

Hybrid training of optical neural networks

JAMES SPALL,^{1,†}  XIANXIN GUO,^{1,2,3,†}  AND A. I. LVOVSKY^{1,2,4}

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

²OxONN, Wood Centre for Innovation, Quarry Road, Headington, Oxford, OX3 8SB, UK

³e-mail: xianxin.guo@physics.ox.ac.uk

⁴e-mail: alex.lvovsky@physics.ox.ac.uk

Received 14 February 2022; revised 25 May 2022; accepted 6 June 2022; published 14 July 2022

Optical neural networks are emerging as a promising type of machine learning hardware capable of energy-efficient, parallel computation. Today's optical neural networks are mainly developed to perform optical inference after *in silico* training on digital simulators. However, various physical imperfections that cannot be accurately modeled may lead to the notorious "reality gap" between the digital simulator and the physical system. To address this challenge, we demonstrate hybrid training of optical neural networks where the weight matrix is trained with neuron activation functions computed optically via forward propagation through the network. We examine the efficacy of hybrid training with three different networks: an optical linear classifier, a hybrid opto-electronic network, and a complex-valued optical network. We perform a study comparative to *in silico* training, and our results show that hybrid training is robust against different kinds of static noise. Our platform-agnostic hybrid training scheme can be applied to a wide variety of optical neural networks, and this work paves the way towards advanced all-optical training in machine intelligence.

Published by Optica Publishing Group under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

<https://doi.org/10.1364/OPTICA.456108>

1. INTRODUCTION

Machine learning powered by artificial neural networks has reshaped the landscape in many different areas over the last decade. This machine learning revolution is fueled by the immense parallel computing power of electronic hardware such as graphics- and tensor-processing units. However, the rapid growth of computational demand in this field has outpaced Moore's law, and today's machine learning applications are associated with high energy cost and terrible carbon footprint [1]. We are in dire need of novel computing systems capable of fast and energy-efficient computation to drive the future development of machine learning.

Optics provides a promising analog computing platform, and optical neural networks (ONNs) have recently been the focus of intense research and commercial interest. Due to the superposition and coherence properties of light, neurons in ONNs can be naturally connected via interference [2–4] or diffraction [5–7] in different settings, while the neuron activation function can be physically implemented with a large variety of nonlinear optical effects [3,8,9]. Together these resources have enabled the optical realization of various neural network architectures, including fully connected [2,5–7], convolutional [10–12], and recurrent [3,4,13,14]. Today's advanced optical technologies have already allowed ONNs to reach a computational speed of ten trillion operations per second [10], comparable to that of their electronic counterparts; the energy consumption can be on a scale of, or even less than, one photon per operation [15], orders of magnitude lower than that of digital computation.

Current ONNs are primarily developed to perform inference tasks in machine learning [16,17], and they are usually trained on a digital computer. During this *in silico* training, one has to simulate the physical system digitally, then apply the standard "backpropagation" algorithm [18], which involves repeated forward and backward propagation of information inside the network. The update of the weight matrices is computed from the combined data obtained in these two processes. Because any physical system exhibits certain experimental imperfections that are hard to accurately model, ONNs trained in this way usually perform worse than expected [6,9,19]. To narrow this reality gap, one can incorporate simulated noise into the *in silico* training [7]. However this approach is suboptimal because it does not incorporate the specific pattern of imperfections that is present in a given ONN. Another approach is to apply platform-specific error correction algorithms [20–24] or iterative optimization algorithms [25] to reduce experimental imperfections, but running these algorithms can be time and resource demanding.

Our group recently proposed a method for obtaining the training signal directly from the optical fields propagating through the network in both directions [26]. This method not only allows faster training, but also helps close the reality gap, as the actual physics of the system, including its imperfections, is built into the training.

In this work, we demonstrate the first step towards this vision. In our training scheme, the forward linear computation and backward propagation of information are implemented with optics and electronics, respectively. The portion of the training signal that is acquired from forward propagation is then obtained through direct

measurement of the optical field followed by digital calculation of the nonlinear activation function.

This hybrid training method, also known as “hardware-in-the-loop training,” is a well-established technique that incorporates the actual hardware response in each update loop to mitigate the impact of hardware imperfections, and has been demonstrated in various electronic analog neural networks as well as neuromorphic computing platforms based on memristors [27–30]. Recently, Wright *et al.* [19] demonstrated “physics-aware training” of an opto-electronic network. However, in this case, the optical system did not contain a fully tunable weight matrix, so an additional fully connected digital linear layer had to be included. Since the computational advantages of ONNs reside in optical linear layers with controllable weight matrices, and experimental imperfections often originate from the optical linear connection, optical propagation through the entire network is required for proper hybrid training.

Another related work is that of Zhou *et al.* [6]. After training a three-layer ONN *in silico*, the authors tested their network optically, made corrections to the digital models of the second and third layers to account for the measured performance of the first layer, and digitally re-trained these layers. The optical testing and re-training were then repeated just for the third layer. While this is a step towards the hybrid training scheme, a major shortcoming of this approach is the requirement of initial *in silico* training. Another shortcoming is that the behavior of later layers must be well characterized and modeled to compensate for the errors caused by previous layers. Furthermore, the training must be repeated multiple times.

In this work, we demonstrate hybrid training on an ONN platform with a fully connected layer. Full connectivity provides compatibility with digital neural network architectures and hence a degree of universality that is not offered by other architectures such as reservoir computing, kernel machines, and spiking neural networks. We construct three different ONNs: an optical linear classifier, a hybrid neural network with optical and electronic layers, and a complex-valued ONN. Our work is different from that of Wright *et al.* because all these networks contain optical linear layers. And unlike Zhou *et al.*, initial *in silico* training is not required, and the optical signal is used to compute every single update of the physical weight matrices. We further analyze the performance of our networks under the influence of various types of noise and find significant improvement in comparison to *in silico* training provided that the noise is static, i.e., does not change in time.

