Dual adaptive training of photonic neural networks

Ziyang Zheng[1,2†], Zhengyang Duan[1†], Hang Chen[1], Rui Yang[2], Sheng Gao[1], Haiou Zhang[1], Hongkai Xiong[2*] and Xing Lin[1,3*]

[1]Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China.
[2]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.
[3]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, 100084, China.

*Corresponding author(s). E-mail(s): lin-x@tsinghua.edu.cn; xionghongkai@sjtu.edu.cn;
†These authors contributed equally to this work.

Abstract

Photonic neural network (PNN) is a remarkable analog artificial intelligence (AI) accelerator that computes with photons instead of electrons to feature low latency, high energy efficiency, and high parallelism. However, the existing training approaches cannot address the extensive accumulation of systematic errors in large-scale PNNs, resulting in a significant decrease in model performance in physical systems. Here, we propose dual adaptive training (DAT) that allows the PNN model to adapt to substantial systematic errors and preserves its performance during the deployment. By introducing the systematic error prediction networks with task-similarity joint optimization, DAT achieves the high similarity mapping between the PNN numerical models and physical systems and high-accurate gradient calculations during the dual backpropagation training. We validated the effectiveness of DAT by using diffractive PNNs and interference-based PNNs on image classification tasks. DAT successfully trained large-scale PNNs under major systematic errors and preserved the model classification accuracies comparable to error-free systems. The results further demonstrated its superior performance over the state-of-the-art in situ training approaches. DAT provides critical support for constructing large-scale PNNs to achieve advanced architectures and can be generalized to other types of AI systems with analog computing errors.

1 Main

Artificial intelligence (AI), powered by deep neural networks (DNNs), utilizes brain-inspired information processing mechanisms to approach human-level performance in complex tasks [1], which has already achieved major applications ranging from translating languages [2], image recognition [3], cancer diagnosis [4] to fundamental science [5]. The vast majority of AI algorithms have been implemented via digital electronic computing platforms, such as graphics- and tensor-processing units, to support their major computing power requirement. However, the requirements of AI for processors' computing performance have grown rapidly, greatly exceeding the development of digital electronic computing imposed by Moore's law and the upper limit of computing energy efficiency [6, 7, 46]. Constructing the photonic neural network (PNN) systems for AI tasks with analog photonic computing has attracted increasing attention and is expected to be the next-generation AI computing modality with the advantages of low latency, high bandwidth, and low power consumption. The fundamental characteristic of photons and principle of light-matter interactions, such as diffraction [18–20] and interference [12, 16, 39] based on free-space optics or integrated photonic circuits, have been utilized to implement various neuromorphic photonic computing architectures, including convolutional neural networks [14, 15, 43, 44], spiking neural networks [10, 11, 17], recurrent neural networks [21, 22], and reservoir computing [23–25].

The effective training approach is one of the most critical aspects for DNNs to learn the reliable model and guarantee high inference accuracy. The DNNs constructed using software on a digital electronic computer generally train using backpropagation algorithm [26]. Such training mechanism provides the basis for the *in silico* training of photonic DNNs, which establishes the PNN models in computer to simulate physical systems, train models through backpropagation, and deploy the trained model parameters to physical systems. However, the inherent systematic errors of analog computing from different sources, e.g., geometric error and fabrication error, causes the deviation between the *in silico* trained PNN model and physical system and results in the performance degeneration during the directly deploying [20, 27, 28]. To address the systematic errors, the *in situ* training approaches, training PNNs on the physical systems with experimental measurements, have drawn increasing attention for optimizing the PNN models for practical applications [20, 28, 29, 31, 45, 48]. Nevertheless, the existing *in situ* training methods still confront great challenges in training large-scale PNNs with major systematic errors, which hinder the construction of advanced architectures and limit the model performance in performing complex AI tasks. The reasons for this are mainly due to the inaccurate gradient calculations during the backpropagation caused by the imprecise modeling of PNN physical systems [28, 45, 48], the requirement of extensive system measurements with layer-by-layer training processes [20], or the additional hardware configurations for backward optical field propagation [29, 31].

