nature photonics

Article

https://doi.org/10.1038/s41566-024-01494-z

Nonlinear processing with linear optics

Received: 16 August 2023

Accepted: 1 July 2024

Published online: 31 July 2024

Check for updates

Mustafa Yildirim [1,2,3], Niyazi Ulas Dinc [1,2,3], Ilker Oguz [1,2], Demetri Psaltis [2] & Christophe Moser [1]

Deep neural networks have achieved remarkable breakthroughs by leveraging multiple layers of data processing to extract hidden representations, albeit at the cost of large electronic computing power. To enhance energy efficiency and speed, the optical implementation of neural networks aims to harness the advantages of optical bandwidth and the energy efficiency of optical interconnections. In the absence of low-power optical nonlinearities, the challenge in the implementation of multilayer optical networks lies in realizing multiple optical layers without resorting to electronic components. Here we present a novel framework that uses multiple scattering, and which is capable of synthesizing programmable linear and nonlinear transformations concurrently at low optical power by leveraging the nonlinear relationship between the scattering potential, represented by data, and the scattered field. Theoretical and experimental investigations show that repeating the data by multiple scattering enables nonlinear optical computing with low-power continuous-wave light. Moreover, we empirically find that scaling of this optical framework follows a power law.

Optical computing has re-emerged as an alternative to electronics for performing computations and handling information, particularly in the context of artificial intelligence applications. Optical neural networks (ONNs) hold promise in terms of speed and energy efficiency compared with traditional electronic computing[1]. However, the development of fully optical ONNs has proven to be a challenging task due to the need for incorporating both linear and nonlinear computations within the optical domain[2]. Although several approaches have demonstrated efficient optical computing hardware for linear calculations[3–8], effectively integrating these capabilities with nonlinear computations remains a substantial obstacle to the complete realization of ONNs. Researchers have explored nonlinear light–matter interactions in the context of reservoir computing[9–11], employing high-intensity pulsed lasers for nonlinear data processing[12–15]. Additionally, the complex dynamics observed in multimode laser cavities interacting with external optical signals are employed for low-power nonlinear transformations in reservoir computing[16]. Platforms like the integrated meshes of Mach–Zehnder interferometers[3], diffractive neural networks[6,17,18], micro-ring resonators[7,19] and free-space linear systems[5,8,20,21] have facilitated linear calculations. However, nonlinear computations have relied on optoelectronic nonlinearity or electronic computation, resulting in limitations such as non-programmable optoelectronic nonlinearity and high energy consumption. Accordingly, there is a need to find a low-power flexible solution to implement programmable nonlinear operations in the optical domain to fully harness the low-power computing potential inherently offered by linear optics.

It has been shown previously that a purely linear transformation can be implemented with a stack of two-dimensional (2D) diffractive layers[22]. The Ozcan group has applied this approach to ONNs employing additive manufacturing techniques[6,18]. These deep learning-enabled multilayer diffractive processors enable computation by facilitating the propagation of free-space light through a sequence of structured passive scattering surfaces. This optical processing technique leverages the 3D connectivity between nodes in consecutive layers, achieved via diffraction, thereby providing a path to scalability[23]. However, one limitation of this approach is that the nonlinearity is limited to square-law detection at the output, which limits the realization of complex ONNs.

[1]Laboratory of Applied Photonics Devices, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. [2]Optics Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. [3]These authors contributed equally: Mustafa Yildirim, Niyazi Ulas Dinc. e-mail: mustafa.yildirim@epfl.ch; niyazi.dinc@epfl.ch