2. RESULTS

A. Network Training

In supervised learning, the weight matrices of a neural network are iteratively updated (trained) via the “backpropagation” algorithm [18], eventually enabling the network to replicate the mapping between the network input and the ground truth. The training is implemented using a labeled dataset (\mathbf{x}, \mathbf{t}) , where \mathbf{x} is sent to the network input ($a_i^{(0)} = x_i$), and \mathbf{t} is the label to be compared with the network output. The neurons in subsequent layers are interconnected as

$$z_j^{(l)} = \sum_i w_{ji}^{(l)} a_i^{(l-1)}, \quad (1)$$

where $a_j^{(l)} = g(z_j^{(l)})$ is the nonlinear activation of each neuron. A loss function, \mathcal{L} , is defined to quantify the divergence between the network output and the correct label. Its gradient with respect to the weights is

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{ji}^{(l)}} = \delta_j^{(l)} a_i^{(l-1)}, \quad (2)$$

where $\delta_j^{(l)} \equiv \partial \mathcal{L} / \partial z_j^{(l)}$ is referred to as the “error” at the j th neuron in the l th layer. By applying the chain rule of calculus, we have

$$\delta_j^{(l)} = \sum_k \frac{\partial \mathcal{L}}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} = g'(z_j^{(l)}) \rho_j^{(l+1)}, \quad (3)$$

where $\rho_j^{(l+1)} = \sum_k w_{kj}^{(l+1)} \delta_k^{(l+1)}$. From Eq. (3), we see that the error vector inside the network can be calculated from the error vector at the subsequent layer, and the error vector at the output layer is directly calculated from the loss function. Once these error vectors, as well as the activations $a^{(l-1)}$ of all neurons, are known, the gradients [Eq. (2)] of the loss function with respect to all weights can be calculated, and hence the weights can be iteratively updated via gradient descent until convergence. This standard procedure is efficient in training digital electronic neural networks (DENNs). To train an ONN, one can model the network architecture on the computer, and implement the backpropagation algorithm digitally. The final weights after training are then transferred to the ONN to perform inference tasks. This is called the *in silico* training method.

As we see from Eq. (2), the gradient matrix in each layer is the outer product of the corresponding activation and error vectors. In our hybrid training scheme, the activation vectors are obtained through optical forward propagation of neuron values [Fig. 1(a), top panel]. These values are measured by photodetectors and recorded digitally. The error vectors, on the other hand, are obtained through digital backpropagation [Fig. 1(a), bottom panel]. Once the weight gradients [Eq. (2)] are calculated, the physical weights of the ONNs are updated. This hybrid training process is repeated until convergence.

B. Experimental System

In a neural network, the interconnection of neurons [Eq. (1)] is achieved by matrix-vector multiplication (MVM), and this basic operation constitutes the major computational workload in machine learning. Here we build a reconfigurable, coherent optical multiplier with spatial light modulators (SLMs) and lenses, as shown in Fig. 1(b).

We encode the neuron values in the electric field amplitude of the light propagating through the network. The positive-valued input vector is converted to a 4-bit integer and then encoded in the spatial field distribution of the light by a digital micromirror device (DMD, Texas Instruments DLP 6500), which is a binary amplitude-only modulator. Each input vector element is encoded in a rectangular pixel block, whose width is 15 pixels, and height is the entire vertical dimension of the DMD. The 4-bit vector element value is represented by the fraction of pixels that are switched “on” in each block. A liquid-crystal phase-only SLM (LC-SLM, Santec SLM 100, 10-bit) is used to further encode an arbitrary real-valued or complex-valued weight matrix. To this end, we apply a phase grating pattern on the LC-SLM and adjust the local offset

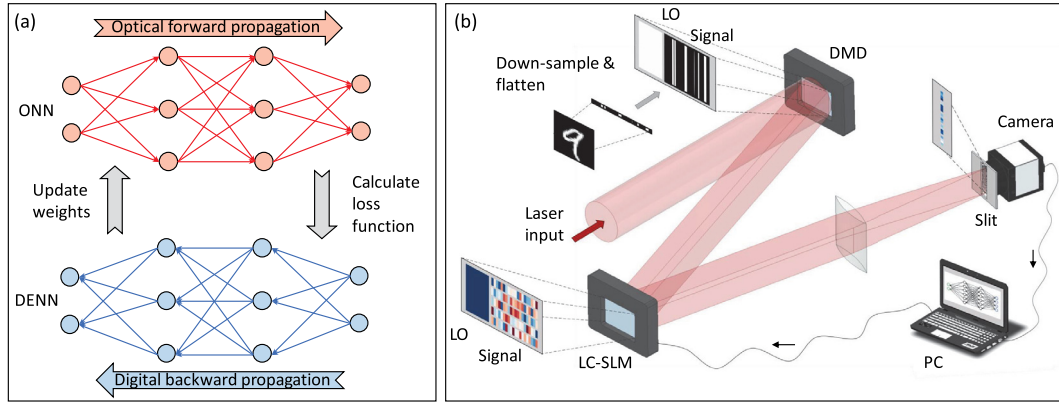


Fig. 1. Hybrid training of ONNs. (a) Conceptual illustration of the scheme, where forward propagation is implemented in the optical system and error backpropagation is implemented digitally. (b) Schematic of our ONNs built upon a single-layer optical multiplier.

and height of the grating so as to obtain the required phase and amplitude pattern in the first diffraction order [31,32]. With this encoding method, we are also able to correct for various optical aberrations and the severe wavefront distortion from both SLMs. As the light passes through the DMD and LC-SLM planes, each row of the weight matrix is multiplied in an element-wise fashion by the input vector, and the results are then summed by means of a cylindrical lens. The output is measured with a fast CMOS camera (Basler ace acA640-750 μm , 8-bit). The actual setup includes several additional lenses [not shown in the simplified schematic of Fig. 1(b)] for the elimination of unwanted diffraction. Details of our encoding method and actual setup were illustrated in our previous work [31].

