up, yielding MVM result with a transpose of the weight matrix. The signal and error beams should be separated either spectrally or temporally, such that the cross-talks or undesired interference at the output ports can be minimized.

Similar backpropagation is possible in other photonic implementations of MVM, such as that based on networks of Mach-Zehnder interferometers (MZIs). However, the crossbar array case is simpler in that the reflectivity of every beamsplitter directly corresponds to an element of the weight matrix. Hence the backpropagation process as described in this paper will directly yield the gradient for the beamsplitters update. In the case of MZI network, the relation between the weight matrix and the beamsplitter reflectivities is complicated. An additional calculation is therefore needed to obtain the gradient of each phase shifter setting. As shown recently shown both theoretically [24] and experimentally [25], this additional computing step can also be implemented optically.

B. Diffraction layer

In recent years, diffractive neural networks have attracted significant research interest [6], and they have been shown to achieve high performance in various machine learning and machine vision tasks such as image recognition and object tracking. Diffraction is a natural optical phenomenon, and diffractive layers can be constructed using 3D printed masks, liquid-crystal SLMs and meta-materials. In Fig. S9(b) we depict the operation principle of a diffractive neural network. In this network, neurons at neighbouring layers are connected via diffraction, and we denote $W_{ki}^{(l-1)}$ as the complex factor acquired by the amplitude of the light field propagating from neuron $z_k^{(l-1)}$ at layer $l-1$ to neuron $z_i^{(l)}$ at layer l. These coefficients are fixed by the ONN geometry and usually not trainable once the network is fabricated. Trainable parameters in the network are the complex transmission coefficients $t_{\cdot}^{(\cdot)}$ of each neuron at different layers.

During the forward direction, signals propagate from one layer to the next as

$$z_i^{(l)} = \sum_k W_{ik}^{(l-1)} t_k^{(l-1)} z_k^{(l-1)}. \tag{S13}$$

To train the network, we calculate the gradient of the loss function with respect to the transmission coefficient of each layer:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial t_i^{(l)}} &= \sum_p \frac{\partial \mathcal{L}}{\partial z_p^{(l+1)}} \cdot \frac{\partial z_p^{(l+1)}}{\partial t_i^{(l)}} \\ &= \sum_p \frac{\partial \mathcal{L}}{\partial z_p^{(l+1)}} W_{pi}^{(l)} z_i^{(l)}, \end{aligned} \tag{S14}$$

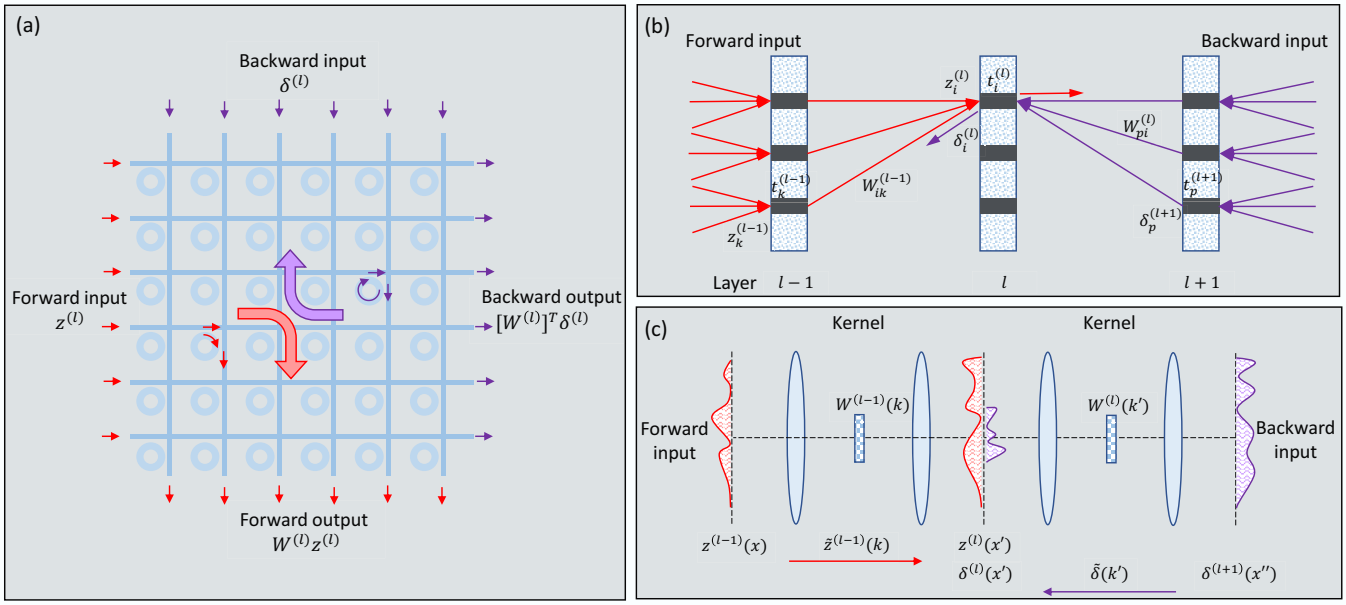

FIG. S9. **Different linear optical layers supporting direct optical backpropagation. (a)** A photonic crossbar MVM built with waveguides and micro-resonators. **(b)** A diffractive neural network constructed with multiple phase or amplitude masks. **(c)** Optical convolution layers implemented via 4f lens configurations.