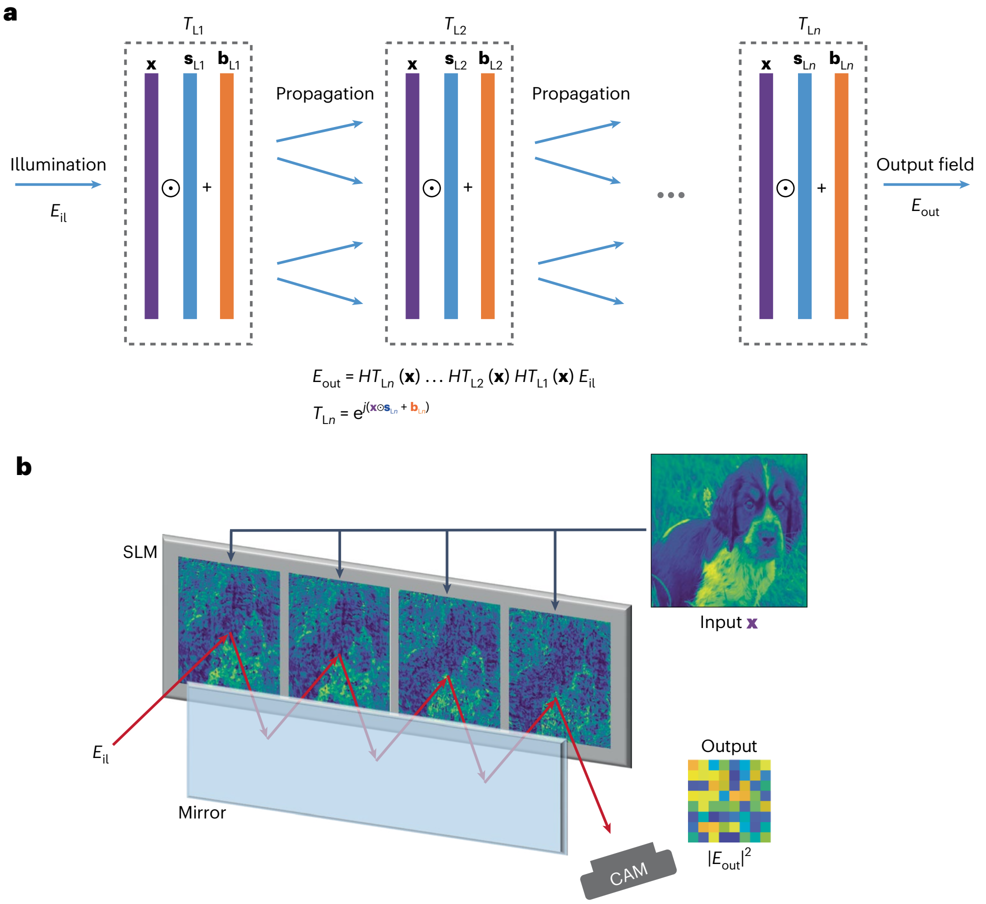

Fig. 1 | The nPOLO framework. a, Computation scheme, showing multiple modulation layers (T_{Ln}) within the framework. The data ($\mathbf{x}$) are presented on these layers, accompanied by trainable scaling ($\mathbf{s}_{Ln}$) and bias ($\mathbf{b}_{Ln}$) parameters to optimize the transformation. Propagation is represented by H (Toeplitz matrix). **b**, The physical implementation of nPOLO, featuring a single-pass multi-bounce cavity configuration. This implementation consists of a SLM and a mirror, positioned in a way that enables the realization of consecutive layers on the SLM side by side. The propagation distance is determined by the reflection from the mirror, allowing the light to propagate between the layers. Output light is captured with a camera. CAM, camera.

Another avenue that can be explored is the relationship between the scattering potential and the scattered light. At low intensity levels, the propagation of light through a scattering medium exhibits linearity in terms of the relation between the input and output light field, but the output light can have a nonlinear dependence on the data encoded in the scattering potential. This form of nonlinearity is referred to as structural nonlinearity, and it has been investigated by Eliezer and colleagues using multiple scatterings within an integrating sphere[24].

In this Article we present a programmable framework called nonlinear processing with only linear optics (nPOLO) for the all-optical realization of neural networks using a low-power continuous-wave laser and diffractive layers. The nPOLO framework enables simultaneous linear and nonlinear operations within the optical domain. In this way, nPOLO unifies multilayer light modulation and structural nonlinearity such that the collective impact of data-modulated layers on propagating light generates a high-order nonlinear transformation of the data. The data are repetitively embedded into the modulation layers, combined with trainable parameters that enable the desired relationship (linear and nonlinear) between the data and the output field. Our results demonstrate that increasing the number of layers and data repetitions leads to the generation of higher-order nonlinearities, such as polynomial orders, which include cross-terms among the different elements of the input data. To illustrate the effectiveness of data repetition, we conducted a comparative analysis of the performance obtained between repeating the data in each modulation layer and presenting the data only once. Our results demonstrate that, when both systems have an equal number of degrees of freedom in terms of the displayed pixels in the modulation layers, the data repetition approach consistently achieves higher accuracy scores and exhibits improved robustness against experimental imperfections and simulated noise. Overall, our findings showcase the ability of the nPOLO framework to synthesize a learnable both linear and nonlinear data transform in a hybrid optical–digital neural network using only low-power continuous-wave light.

nPOLO framework

The core of the nPOLO technique involves utilizing multiple data planes that are evenly spaced. The physical implementation of nPOLO includes a liquid-crystal spatial light modulator (SLM) and a mirror positioned opposite it[25], allowing the simultaneous display of multiple modulation planes on a single SLM device. This configuration forms a multi-bounce single-pass cavity, where each plane serves as a reflecting surface that modulates the phase of light as it propagates from one plane to the next. Figure 1 provides an unfolded representation of the multi-bounce nPOLO architecture. We can write the output field as a function of the modulation layers as follows:

$$E_{\mathrm{out}} = HT_{\mathrm{L}n}(\mathbf{x}) \ldots HT_{\mathrm{L2}}(\mathbf{x})HT_{\mathrm{L1}}(\mathbf{x})E_{\mathrm{il}} \qquad (1)$$

where E_{il} is the illumination beam, $T_{\mathrm{L}n}$ is the transmittance of the nth modulation layer, and the modulation layer number goes up to N. H is the diffraction operator, which is the same for all the layers, as they are equally spaced. Without loss of generality, let us assume that we have an optical system that has an input aperture and output aperture that are sampled by a total number of K pixels. The input and output electric fields then become $K \times 1$ vectors. We can write the incident field as a vector of ones, assuming the incident optical field is a plane wave propagating along the optical axis. Assuming the same number of pixels in the modulation layer as in the input/output apertures, $T_{\mathrm{L}n}$ becomes a diagonal matrix with size $K \times K$. H is a $K \times K$ Toeplitz matrix that represents diffraction for an arbitrary distance[23]. If one directly inserts the data into $T_{\mathrm{L}n}$, there is no trainable parameter to tune the transform. Thus, instead of direct use of the data in $T_{\mathrm{L}n}$, we feed a linearly transformed version of it in the form of $t_j^{(n)} = s_j^{(n)} x_j^{(n)} + b_j^{(n)}$ to introduce trainable parameters, where $s_j^{(n)}$ is a scaling parameter, $x_j^{(n)}$ is an element of the data vector, and $b_j^{(n)}$ is a bias parameter where j refers to indexing of the vectors and n is the layer index. When we expand equation (1) for the general case of complex modulation with these parameters, we obtain

$$\begin{aligned} o_i = \sum_{j=1}^{K} h_{ij}\left(s_j^{(N)} x_j + b_j^{(N)}\right)\Bigg(\ldots \sum_{j=1}^{K} h_{ij}\left(s_j^{(2)} x_j + b_j^{(2)}\right) \\ \left(\sum_{j=1}^{K} h_{ij}\left(s_j^{(1)} x_j + b_j^{(1)}\right)\right)\ldots\Bigg) = \sum_{m=0}^{N} \alpha_{i,m}\left(\mathbf{s},\mathbf{b}\right)\mathbf{x}^m \end{aligned} \qquad (2)$$

where h_{ij} is the element of the diffraction matrix. As seen in the left-hand side of equation (2), multiplication by $\mathbf{x}$ at each layer gives rise to polynomial orders. We can then write the overall transform in a generic polynomial form up to order N where coefficients ($\alpha_{i,m}$) are functions of $\mathbf{s}$ and $\mathbf{b}$, the right-hand side of equation (2).