To read out the neuron values encoded in the optical field amplitudes, we perform homodyne detection. This detection requires a local oscillator (LO) that must be phase-stable with respect to the signal. We address this requirement by allocating part of the DMD and LC-SLM active area as the reference region, and keeping these pixels fully reflective. The portion of the beam that reflects from the reference region serves as the LO. Both the signal and LO fields propagate through the entire system, and so the cylindrical lens not only completes the MVM, but also mixes the LO field with the MVM result at the output plane. For real-valued MVM, we set the LO phase equal to that of the signal. The intensity measured by the camera then equals $I = (E_{\text{LO}} + E_s)^2$, where E_{LO} and E_s are the amplitudes of the two fields, and we ensure that $E_{\text{LO}} > E_s$. The signal field amplitude is then found according to $E_s = \sqrt{I} - E_{\text{LO}}$. Since both the LO and signal share the same optical path together with all optical elements, their relative phase barely fluctuates, and the phase offset can be conveniently set by the LC-SLM. Our homodyne detection scheme avoids the extra experimental complexity of introducing an external LO beam and actively stabilizing the relative phase.

C. Real-Valued ONNs

We first characterize our optical multiplier by running a large number of real-valued MVMs with matrix sizes of 100×10 and 100×25 , and the results are shown in Figs. 2(a) and 2(b). In these measurements, the input vector and weight matrix elements are random, approximately normally distributed around zero with a standard deviation of 0.5, which is consistent with typical distributions in neural networks. The output vector is normalized

by the maximum possible output obtained when all vector and matrix elements are at maximum. The experimentally measured data fall along the diagonal theoretical line. The root mean square errors (RMSEs) for the two matrix sizes are 0.0024 and 0.0036, respectively, normalized to the output vector range $[-1, 1]$.

Our optical multiplier also naturally supports complex-valued operation; therefore, we next perform complex-valued MVM with the matrix size of 100×10 , and measure both the real and imaginary parts of the output by setting different LO phases on the LC-SLM. In this work, the inputs encoded by the DMD are always positive-valued, while the weights and outputs are real- or complex-valued. However, complex-valued inputs could also be generated by modulating a binary grating on the DMD [33]. Figure 2(c) plots examples of the experimentally measured result on the complex plane, compared with theoretical target values. Figure 2(d) shows the complex-valued MVM precision for real and imaginary parts, and the RMSE values are 0.0034 and 0.0039, respectively.

The precise optical multiplier forms the basis of our ONN, and it can directly work as an optical linear classifier (ONN-1), as shown in Fig. 3(a), top. We perform hybrid training of ONN-1 on the MNIST handwritten digits dataset. The dataset consists of 70,000 grayscale bitmaps of size 28×28 . These images are downsampled to size 10×10 , and then flattened into vectors to be fed to the input layer of the optical linear classifier. At the output layer, 10 neuron values are measured by the camera and used to calculate a digital cross-entropy loss function after applying a digital softmax activation. The target values of the output neurons are zero or one, dependent on the label of the bitmap (“one-hot encoding”). We split the entire dataset into training, validation, and test sets, each consisting of 60,000, 5000, and 5000 bitmaps, respectively. The training is implemented in 500 randomly sampled mini-batches with a mini-batch size of 240. Backpropagation was performed digitally with an Adam optimizer, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and an initial learning rate of 0.01. The weight matrices are initialized randomly from a normal distribution $N(0, 0.5)$.

It is essential that the system errors do not accumulate and blow up during the hybrid training. In Fig. 3(b), we plot the evolution of the optical MVM error during the entire training process, and it is clear that our optical system maintains the precision steadily. Figure 3(c) shows the learning curves during the hybrid training, and we see that both the loss function and validation accuracy