where $W_{pi}^{(l)}$ is known *a priori*, and $z_i^{(l)}$ can be measured experimentally in the forward propagation step. We denote $\delta_p^{(l+1)} = \partial\mathcal{L}/\partial z_p^{(l+1)}$ as the error vector at layer $l+1$, then the error vector at the preceding layer l can also be calculated by applying the chain rule of calculus:

$$\begin{aligned}\delta_i^{(l)} &= \sum_p \frac{\partial\mathcal{L}}{\partial z_p^{(l+1)}} \cdot \frac{\partial z_p^{(l+1)}}{\partial z_i^{(l)}} \\ &= \sum_p \delta_p^{(l+1)} W_{pi}^{(l)} t_i^{(l)}.\end{aligned} \tag{S15}$$

The last line indicates that the error vector at a given layer is calculated by multiplying the error vector at a subsequent layer with the diffractive connection matrix transpose, then element-wise multiplied with the mask transmission coefficients. This is exactly the optical backpropagation process. The physical forward and backward propagation process is better evidenced by re-writing Eq. S13 and Eq. S15 in the matrix form:

$$z^{(l)} = z^{(l-1)} \cdot t^{(l-1)} \times W^{(l-1)}, \tag{S16}$$

$$\delta^{(l)} = \delta^{(l+1)} \times [W^{(l)}]^T \cdot t^{(l)}. \tag{S17}$$

C. Convolution layer

The convolutional neural network represents a basic type of neural networks, and it is ubiquitous in processing complex machine vision tasks. It can capture key features of images efficiently through the use of convolution kernels layer-by-layer. It is well known that convolution is equivalent to multiplication in the Fourier domain, and Fourier transform can be optically performed with a simple lens.

Figure S9(c) depicts two cascaded optical convolution layers using two lenses per layer comprising a 4f system. The first lens at layer $l-1$ performs Fourier transform of the incoming signal $z^{(l-1)}(x)$:

$$\tilde{z}^{(l-1)}(k) = \mathcal{F}[z^{(l-1)}(x)] = \int z^{(l-1)}(x) e^{-ikx} dx. \tag{S18}$$

Then a mask $W(k)$ (which is the Fourier transform of the convolution kernel) placed at the focal plane modulates the signal spectrum $\tilde{z}^{(l-1)}(k)$, and a second lens performs another Fourier transform to complete the convolution:

$$z^{(l)}(x') = \mathcal{F}[\tilde{z}^{(l-1)}(k)W^{(l-1)}(k)] = \iint z^{(l-1)}(x)W^{(l-1)}(k)e^{-ik(x+x')}dxdk. \tag{S19}$$

To update the kernels, we calculate the gradient of the loss function with respect to the transmissivity $W(k)$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W^{(l-1)}(k)} &= \int \frac{\partial \mathcal{L}}{\partial z^{(l)}(x')}\frac{\partial z^{(l)}(x')}{\partial W^{(l-1)}(k)}dx' \\ &= \iint \frac{\partial \mathcal{L}}{\partial z^{(l)}(x')} z^{(l-1)}(x)e^{-ik(x+x')}dxdx' \\ &= \iint \delta^{(l)}(x')z^{(l-1)}(x)e^{-ik(x+x')}dxdx' \\ &= \tilde{z}^{(l-1)}(k)\cdot\tilde{\delta}^{(l)}(k).\end{aligned} \tag{S20}$$

In the calculation above we have introduced the error vector $\delta^{(l)}(x') = \dfrac{\partial \mathcal{L}}{\partial z^{(l)}(x')}$ and its Fourier spectrum $\tilde{\delta}^{(l)}(k) = \int \delta(x)e^{-ikx'}dx$. This result means that the kernel gradient is the dot product of the signal spectrum and error vector spectrum. Next we show how the error vector spectrum can be obtained by applying the chain rule of calculus:

$$\begin{aligned}\tilde{\delta}^{(l)}(k) &= \int \frac{\partial \mathcal{L}}{\partial z^{(l)}(x')}e^{-ikx'}dx' \\ &= \iint \frac{\partial \mathcal{L}}{\partial z^{(l+1)}(x'')}\frac{\partial z^{(l+1)}(x'')}{\partial z^{(l)}(x')}e^{-ikx'}dx'dx'' \\ &= \iiint \delta^{(l+1)}(x'')W^{(l)}(k')e^{-ik'(x'+x'')}e^{-ikx'}dk'dx'dx'' \\ &= \int \delta^{(l+1)}(x'')W^{(l)}(k')e^{-ik'x''}dx'' \int e^{-i(k'+k)x'}dx' \\ &= \tilde{\delta}^{(l+1)}(-k)\cdot W^{(l)}(k).\end{aligned} \tag{S21}$$

Here we used the fact that $\int e^{-i(k'+k)x'}dx'$ is the Dirac delta function with respect to $k'+k$.

From this result, we see that the error vector spectrum at layer l is obtained by multiplying the error vector spectrum at the next layer $l+1$ with the kernel at layer l. The negative sign in front of k in $\tilde{\delta}^{(l+1)}(-k)$ is due to the image inversion in a 4f system. Therefore, optical convolution layers also support direct optical backpropagation.

To implement optical pooling after the convolution layer, we can now include an aperture function $A(k)$ on top of the kernel mask $W(k)$, such that the field distribution $z^{(l)}(x')$ in Eq. (S19) is further convolved with the point-spread function $\mathcal{F}[A^{(l-1)}(k)]$. For example, a Sinc function aperture on top of the kernel mask would produce average pooling result. Since the aperture function can be absorbed into the mask $W(k)$ at each layer, the optical backpropagation analysis (Eq. (S18) - Eq. (S21)) is still valid for a convolution layer following by average pooling.