Now, we look at how the input is transformed by the nonlinear activation function applied to the linear transformation of $\mathbf{x}$ for a perceptron in a digital neural network:

$$o_i = g\left(\sum_{j=1}^{K} w_{ij} x_j + b_i\right) = \sum_{m} \beta_{i,m}\left(\mathbf{w},\mathbf{b}\right)\mathbf{x}^m \qquad (3)$$

In equation (3), w_{ij} and b_i are the trainable weight and bias parameters of the perceptron, and g is an algorithmic nonlinear function of choice (sigmoid, ReLU and so on). The nonlinear activation function can be expressed as a polynomial expansion as well, where the coefficients ($\beta_{i,m}$) depend on $\mathbf{w}$ and $\mathbf{b}$. In a multilayer perceptron (MLP), the output is a cascade of such polynomials, which is also a polynomial. Therefore, the functional form of equation (3) applies to an MLP as well. Equations (2) and (3) have a similar polynomial form except with different coefficients. Ideally, if the nPOLO and the MLP are both trained perfectly to implement the same function, then the coefficients of the two polynomial expansions should converge. In Supplementary Section 1 we present fully developed output vector $\mathbf{o}$ in terms of explicit polynomial orders.

Trainable parameters, in the form of scaling and bias, are applied to each pixel value of the data presented on the modulation layers. These parameters are trained digitally via a computer model (Methods). Once the desired nonlinear transformation is achieved on the computer, the parameters are applied to the multiple layers (adjacent planes on the SLM) in the experimental set-up, resulting in an intensity pattern that is recorded by a camera. Subsequently, a compact representation of the recorded camera pattern is obtained through average pooling, resulting in a 2D matrix of values such as a 4-by-4 or an 8-by-8 grid. This compact representation is then fed into a digital linear classifier, which processes the data via a single fully connected linear layer, thereby producing the final classification results.

Results

By increasing the number of layers—that is, adjacent planes on the SLM—one can assess the impact of the polynomial orders resulting from structural nonlinearity. However, the increase in the number of layers also leads to an increase in the system's degrees of freedom and space–bandwidth product. We thus devised an alternative comparison experiment to mitigate these effects. In our experiments, we maintained the same number of layers and pixels but modified the data allocation. Specifically, we initially incorporated the data only in the first layer, and the subsequent layers consisted exclusively of trainable bias parameters without any data or scaling parameters. To evaluate the performance of the nPOLO framework, we conducted experiments using the Imagenette, Fashion MNIST and Digit MNIST datasets[26–28]. The obtained numerical and experimental results are provided in Fig. 2.

We used 300 × 300 pixels on the SLM for each modulation layer. The original Fashion and Digit MNIST dataset samples contain 28 × 28-pixel images and the Imagenette dataset samples 320 × 320-pixel images. For the Fashion and Digit MNIST datasets we used 4 × 4 superpixels on the SLM, yielding a 75 × 75 grid for assigning trainable parameters, yielding 11,250 parameters (scaling and bias) per layer. We linearly upsampled the images of those datasets to 75 × 75. For the Imagenette dataset we linearly downsampled the images to 300 × 300 and used all the pixels for assigning trainable parameters, yielding 180,000 parameters (scaling and bias) per layer.

For clarity, we provide examples of the displayed masks in Fig. 3 in 'with data repeat' and 'without data repeat' configurations using an example from the Fashion MNIST and Imagenette datasets with four modulation layers. Supplementary Section 2 presents a comparative depiction of parameter allocation. The trainable parameters were optimized by computer simulation, where the physical light propagation was modelled using the beam propagation method (BPM).

Because BPM consists of differentiable calculation steps, the error can be backpropagated to the trainable parameters, and they are optimized using stochastic gradient descent. In Fig. 4 we present the training scheme used, in which the digital model of the optical system and the digital classifier were co-trained for the classification tasks. By following this co-training approach, we obtained scaling and bias masks for different layer configurations, ranging from layer $N = 1$ to layer $N = 4$, both with and without data repetition. It is important to note that in the case of a single layer ($N = 1$), both data repetition options are equivalent, as we only had a single layer available to introduce the data.

Our experimental findings in Fig. 2a–c consistently demonstrate that when data are repeated across multiple layers, we achieve higher classification accuracy compared with configurations without data repetition. Moreover, increasing the number of layers also contributes to improved accuracy. We also observed that when the number of layers is held constant, eliminating data repetition leads to a reduction in accuracy. These results highlight the contribution of higher-order optical nonlinearities generated via data repetition. We observed a similar trend in our simulations while calculating the trainable parameters (Fig. 2d–f). Both experimental and simulated results validate the contribution of the nPOLO framework. However, the experimental accuracies were lower than the simulated ones. Figure 2g–i illustrates the accuracy difference between simulations and experiments in all three datasets. Interestingly, the decrease in accuracy is less pronounced in cases involving data repetition. We attribute this discrepancy to imperfections between the simulated model and the physical implementation, primarily the non-ideal phase response and flickering of the SLM. For example, we were able to reduce the discrepancy between the optical experiment and the digital simulation to 2% by upgrading the SLM for the Imagenette database with the data repeat configuration (Supplementary Section 3). Such engineering aspects of the nPOLO framework will be further explored in a follow-up study. We also trained a simple convolutional neural network to assess how nPOLO compared with a fully digital counterpart for these tasks and obtained comparable performance (Supplementary Section 4).