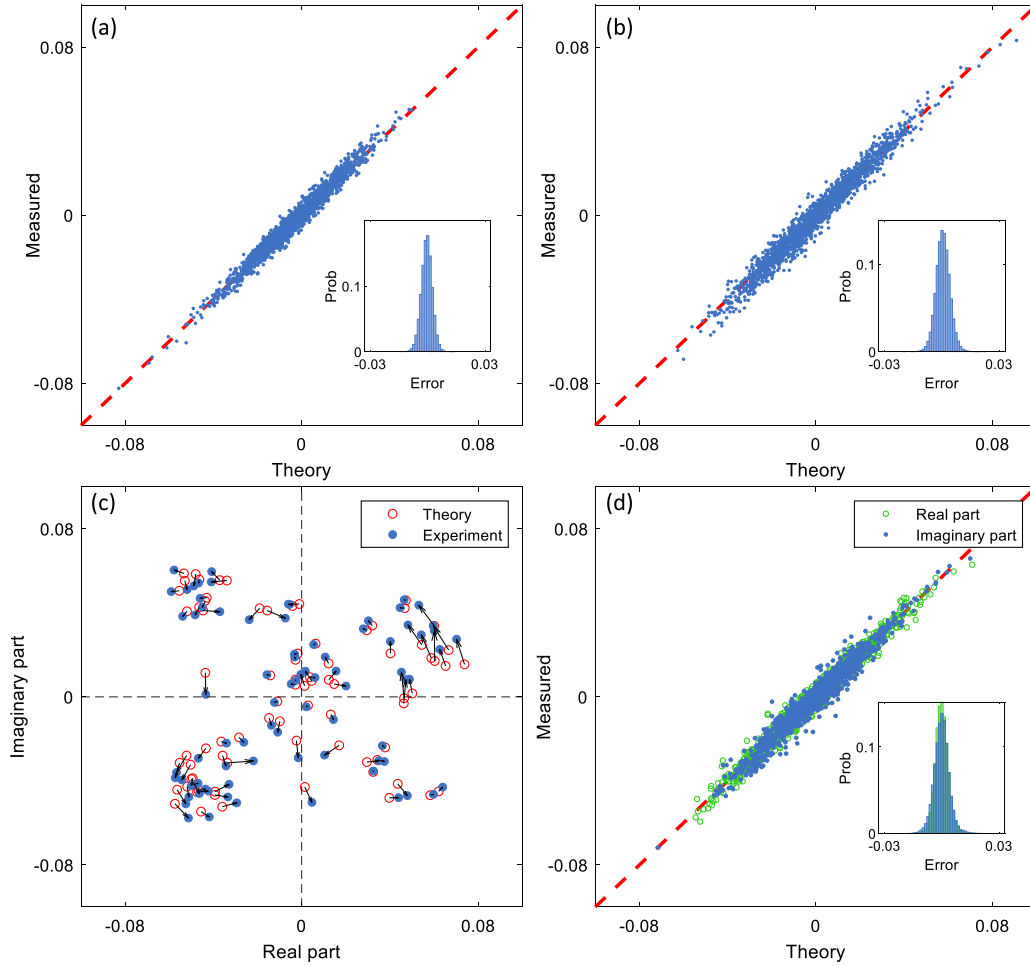


Fig. 2. Characterization of optical matrix-vector multiplication. Performance of real-valued optical MVM with matrix size of (a) 100×10 and (b) 100×25 . Ideal results would fall along the red dashed diagonal line. Distributions of the multiplication error are shown in insets. All values are normalized by the maximum possible output. Because the weight matrix elements are normally distributed around zero, the vast majority of neuron values fall within a narrow range, also centered around zero. (c) Examples of the complex-valued optical MVM output (blue dots) compared with theory targets (red circles) on the complex plane. The differences are indicated by black arrows. (d) Performance of complex-valued optical MVM with matrix size of 100×10 . Real and imaginary parts of the experimental result are shown in green circles and blue dots, respectively.

converge quickly after the first few iterations. The learning curve of a similar digital electronic linear classifier (DENN-1) is also presented for comparison. The DENN is trained with the same hyperparameters and number of iterations. After the hybrid training, we perform image classification on the test set, and the confusion matrix is shown in Fig. 3(d). The classification accuracy reaches 88.0%. In comparison, DENN-1 scores 89.9% after training with the same number of iterations. The slightly slower convergence and lower accuracy of ONN-1 as compared to DENN-1 is mainly due to random dynamic experimental noise, as will be discussed later. We emphasize that during the hybrid training, we apply the standard error backpropagation algorithm without any modeling of the optical system or inclusion of noise.

We next build a more complicated hybrid opto-electronic network (ONN-2) consisting of one optical input layer, one hidden layer with digital rectified linear unit (ReLU) activation and one digital output layer with 100, 25, and 10 neurons at each layer, respectively, as shown in Fig. 3(a), bottom. Such hybrid networks may simplify the acquisition and processing of data in deep optics and Internet of Things (IoT) applications by extracting salient features from the optical front-end [16]. Figure 3(c) plots the learning

curve of this ONN in comparison with a similar digital electronic network (DENN-2). Figure 3(e) shows the confusion matrix of the test set for ONN-2. The test accuracy reaches 92.7%, above the benchmark level of a linear classifier and close to the performance of DENN-2 at 93.2%.

D. Complex-Valued ONN

It has been recently observed [6,7] that diffractive neural networks employing complex-valued operations can outperform linear classifiers, even though the diffractive connections are entirely linear. This is because the intensity detection at the output layer of the complex-valued ONN is equivalent to creating a hidden layer with a square nonlinearity. Consider a single-layer complex-valued optical linear classifier with real-valued inputs E_i and complex-valued weights w_{ji} . At the output layer, we detect the intensity of each output unit:

$$I_j = \left| \sum_i w_{ji} E_i \right|^2 = \left(\sum_i \text{Re}(w_{ji}) E_i \right)^2 + \left(\sum_i \text{Im}(w_{ji}) E_i \right)^2. \quad (4)$$

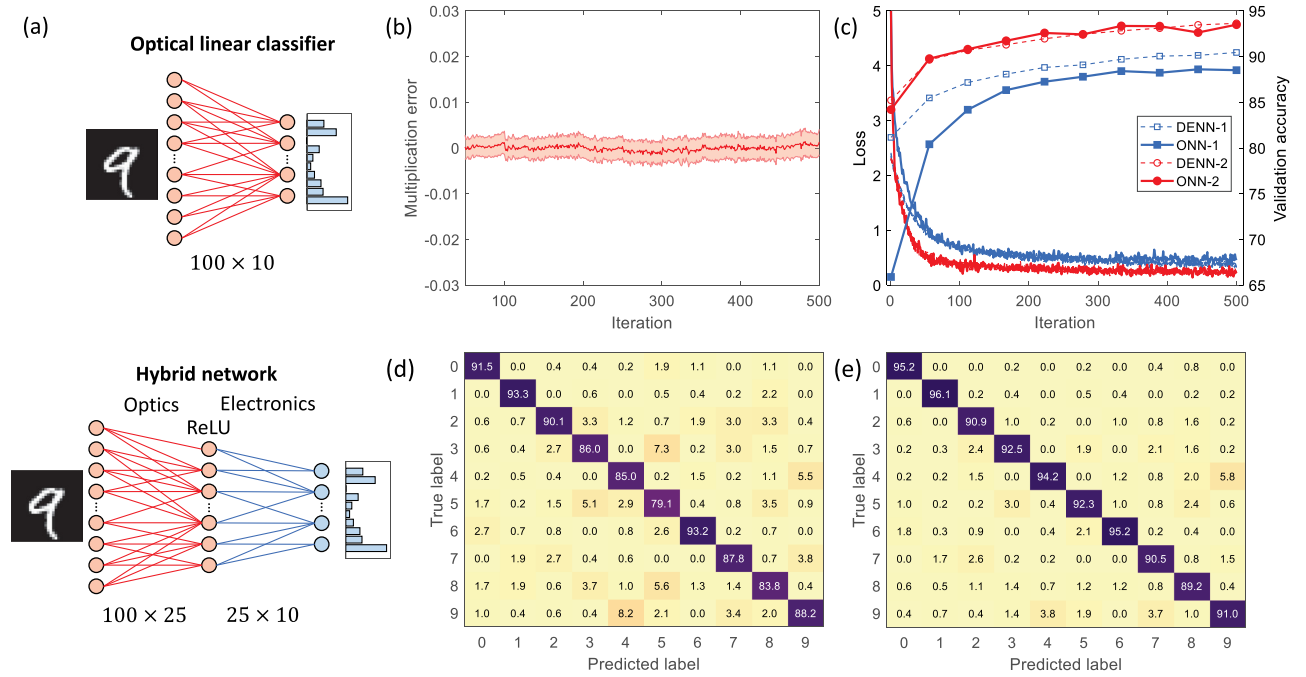


Fig. 3. Hybrid training of an optical linear classifier (ONN-1) and a hybrid opto-electronic network (ONN-2). (a) Top: network architecture of ONN-1 with 100 input neurons and 10 output neurons; bottom: network architecture of ONN-2 with an optical layer, digital ReLU activation, and a digital layer. The neuron numbers are 100, 25, and 10 for the input, hidden, and output layers, respectively. (b) Evolution of the optical MVM error during training of ONN-1. (c) Learning curves of hybrid-trained ONNs compared with DENN benchmarks. DENNs have the same network architectures as those of the two ONNs. (d) Confusion matrix of the test set for ONN-1. (e) Confusion matrix of the test set for ONN-2.