In this work, we propose dual adaptive training (DAT) for the end-to-end dual backpropagation training of large-scale PNNs, allowing the models to adapt to significant systematic errors without additional hardware configurations for backward optical field propagation. The basic principle of DAT for training PNNs with systematic errors is illustrated in Fig. 1. To precisely model the PNN physical system, we introduce the systematic error prediction networks (SEPNs) in addition to the PNN physical model and develop the task-similarity joint optimization approach for dual backpropagation training. The DAT iteratively updates the network parameters of

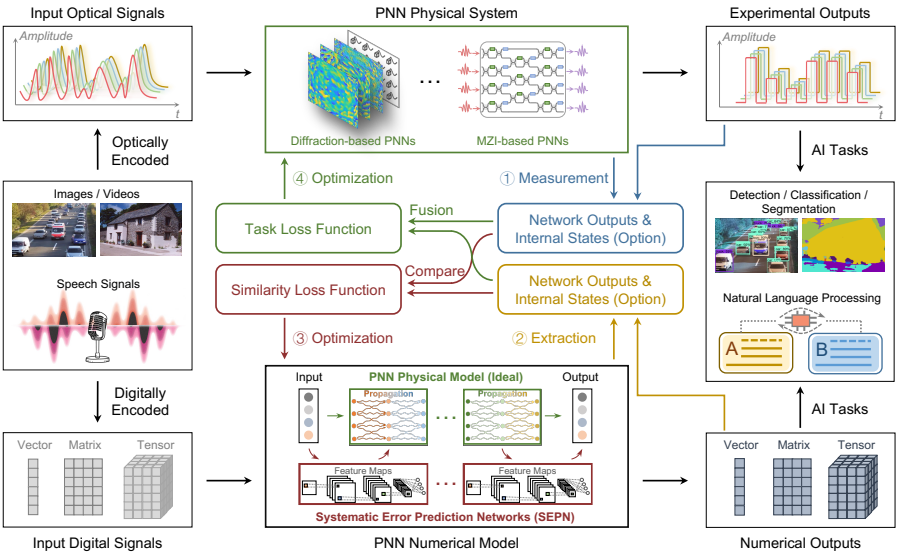


Fig. 1 Training PNNs with DAT. The input information is optically and digitally encoded to feed into the PNN physical system and numerical model, respectively. To adapt to the systematic errors, each DAT training cycle consists of four steps to perform the dual backpropagation, which iteratively updates the PNN physical model and SEPN parameters by optimizing the task and similarity loss functions, respectively.

PNN and SEPNs in an end-to-end form for each input training sample. With the training of SEPNs to characterize the inherent systematic errors, the DAT establishes high similarity mapping between the PNN numerical models and physical systems, leading to high-accurate gradient calculation for PNN training. Each training sample is optically and digitally encoded as the input to the PNN physical system and forward numerical model, respectively. The physically measured and the numerically extracted network outputs are fused to obtain the task loss function and compared to obtain the similarity loss function. The network's internal states, if provided, can be further used to boost the DAT performance. The dual backpropagation training process of DAT minimizes the task and similarity loss functions to update the network parameters of the physical model and SEPNs, respectively, by calculating the gradients of the PNN numerical model. After the training, the PNN physical model, deployed on the physical system, can adapt to significant systematic errors from various sources. Therefore, DAT supports large-scale PNN training and mitigates the requirement of high-precision fabrication and system configurations.

The constructed PNN numerical model in a digital computer comprises the ideal PNN physical model and SEPNs for modeling the photonic computing process and inherent systematic errors, respectively. To facilitate learning the systematic errors of PNN layers, SEPNs are incorporated in the manner of residual connections for the PNN layers (see Fig. 1 and Methods), inspired by the residual neural networks proposed in [32]. In this work, each SEPN module is configured with a complex-valued mini-UNet [33] to guarantee its learning capacity for fitting the systematic errors of PNN layers. The target is to eliminate performance degradation while deploying PNN physical model parameters to the physical system. With the established PNN

numerical model, the DAT for dual backpropagation training of the PNN numerical model consists of four main steps for each input training sample, including (1) the measurement of network outputs and optional internal states from the physical system; (2) the extraction of corresponding network outputs and optional internal states from the numerical model; (3) minimizing the similarity loss for backpropagation that updates the network parameters of SEPNs by comparing between the corresponding physical and numerical network's outputs and internal states; (4) minimizing the task loss for backpropagation that updates the network parameters of PNN physical model by replacing the numerical network's outputs and internal states with the physical measurements. We detail and formulate each of the training steps in Fig. 1 as follows:

First, each training sample is optically encoded and input to the PNN physical system to perform the forward inference, with which we obtain the physical network output $\mathbf{P}_N$ for a N-layer neural network with input $\mathbf{I}$. To further improve the training performance, the network internal states $\{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_{N-1}\}$ can be measured at each layer's output. All the observations $\{\mathbf{P}_n\}_{n=1}^{N}$ are set to be intensity, i.e., the absolute square of output complex optical fields, for facilitating the measurement. The optional internal states can boost the PNN training performance, especially under more severe systematic errors, but cause the additional cost of measurements. In practice, we can selectively measure a certain amount of internal states to reduce the number of measurements.