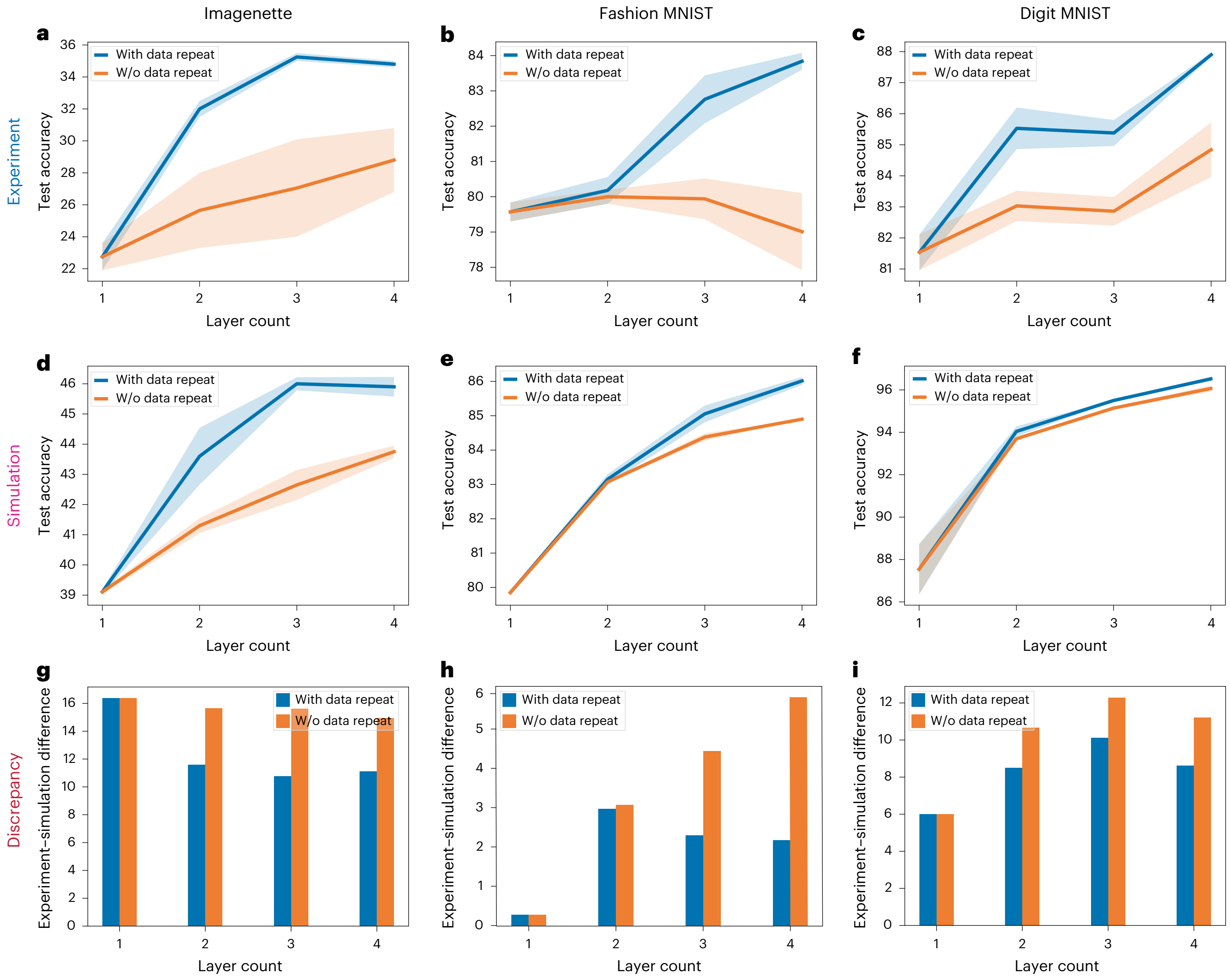

Fig. 2 | Classification accuracy results comparing two different schemes, one with data repeat and one without data repeat. **a**–**i**, Classification accuracy results obtained for the Imagenette (**a**,**d**,**g**), Fashion MNIST (**b**,**e**,**h**) and Digit MNIST (**c**,**f**,**i**) datasets, comparing two different schemes: one with data repeat and one without data repeat. The layer count (N) varies from 1 to 4 for both schemes. Each configuration is trained independently, resulting in layer masks that are applied to the SLM as phase masks. **a**–**c**, Experimentally obtained test accuracies for all datasets, representing the performance with and without structural nonlinearity. **d**–**f**, Test accuracies of the corresponding simulations, plotted for both schemes, as in experiments. The mean and s.d. values are obtained by testing the models trained from scratch. **g**–**i**, Mean accuracy difference between experimental and simulated results, shown as bar plots for varying layer number with and without data repeat. In plots **a**–**f**, solid lines are the mean results, and shaded regions represent ±1 s.d.

Scaling study

The performance is influenced by three factors: the number of trainable parameters, the dataset size and the available compute budget for training[29]. Although the dataset size remains constant in our study, the model size is subject to variation within the constraints of a limited computational budget. Specifically, we explored the impact of changing the model size on the performance of nPOLO, determined by the width, W^2, (the number of SLM pixels in the illuminated 2D patch) of its diffractive layers and the depth, N, denoting the number of diffractive layers. Our empirical investigation involved varying W and N for classification tasks of the Imagenette and Fashion MNIST datasets (Methods). The findings are summarized in Fig. 5. The parameter count of our model is $2NW^2$. The factor of 2 is included because of the two trainable parameters per pixel. Initially, we observe a continuous upward trend in performance as the parameter count increases, regardless of N and W. In other words, configurations with different depths perform similarly for the same parameter count, up to a few hundred thousand, as shown in Fig. 5a,c. However, further increase in the parameter count leads to performance saturation. This phenomenon is also observed in conventional digital neural networks. Deeper implementations of nPOLO ($N > 1$) extend the onset of this saturation point. Recent studies at OpenAI have revealed that the scaling of deep MLP networks adheres to a power-law relationship between the test loss and the size of the model[29,30], extending across numerous layers and billions of parameters. The test loss of the simulated nPOLO is plotted in Fig. 5b,d against the number of parameter counts. This reveals a power-law scaling within the examined network depth, encompassing up to four layers and ~8.4 million parameters. Despite the orders of difference in parameter counts, the presence of power-law scaling in this range is encouraging for the development of future large-scale networks.