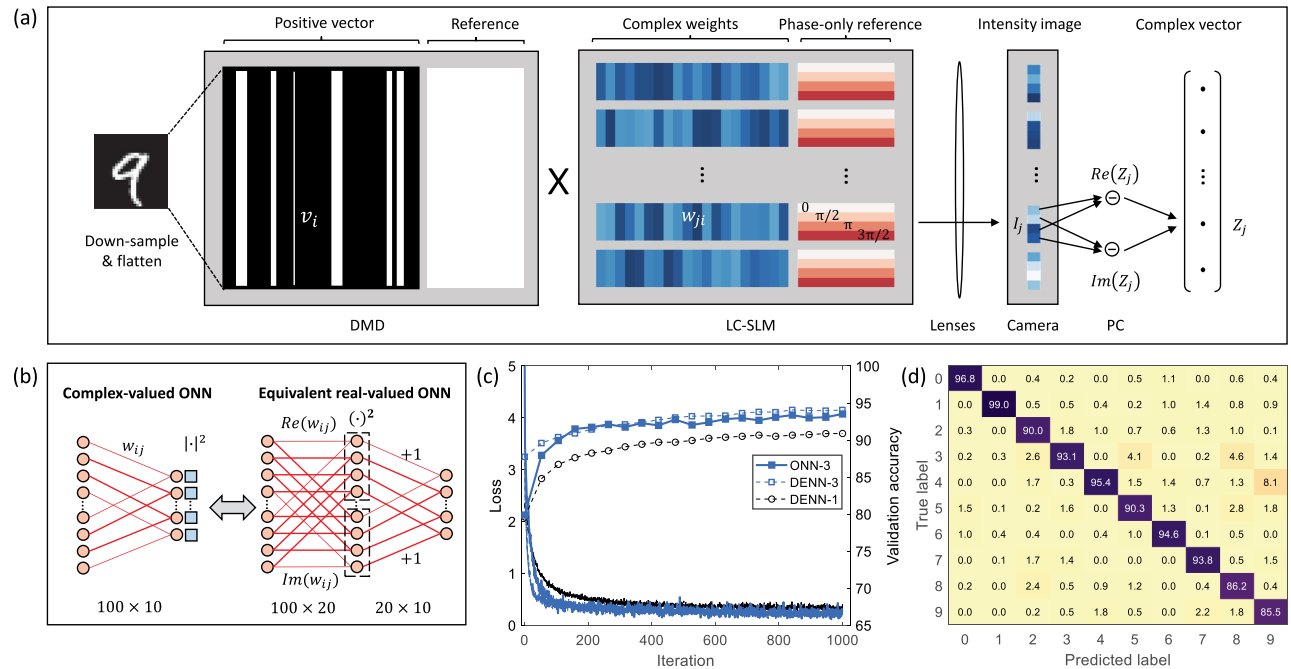


Fig. 4. Hybrid training of a complex-valued ONN (ONN-3). (a) Optical encoding and complex-valued measurement scheme. We encode positive vectors on the DMD, and complex-valued weights along with phase-only reference on the LC-SLM. At the output, we measure intensities and reconstruct the complex values digitally. (b) Network architecture of ONN-3 and the equivalent real-valued ONN. (c) Learning curve of the hybrid-trained ONN-3 compared with DENN-3 with the same network architecture. (d) Confusion matrix of the test set for ONN-3.

We see that this is equivalent to a two-layer real-valued ONN with square activation at the hidden layer, followed by a weight matrix with fixed values of zero and one, connecting each output neuron to exactly two hidden neurons. This equivalence is depicted in Fig. 4(b).

As previously stated, our optical multiplier naturally supports both real-valued and complex-valued operations. We can therefore build a complex-valued ONN (ONN-3) with stronger learning capabilities. A complex-valued ONN was recently built on a photonic integrated circuit, but only with a few neurons per layer

[34]. In comparison, our network size is an order of magnitude larger.

Even though the ONN-3 output is a set of intensities, complex-valued output neuron amplitudes are required for calculation of the weight matrix update. To measure these amplitudes, we change the relative phase ϕ between the LO field and the signal, as shown in Fig. 4(a). The real parts are extracted by setting $\phi = 0$ and $\phi = \pi$ and subtracting the measured intensities, and the imaginary parts are extracted similarly by setting $\phi = \pi/2$ and $\phi = 3\pi/2$. Instead of setting these four phases sequentially, we apply four spatially separated reference blocks of different phases for each output unit. That is to say, each output unit is spatially split into four regions yielding the four required interference results. Therefore, the real and imaginary parts of the output can be read out from one single camera frame. Upon readout, the modulus square activation is computed digitally. To complete the hybrid training, digital error backpropagation can be run with the ONN modeled as the equivalent two-layer real-valued network.

The learning curve in Fig. 4(c) converges quickly, with the highest classification accuracy of 93.6% achieved with the validation set, and 92.6% with the test set. We can also carry out *in silico* training and perform intensity measurement directly, by simply switching off the LO field. In this case, we obtain a classification accuracy of 92.2% with the test set.

E. Impact of Noise

We have demonstrated the efficacy of our hybrid training scheme with three different types of ONNs, as summarized in Table 1. We now use ONN-1 to explore the performance of hybrid training compared to traditional *in silico* training in different noisy environments.

To systematically compare different types of noise, we start with a well-controlled, low-noise environment. After carefully calibrating our system and performing the *in silico* training, ONN-1 reaches 87.7% classification accuracy, nearly the same as that of the hybrid training. This indicates that we have eliminated most aberrations and systematic errors, which is consistent with our small RMSE of the optical multiplier.

In the comparative study, we introduce different imperfections to the optical setup via the LC-SLM, and the results are shown in Fig. 5. The first imperfection is static additive noise: a random bias

Table 1. List of ONNs Demonstrated in This Work

ONN Type	Nonlinearity	Network Size	Hybrid Training Accuracy	DENN Accuracy
Optical linear classifier (ONN-1)	None	100×10	88.0%	89.9% (91.8%)
Hybrid opto-electronic network (ONN-2)	ReLU	$100 \times 25 \times 10$	92.7%	93.2% (95.7%)
Complex-valued ONN (ONN-3)	Modulus square	100×10	92.6%	93.8% (94.8%)

$w_{ji} \rightarrow w_{ji} + \epsilon_{ji}$, $\epsilon \in N(0, \sigma)$ applied to each weight matrix element, which remains unchanged during the training and testing. This can arise from ambient light, imprecise device calibration, etc. In our experiment, we randomly sample the bias that is fixed during the entire training and testing process. As seen from Fig. 5(a), hybrid training is robust to such static additive noise, while the accuracy of *in silico* training drops to 72.8% at 20% noise level ($\sigma = 0.2$).

A second common imperfection is static multiplicative noise $w_{ji} \rightarrow w_{ji} \times \eta_{ji}$, $\eta \in N(1, \sigma)$. This may be caused by non-uniform transmission of different optical channels, imperfect interference, different responses of photodetectors, etc. From Fig. 5, we see that hybrid training is also robust against such noise, while the performance of *in silico* training degrades to 80.6% at 50% noise level.

The last major type of imperfection is dynamic noise, which fluctuates over time. This may arise from imprecise device calibration, environmental fluctuation, etc. In the experiment, we model dynamic noise by additive noise (as defined above) applied to each weight element, which is randomly re-sampled at each weight update. Our results show that the ONN trained in either the hybrid or *in silico* scheme is sensitive to such random dynamic noise, and the accuracy drops to about 69% at 30% noise level, i.e., at a level similar to an equivalent DENN. In contrast to static errors, the cancellation of dynamic noise by hybrid training is not possible because the errors change from one iteration to the next. We believe that the difference in the ONN and DENN performance that we see in Table 1 arises due to dynamic noise-like effects in the ONN. Specifically, the SLM pixel cross talk and diffraction

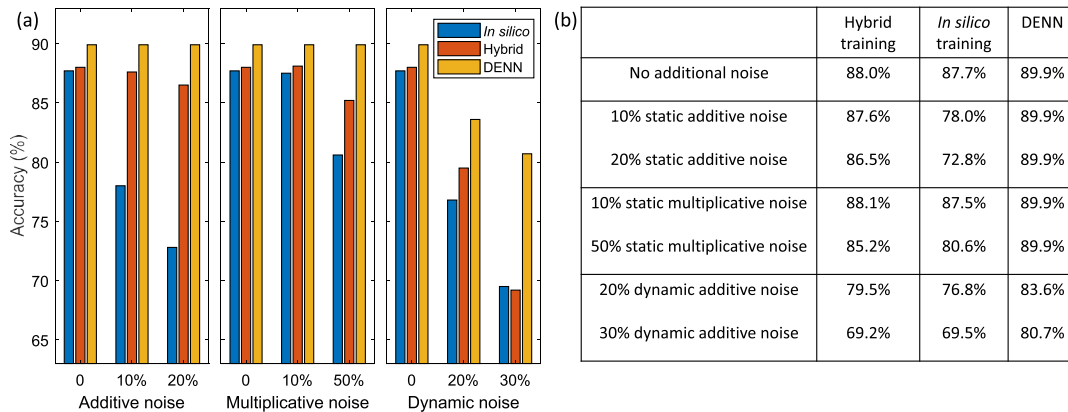


Fig. 5. Impact of noise during network training. (a) Comparison of hybrid training and *in silico* training of ONNs and DENN in presence of experimental imperfection. Left, middle, and right panels show effects of static additive noise, static multiplicative noise, and dynamic additive noise, respectively. DENN accuracy shown in brackets is obtained after training for twice the number of iterations. (b) List of accuracy at different noise levels. The noise level is defined as the value of σ as defined in the text.

result in deviation of the effective weight matrix elements from the desired values, with these deviations being different in every iteration. To mitigate these effects, every 50 iterations, we re-calibrate the system by running a few MVM examples with known outputs and adjust the parameters that map the camera gray level to MVM output. In the future, to increase the accuracy of our ONNs and extend our method to more complicated networks, more elaborate error correction methods would be required.

F. Optical Calculation of the Error Vector

There is a long-standing challenge to implement all-optical ONN training [35], such that the error terms are calculated and back-propagated optically. Our system can be modified to demonstrate an important step towards realizing this goal. To this end, we replace the categorical cross-entropy loss function in ONN-1 with the mean-squared error (MSE) loss function:

$$\mathcal{L} = \frac{1}{2} \left(z_j^{(1)} - y_j \right)^2, \quad (5)$$

where $z_j^{(1)}$ is the output of the linear network, and y_j is the corresponding one-hot encoded label. With this network architecture, the error vector to be calculated is

$$\delta_j = z_j^{(1)} - y_j. \quad (6)$$

This is easy to implement optically by destructive interference between the ONN output and an optically encoded label, and the schematic is shown in Fig. 6. We introduce a third active region on both the DMD and LC-SLM, which we call the label region. During hybrid training, in addition to encoding the input vector, we encode the target label on the DMD, and the label region on the LC-SLM is set to maximum reflection, with a π phase shift relative to the reference region. After passing through the cylindrical lens,