Discussion

We introduced trainable scaling and bias parameters for the pixel values of the data displayed on the SLM to synthesize a programmable

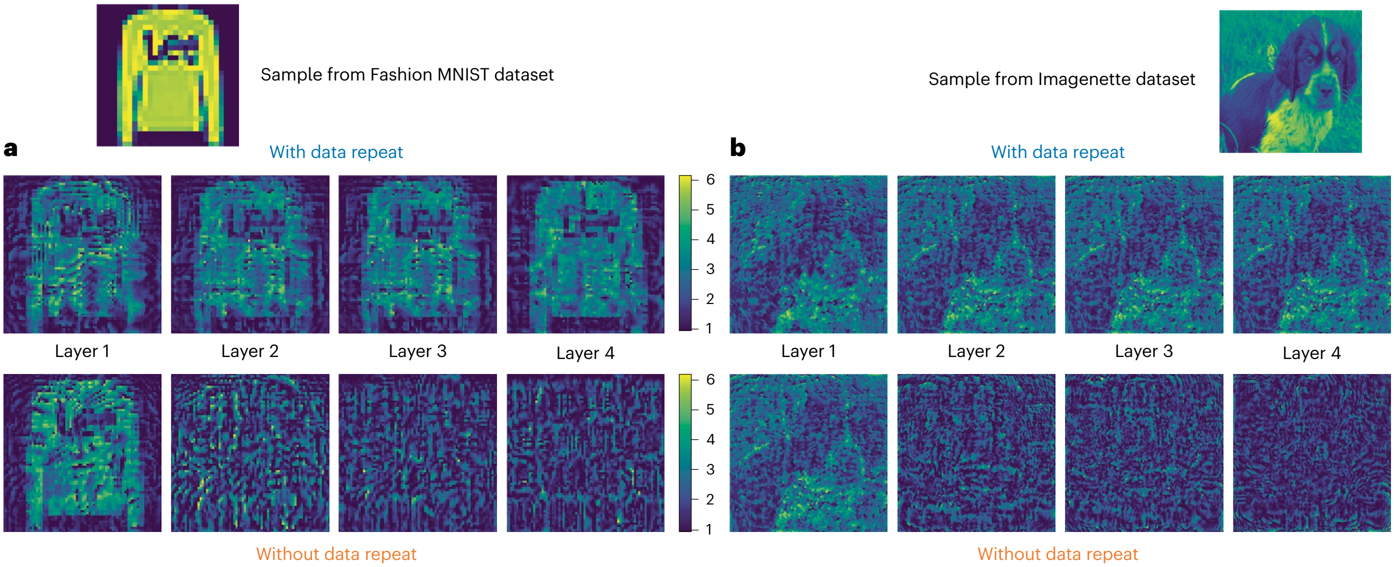

Fig. 3 | Samples of trained layer masks to be displayed on SLM for Fashion MNIST and Imagenette. a, Example sample from the Fashion MNIST dataset alongside its corresponding layer masks, combined with trainable parameters, according to both schemas. The colour code represents phase modulation ranging from 0 to 2π. **b**, Example sample from the Imagenette dataset, and its corresponding layer masks combined with trainable parameters for both schemas.

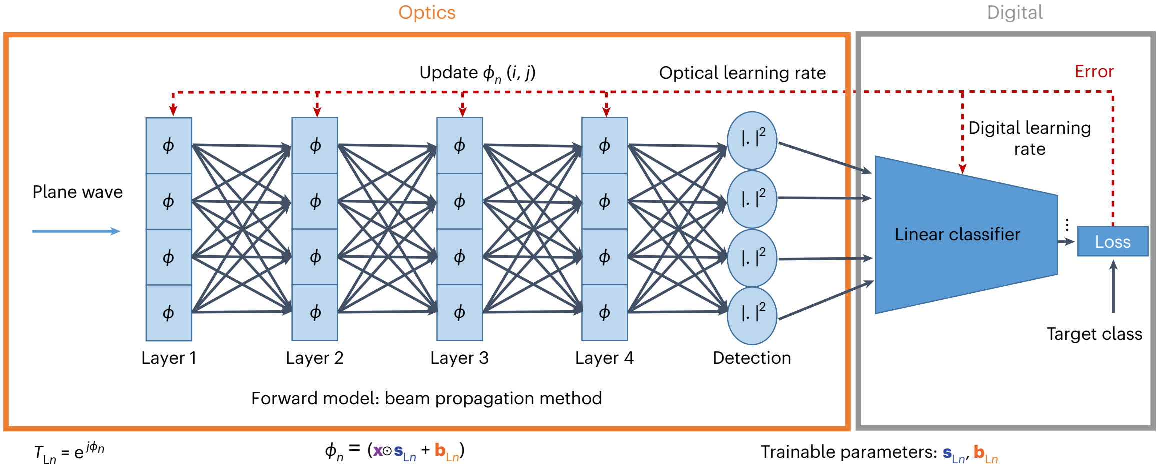

Fig. 4 | Co-training of optical trainable parameters and digital trainable parameters. Simulation model of the four-layer system, built using the beam propagation method, where layers are composed of pixels having phase ϕ, defined by trainable parameters ($\mathbf{s}_{Ln}$ and $\mathbf{b}_{Ln}$). Training of the optical layers and linear classifier is performed simultaneously using separate learning rates.

computation. We observed that the contribution of structural nonlinearity became more pronounced when dealing with more challenging classification tasks. The benefit of using repeated data versus not using repeated data was more pronounced for Imagenette than for Digit MNIST. This disparity arises because Digit MNIST represents an easier task, and the structural nonlinearity becomes redundant in the presence of the square-law detection nonlinearity.

The scaling study we performed revealed that as the model size of nPOLO increases the performance improves, and the optimal performance relies on a balanced scaling of width and depth of the model. Remarkably, similar properties are also found in digital deep neural networks[30,31]. Future studies could explore alternative approaches of employing trainable parameters or implementing different functional forms, such as using convolutional kernels or eigenmodes of the optical system as trainable parameters, similar to investigations conducted for fibre-based optical-learning machines[32]. During the final stage of this work, we became aware of an independent and different approach to perform passive optical nonlinearity, exploiting reflections inside a disordered cavity[33]. More recently, another article appeared investigating, through digital only simulations, structural nonlinearity for the implementation of neural networks[34].

An interesting observation is the increased robustness of data repetition across multiple layers against experimental imperfections. We initially noticed this phenomenon during experiments conducted on different datasets and layer configurations. To further investigate this, we introduced phase noise and misalignment in BPM simulations to emulate experimental imperfections while keeping the trained masks fixed (Supplementary Section 3). Gradually increasing the simulated noise level or misalignment degree, we observed that the configurations with data repetition exhibited greater robustness. The configuration without data repetition experienced a more rapid drop in accuracy, consistent with the experimental results. This finding strengthens the argument for the noise robustness of the data repetition scheme, as we have empirical data from both experiments and simulations.

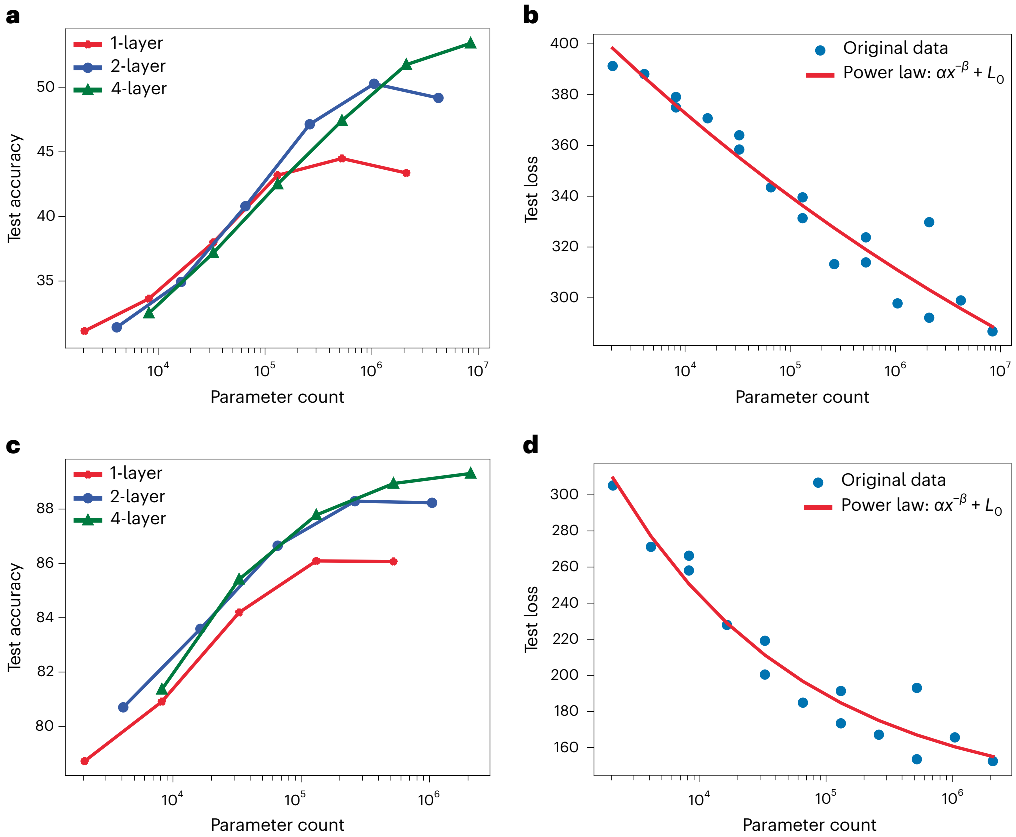

Fig. 5 | Scaling dynamics of nPOLO with increasing parameter count, showing test accuracies and losses for both the Imagenette and Fashion MNIST datasets across various parameter configurations. Manipulation of layer width and depth enables the realization of different parameter counts. **a**–**d**, Detailed breakdown of test accuracies for different layer counts (1, 2 and 4) individually (**a**,**c**) and all test results in relation to parameter count, irrespective of the specific layer number (**b**,**d**) for the Imagenette (**a**,**b**) and Fashion MNIST (**c**,**d**) datasets. Results (**b**,**d**) are fitted to a power function defined by parameters α, β and L_0, where x represents parameter count.

One possible explanation for this phenomenon is that by introducing the data multiple times, the model learns multiple paths from the input data to the output plane during training, resulting in not only higher polynomial orders but also cross-terms that couple with different optical paths and reach the detector plane. The existence of multiple routes for highlighting useful features in the output plane may make the data repetition scheme less susceptible to noise.

Our scaling study demonstrates a power-law scaling trend akin to the observations in OpenAI's deep neural network studies[29,30]. While exploring deeper models with more than four layers for improved performance, we found that increased depth did not yield substantial enhancements. We hit the dataset bottleneck, as we observe overfitting for the large number of parameters that deeper models employ. The increased deviations towards a high parameter count (Fig. 5) also exist for this reason. This aligns with the findings reported in refs. 29,30, whereby empirical performance individually exhibits a power-law relationship with three factors—model size, dataset and compute budget—when not constrained by the other two. In practical terms, pursuing deeper models might offer diminishing returns unless accompanied by adjustments in dataset size and the employment of greater computational resources. We note that the highest parameter count in the scaling study yielded ~4.2 million pixels for Imagenette and ~1 million for the Fashion and Digit MNIST datasets, which can be employed using commercially available devices. Other possibilities to further scale the number of parameters include using multiple SLMs or other structured media such as computer-generated holograms[35] or volume holograms[36,37], instead of the flat mirror, to accommodate additional parameters to boost performance.