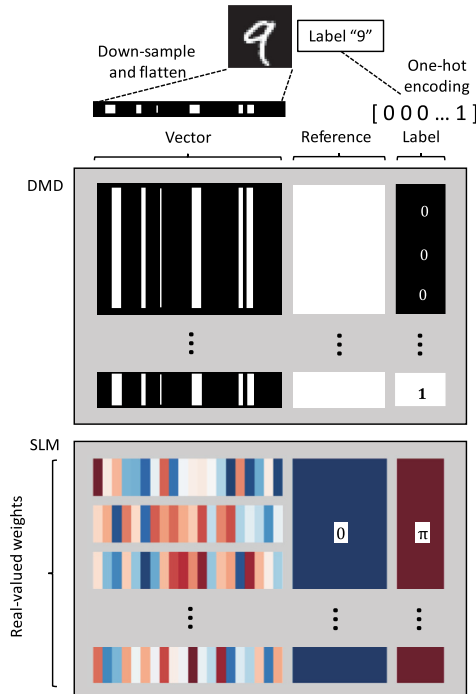


Fig. 6. Encoding scheme for optical calculation of the error vector in ONN-1. The destructive interference between the label region and signal region yields the optical calculation of the error vector when we use MSE loss function.

the label destructively interferes with both the signal and LO fields at the output plane. Intensity measurement therefore directly yields the error term [Eq. (6)]. This optically calculated error is then processed digitally to update the weights.

In this hybrid training setting, we achieve a peak validation accuracy of 83.3%. We then perform inference on the test set, by switching off the label region and measuring only the network output, and achieve an accuracy of 83.4%. By comparison, simulating this ONN with a DENN of the same architecture and MSE loss function, yields a test accuracy of 83.6%. The lower accuracy as compared to that of ONN-1 or DENN-1 is because the MSE loss function is not well suited to classification tasks. We can expect better performance with optical calculation of the error vector when solving regression problems where MSE is more suitable.

3. DISCUSSION

Our ONNs support up to 100 neurons per layer, limited by the resolution and the chosen grating period of the LC-SLM, as well as our matrix encoding method. Our LC-SLM model has a resolution of 1440×1050 , of which we use 1140×1050 pixels as the signal and 300×1050 pixels as the reference region. We use a diagonal blazed grating with horizontal and vertical periods of 10 pixels, so our maximum matrix size is about 110×100 . We found the precision of our MVM to be largely independent of the input vector dimension within these limits. A larger network size can be achieved by using LC-SLM models with higher resolution and reducing the grating period. Up to 1000 neurons per layer should be within reach.

With our hybrid training scheme, we reach similar classification accuracy as that of DENNs, and even higher accuracy can be achieved with longer training times. As an illustration, the three DENNs trained with twice the number of iterations converge at 91.8%, 95.7%, and 94.8%, as shown in Table 1. Therefore, future ONNs with long-term system stability can be expected to improve performance. A second important limiting factor of classification accuracy is the dynamic noise and imperfect device calibration that leads to weight-dependent fluctuations, such as SLM pixel cross talk. These imperfections cannot be corrected during training, and need to be suppressed to improve network performance. A final limitation is that the weights are constrained in passive ONNs. In our system, we chose to clip any weights above the bound, and although this leads to a reduced search space during training, it did not significantly affect the network performance in this work.

During the hybrid training, our computation speed is limited by the frame rates of the DMD, LC-SLM, and camera. In our chosen mode of operation, the DMD works at 1440 Hz frame rate in the binary mode, while our camera works at a maximum frame rate of 1480 Hz. The LC-SLM needs to update only once per mini-batch, i.e., at 6 Hz for a mini-batch size of 240 images. Therefore, its maximum operational refresh rate of 60 Hz supports up to a 14.4 kHz DMD frame rate with this mini-batch size. Therefore, our system frame rate is limited by the DMD at 1440 Hz, and the computation speed is $1440 \times 100 \times 25 \times 2 = 7.2 \times 10^6$ operations per second (where 100 and 25 are the first ONN layer dimensions). Today's advanced DMD and LC-SLM models support a maximum frame rate of up to 20 kHz and 1 kHz, respectively, and one can replace the camera by an ultra-fast photo-detector array. Therefore, the system frame rate can be increased by at least 10 times. Assuming a two-layer ONN with 1000 neurons per layer that updates at 20 kHz, the computation speed would be

4×10^{10} operations per second. Although ONNs with similar or even higher computing rates have been demonstrated [10–12], these demonstrations are limited to convolutional architectures. In contrast, our ONNs are fully connected, can be readily scaled further, and are the only ones capable of rapid update.

Instead of using a DMD as the input modulator, one can also use an electro-optical modulator array with a bandwidth that can exceed 10 GHz [36,37]. Such a two-layer ONN with 100 input neurons and 1000 neurons in subsequent layers would reach a computation speed of 2×10^{15} operations per second with a power efficiency that is a fraction of the same computational power provided by a cluster of GPUs.

Our work shows that analog systems with limited signal-to-noise ratios can still be physically trained to reach high performance, and this is a crucial step towards the more advanced goal of all-optical training of neural networks. We demonstrate a further step towards this goal by modifying our ONN to allow optical calculation of the error vector. The final remaining challenge is optical backpropagation in a deeper network with optical nonlinearity.

Funding. Engineering and Physical Sciences Research Council (IAA); Royal Commission for the Exhibition of 1851.

Acknowledgment. This work is supported by an EPSRC IAA award. X.G. acknowledges support from the Royal Commission for the Exhibition of 1851 Research Fellowship.

Author contributions. J.S. and X.G. carried out the experiment and performed the data analysis. All authors jointly prepared the paper. This work was done under the supervision of A.L.

Disclosures. J.S., X.G., and A.L. are inventors of patent application GB 2203480.5 on the method and apparatus for hybrid training of optical neural networks.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

[†]These authors contributed equally to this paper.