Overall, the nPOLO framework presents a novel approach for generating optical nonlinearity using low-power optical devices, eliminating the need for electronic components to achieve higher orders of nonlinearity. Networks implemented with structural nonlinearities are not MLPs, but they can be trained to reach comparable performance and they have similar scaling laws. We also discovered that the introduction of data repetition to generate polynomial nonlinearities enhances robustness against noise. Note that this framework is applicable to the cascade of any optical linear system, including integrated-waveguide Mach–Zehnder interferometers[3]. These characteristics make nPOLO a promising platform for realizing ONNs.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41566-024-01494-z.

References

1. Denz, C. *Optical Neural Networks* (Springer, 2013).
2. Wetzstein, G. et al. Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39–47 (2020).
3. Shen, Y. et al. Deep learning with coherent nanophotonic circuits. *Nat. Photonics* **11**, 441–446 (2017).
4. Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. & Pernice, W. H. P. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569**, 208–214 (2019).

5. Farhat, N. H., Psaltis, D., Prata, A. & Paek, E. Optical implementation of the Hopfield model. *Appl. Opt.* **24**, 1469–1475 (1985).
6. Lin, X. et al. All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008 (2018).
7. Tait, A. N. et al. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* **7**, 7430 (2017).
8. Chang, J., Sitzmann, V., Dun, X., Heidrich, W. & Wetzstein, G. Hybrid optical–electronic convolutional neural networks with optimized diffractive optics for image classification. *Sci. Rep.* **8**, 12324 (2018).
9. Van der Sande, G., Brunner, D. & Soriano, M. C. Advances in photonic reservoir computing. *Nanophotonics* **6**, 561–576 (2017).
10. Gao, C., Gaur, P., Rubin, S. & Fainman, Y. Thin liquid film as an optical nonlinear-nonlocal medium and memory element in integrated optofluidic reservoir computer. *Adv. Photon.* **4**, 046005 (2022).
11. Marcucci, G., Pierangeli, D. & Conti, C. Theory of neuromorphic computing by waves: machine learning by rogue waves, dispersive shocks and solitons. *Phys. Rev. Lett.* **125**, 093901 (2020).
12. Teğin, U., Yıldırım, M., Oğuz, I., Moser, C. & Psaltis, D. Scalable optical learning operator. *Nat. Comput. Sci.* **18**, 542–549 (2021).
13. Wright, L. G. et al. Deep physical neural networks trained with backpropagation. *Nature* **601**, 549–555 (2022).
14. Zhou, T., Scalzo, F. & Jalali, B. Nonlinear Schrödinger kernel for hardware acceleration of machine learning. *J. Light. Technol.* **40**, 1308–1319 (2022).
15. Yildirim, M. et al. Nonlinear optical feature generator for machine learning. *APL Photonics* **8**, 106104 (2023).
16. Porte, X. et al. A complete, parallel and autonomous photonic neural network in a semiconductor multimode laser. *J. Phys. Photonics* **3**, 024017 (2021).
17. Zhou, T. et al. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nat. Photonics* **15**, 367–373 (2021).
18. Kulce, O., Deniz, M., Yair, R. & Aydogan, O. All-optical synthesis of an arbitrary linear transformation using diffractive surfaces. *Light Sci. Appl.* **10**, 196 (2021).
19. Huang, C. et al. Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits. *APL Photonics* **5**, 040803 (2020).
20. Spall, J., Guo, X., Barrett, T. D. & Lvovsky, A. I. Fully reconfigurable coherent optical vector-matrix multiplication. *Opt. Lett.* **45**, 5752–5755 (2020).
21. Wagner, K. H. & McComb, S. Optical rectifying linear units for back-propagation learning in a deep holographic convolutional neural network. *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–18 (2019).
22. Labroille, G. et al. Efficient and mode selective spatial mode multiplexer based on multi-plane light conversion. *Opt. Express* **22**, 15599–15607 (2014).
23. Dinc, N. U. et al. From 3D to 2D and back again. *Nanophotonics* **12**, 777–793 (2023).
24. Eliezer, Y., Rührmair, U., Wisiol, N., Bittner, S. & Cao, H. Tunable nonlinear optical mapping in a multiple-scattering cavity. *Proc. Natl Acad. Sci. USA* **120**, e2305027120 (2023).
25. Wang, H. & Piestun, R. Dynamic 2D implementation of 3D diffractive optics. *Optica* **5**, 1220–1228 (2018).
26. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Preprint at https://arxiv.org/abs/1708.07747 (2017).
27. Howard, J. & Gugger, S. fastai: a layered API for deep learning. *Information* **11**, 108 (2020).
28. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
29. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
30. Henighan, T. et al. Scaling laws for autoregressive generative modeling. Preprint at https://arxiv.org/abs/2010.14701 (2020).
31. Tolstikhin, I. O. et al. MLP-mixer: an all-MLP architecture for vision. In *Advances in Neural Information Processing Systems* Vol. 34 (eds Beygelzimer, A. et al.) 24261–24272 (NeurIPS, 2021).
32. Oguz, I. et al. Programming nonlinear propagation for efficient optical learning machines. *Adv. Photonics* **6**, 016002 (2024).
33. Xia, F. et al. Deep learning with passive optical nonlinear mapping. Preprint at https://arxiv.org/abs/2307.08558 (2024).
34. Wanjura, C. C. & Marquardt, F. Fully nonlinear neuromorphic computing with linear wave scattering. *Nat. Phys.* https://doi.org/10.1038/s41567-024-02534-9 (2024).
35. Dinc, N. U., Lim, J., Kakkava, E., Moser, C. & Psaltis, D. Computer generated optical volume elements by additive manufacturing. *Nanophotonics* **9**, 4173–4181 (2020).
36. Psaltis, D., Brady, D., Gu, X. G. & Lin, S. Holography in artificial neural networks. *Nature* **343**, 325–330 (1990).
37. Dinc, N. U., Moser, C. & Psaltis, D. Volume holograms with linear diffraction efficiency relation by (3+1)D printing. *Opt. Lett.* **49**, 322–325 (2024).