REFERENCES

1. D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," arXiv preprint arXiv:2104.10350 (2021).
2. Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**, 441–446 (2017).
3. Y. S. Abu-Mostafa and D. Psaltis, "Optical neural computers," *Sci. Am.* **256**, 88–95 (1987).
4. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the hopfield model," *Appl. Opt.* **24**, 1469–1475 (1985).
5. X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science* **361**, 1004–1008 (2018).
6. T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nat. Photonics* **15**, 367–373 (2021).
7. J. Li, D. Meng, N. T. Yardimci, Y. Luo, X. Li, M. Veli, Y. Rivenson, M. Jarrahi, and A. Ozcan, "Spectrally encoded single-pixel machine vision using diffractive networks," *Sci. Adv.* **7**, eabd7690 (2021).
8. A. A. Cruz-Cabrera, M. Yang, G. Cui, E. C. Behrman, J. E. Steck, and S. R. Skinner, "Reinforcement and backpropagation training for an optical neural network using self-lensing effects," *IEEE Trans. Neural Netw.* **11**, 1450–1457 (2000).
9. Y. Zuo, B. Li, Y. Zhao, Y. Jiang, Y.-C. Chen, P. Chen, G.-B. Jo, J. Liu, and S. Du, "All-optical neural network with nonlinear activation functions," *Optica* **6**, 1132–1137 (2019).
10. X. Xu, M. Tan, B. Corcoran, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, D. G. Hicks, R. Morandotti, A. Mitchell, and D. J. Moss, "11 tops photonic convolutional accelerator for optical neural networks," *Nature* **589**, 44–51 (2021).
11. J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, and H. Bhaskaran, "Parallel convolutional processing using an integrated photonic tensor core," *Nature* **589**, 52–58 (2021).
12. M. Miscuglio, Z. Hu, S. Li, J. K. George, R. Capanna, H. Dalir, P. M. Bardet, P. Gupta, and V. J. Sorger, "Massively parallel amplitude-only Fourier neural network," *Optica* **7**, 1812–1819 (2020).
13. J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, "Reinforcement learning in a large-scale photonic recurrent neural network," *Optica* **5**, 756–760 (2018).
14. L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," *Opt. Express* **20**, 3241–3249 (2012).
15. T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. Richard, and P. L. McMahon, "An optical neural network using less than 1 photon per multiplication," *Nat. Commun.* **13**, 123 (2022).
16. G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. Miller, and D. Psaltis, "Inference in artificial intelligence with deep optics and photonics," *Nature* **588**, 39–47 (2020).
17. B. J. Shastri, A. N. Tait, T. F. de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nat. Photonics* **15**, 102–114 (2021).
18. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**, 436–444 (2015).
19. L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, "Deep physical neural networks trained with backpropagation," *Nature* **601**, 549–555 (2022).
20. D. A. Miller, "Perfect optics with imperfect components," *Optica* **2**, 747–750 (2015).
21. S. Bandyopadhyay, R. Hamerly, and D. Englund, "Hardware error correction for programmable photonics," *Optica* **8**, 1247–1255 (2021).
22. D. A. Miller, "Setting up meshes of interferometers—reversed local light interference method," *Opt. Express* **25**, 29233–29248 (2017).
23. S. Pai, I. A. Williamson, T. W. Hughes, M. Minkov, O. Solgaard, S. Fan, and D. A. Miller, "Parallel programming of an arbitrary feedforward photonic network," *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–13 (2020).
24. B. Cramer, S. Billaudelle, S. Kanya, A. Leibfried, A. Grübl, V. Karasenko, C. Pehle, K. Schreiber, Y. Stradmann, J. Weis, J. Schemmel, and F. Zenke, "Surrogate gradients for analog neuromorphic computing," *Proc. Natl. Acad. Sci. USA* **119**, e2109194119 (2022).
25. R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik* **35**, 237–246 (1972).
26. X. Guo, T. D. Barrett, Z. M. Wang, and A. Lvovsky, "Backpropagation through nonlinear units for the all-optical training of neural networks," *Photon. Res.* **9**, B71–B80 (2021).
27. S. M. Tam, B. Gupta, H. A. Castro, and M. Holler, "Learning on an analog VLSI neural network chip," in *IEEE International Conference on Systems, Man, and Cybernetics Conference* (IEEE, 1990), pp. 701–703.
28. L. Mennel, J. Symonowicz, S. Wachter, D. K. Polyushkin, A. J. Molina-Mendoza, and T. Mueller, "Ultrafast machine vision with 2D material neural network image sensors," *Nature* **579**, 62–66 (2020).
29. C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nat. Commun.* **9**, 2385 (2018).
30. Z. Wang, C. Li, P. Lin, M. Rao, Y. Nie, W. Song, Q. Qiu, Y. Li, P. Yan, J. P. Strachan, N. Ge, N. McDonald, Q. Wu, M. Hu, H. Wu, R. S. Williams, Q. Xia, and J. J. Yang, "In situ training of feed-forward and recurrent convolutional memristor networks," *Nat. Mach. Intell.* **1**, 434–442 (2019).
31. J. Spall, X. Guo, T. D. Barrett, and A. Lvovsky, "Fully reconfigurable coherent optical vector–matrix multiplication," *Opt. Lett.* **45**, 5752–5755 (2020).

32. V. Arrizón, U. Ruiz, R. Carrada, and L. A. González, "Pixelated phase computer holograms for the accurate encoding of scalar complex fields," *J. Opt. Soc. Am. A* **24**, 3500–3507 (2007).
33. W.-H. Lee, "Binary computer-generated holograms," *Appl. Opt.* **18**, 3661–3669 (1979).
34. H. Zhang, M. Gu, X. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. Yung, Y. Z. Shi, F. K. Muhammad, G. Q. Lo, X. S. Luo, B. Dong, D. L. Kwong, L. C. Kwek, and A. Q. Liu, "An optical neural chip for implementing complex-valued neural network," *Nat. Commun.* **12**, 457 (2021).
35. D. Brunner and D. Psaltis, "Competitive photonic neural networks," *Nat. Photonics* **15**, 323–324 (2021).
36. D. A. Miller, "Attojoule optoelectronics for low-energy information processing and communications," *J. Lightwave Technol.* **35**, 346–396 (2017).
37. R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-scale optical neural networks based on photoelectric multiplication," *Phys. Rev. X* **9**, 021032 (2019).