Methods

Digital training

Optimization methods have already been demonstrated to reconstruct 3D phase objects from experimental recordings of 2D projections[38,39]. In ref. 38, the forward model in the optimization is the BPM. The iterative error reduction scheme and multilayer structure of the BPM resembles a multilayer neural network. Accordingly, this method is referred to as 'learning tomography' (LT). We show that, instead of imaging an object, we can reconstruct the 3D structure that performs the desired task as defined by its input–output functionality[35]. To establish the target functionality, 3D phase modulation (either through a continuous medium or multiple planes) and the scattered field caused by the phase modulation must be modelled accurately. Unlike conventional reconstruction algorithms that rely on first-order approximations, LT incorporates higher-order scattering effects by employing BPM. The LT algorithm involves an iterative reconstruction process using the forward model, along with the constraints arising from experimental considerations such as the pixel pitch of the SLM. In this Article we have adapted this approach, presented in ref. 35 (where additional details can be found), to generate scaling and bias parameters for the demonstrated classification tasks. The output intensity pattern of the forward model is average pooled to yield a 4 × 4 matrix for each sample of the Fashion MNIST dataset and an 8 × 8 matrix for each sample of the Imagenette dataset. These matrices are flattened to act as an input layer of a digital classifier that has ten output neurons for each class of the datasets, without any hidden layer and nonlinear activation function. The trainable parameters employed in the BPM model and digital classifier are co-trained by a continuous error backpropagation where different learning rates are assigned to digital weights (10^{-4}) and optical scaling and bias parameters (10^{-3}), using categorical cross-entropy as the loss function. We used batch learning with a batch size of 20 and a random shuffle in every batch. PyTorch libraries were used for the whole training process.

For the scaling study, we followed an approach akin to classical neural networks by augmenting the number of trainable parameters in nPOLO. The parameter count per plane in nPOLO is W^2, where W is the plane's width. We conducted a sweep of W values from 32 to 1,024 (up to 512 for Fashion MNIST) for bounce numbers of 1, 2 and 4. The test accuracies over 50 epochs for the Imagenette and Fashion MNIST datasets are presented in Fig. 5. During scale-up studies we encountered overfitting, and slight data augmentation (random flip and rotation) was introduced to reduce this.

Experimental set-up

A photograph of our optical set-up is presented in Supplementary Section 3. In our experiments, we employed a continuous-wave Solstis M2 laser operating at a wavelength of $\lambda = 850$ nm. The mirror we used had a width of 11 mm, providing ample space for the four reflections. To deliver the beam to the SLM, we implemented $4f$ imaging to relay the beam reflected from a digital micromirror device (DMD). The use of a DMD offered the advantage of flexible beam sizing. Specifically, we configured the beam shape as a square with a side length of 2.4 mm, corresponding to a 300 × 300-pixel area on the SLM. It is worth noting that the SLM utilized in our set-up has a pixel pitch of $\Lambda = 8$ µm. To construct a multi-bounce cavity, we utilized a Holoeye Pluto SLM and positioned the mirror at a distance of $d = 15.2$ mm from the SLM screen. This distance was set so that diffraction from one corner pixel of a layer can reach the opposite corner of the subsequent layer ($d \times \lambda/\Lambda \geq 2.4$ mm). The distance between the SLM and mirror enabled all-to-all pixel connectivity. Zero-order reflection from the SLM was used, which has 69% percent reflectivity. This configuration allowed the input beam to undergo four reflections from the mirror. Therefore, the total transmission due to the SLM efficiency after four bounces was 23%. After the fourth bounce, the beam on the SLM was magnified by a factor of 1.2, and the resulting output intensity was detected by a complementary metal–oxide–semiconductor (CMOS) camera with a pixel pitch of 3.45 µm. The corresponding beam in the camera occupied an area of 834 × 834 pixels. During image acquisition, we applied average pooling to resize the obtained images to either 4 × 4 or 8 × 8 dimensions. For the Imagenette dataset, we used the entire training and test samples as originally prepared. For the Digit and Fashion MNIST datasets, we used the entire training set (60,000 samples) for the simulations, but we used the first 10,000 samples for the re-training of digital weights after the experiments, and we used the first 2,500 samples of the test set for blind testing of the experimental results.

Data availability

The datasets containing the raw information for the Imagenette dataset are from https://github.com/fastai/imagenette. The Digit and Fashion MNIST datasets can be retrieved from https://pytorch.org/vision/stable/datasets.html. Source data are provided with this paper.

Code availability

All relevant code is available from the corresponding authors upon reasonable request.

References

38. Kamilov, U. S. et al. Learning approach to optical tomography. *Optica* **2**, 517–522 (2015).
39. Tian, L. & Waller, L. 3D intensity and phase imaging from light field measurements in an LED array microscope. *Optica* **2**, 104–111 (2015).

Acknowledgements

M.Y. and N.U.D. are supported by the Swiss National Foundation, Sinergia grant CRSII5_216600 LION: Large Intelligent Optical Networks.

Author contributions

M.Y., N.U.D., D.P. and C.M. conceived and initiated the project. N.U.D. provided the first version of the numerical training module. M.Y. and N.U.D. modified the training scheme according to the needs of the project. M.Y., N.U.D. and I.O. worked on the experimental set-up and used datasets. All authors contributed to the guidance of the experiments and discussion of the results. M.Y. and N.U.D. prepared the first draft of the paper. All authors revised the paper. D.P. and C.M. supervised the project.

Competing interests

M.Y., N.U.D., I.O., D.P. and C.M. have filed a patent related to optical networks with structural nonlinearity, under application no. PCT/EP 2023/066812.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41566-024-01494-z.

Correspondence and requests for materials should be addressed to Mustafa Yildirim or Niyazi Ulas Dinc.

Peer review information *Nature Photonics* thanks Gordon Wetzstein and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.