Second, the same training sample is also digitally encoded and input to the PNN numerical model to extract the internal states $\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_{N-1}\}$ and final observation $\mathbf{S}_N$. Different from the counterpart physical measurements $\mathbf{P}_n$, we set $\mathbf{S}_n = |\mathbf{S}_n| \exp(j\mathbf{\Phi}_{\mathbf{S}_n})$ to be the complex optical fields with the amplitude $|\mathbf{S}_n|$ and phase $\mathbf{\Phi}_{\mathbf{S}_n}$ for facilitating the formulation of DAT process, which can be easily obtained during the numerically modeling of PNN forward inference. Ideally, the $|\mathbf{S}_n|^2 = \mathbf{P}_n$ if the systematic errors can be perfectly characterized with SEPNs.

Third, we optimize the SEPNs' parameters of the PNN numerical model by minimizing the similarity loss function L_s as follows:

$$\min_{\Lambda} \left\{ L_\mathrm{s}(\mathbf{P}, |\mathbf{S}|^2) = \sum_{n=1}^{N} \alpha_n l_\mathrm{mse}(\mathbf{P}_n, |\mathbf{S}_n|^2) = \sum_{n=1}^{N} \alpha_n \|\mathbf{P}_n - |\mathbf{S}_n|^2\|_2^2 \right\}, \tag{1}$$

where $\mathbf{P} = \{\mathbf{P}_n\}_{n=1}^{N}$; $\mathbf{S} = \{\mathbf{S}_n\}_{n=1}^{N}$; Λ refers to the learnable parameters of SEPNs; $l_\mathrm{mse}(\cdot)$ denotes a mean square error (MSE) function; α_n is the coefficient to weight the n-th MSE function. For each training sample, the parameters of the PNN physical model are fixed, and the gradients of L_s with respect to Λ are calculated during the backpropagation to update SEPNs' parameters for one step. The optimization in Eq. (1) aims to train SEPNs to minimize the deviation between the physically measured and numerically extracted network output and internal states for accurately modeling the PNN physical system. We term the aforementioned training step for SEPNs as unitary mode since all SEPNs' parameters are optimized with a unitary loss function. In addition, the gradient calculation for SEPN modules can be implemented with separable mode (see Methods) when measuring internal states, where all SEPN modules are separated into several groups and optimized independently from each other.

Fourth, we optimize the physical parameters of the PNN numerical model and deploy them to the physical system by minimizing the following task loss L_t:

$$\min_{\Omega} \left\{ L_\mathrm{t}(|F_N(\mathbf{P}_N, \mathbf{S}_N)|^2, \mathbf{T}) \right\}, \tag{2}$$

where Ω refers to the learnable parameters of the physical model; $\mathbf{T}$ denotes the desired output; L_{t} is defined based on the target task implemented by PNNs, which is set to be the cross-entropy loss [1] for classification tasks in this work; and $F_N(\mathbf{P}_N, \mathbf{S}_N)$ represents the output of the fusion function F_N that replaces the amplitude of the numerically extracted network output with the physically measured counterpart. Furthermore, such fusion processes are applied for not only the network output but also the network internal states to maintain the interactions with the physical system; therefore, we have $\{F_n(\mathbf{P}_n, \mathbf{S}_n) = \sqrt{\mathbf{P}_n}\exp(j\mathbf{\Phi}_{\mathbf{S}_n})\}_{n=1}^{N}$. During the backpropagation for updating physical parameters with one step for each training sample, the parameters of SEPNs are fixed, and the fused network output and internal states are used for calculating the gradients of L_{t} with respect to the physical system parameters Ω. The optimization in Eq. (2) aims to train the PNN physical model under systematic errors so that the PNN physical system deployed with physical parameters Ω can perform the target tasks.

The above training steps are repeated over all training samples to minimize the loss functions until convergence for obtaining the PNN numerical model and physical parameters Ω for the physical system. We term the training process as dual backpropagation training since the gradient calculation for updating the parameters of the PNN physical model and SEPNs rely on each other. Furthermore, the training of the PNN physical model promotes the training of SEPNs and vice versa. On the one hand, the optimization of physical parameters facilitates characterizing inherent systematic errors with SEPNs for task-specific physical models. On the other hand, the optimization of SEPNs' parameters facilitates performing the inference tasks with physical models under practical systematic errors. Besides, the state and output fusion processes allow the PNN physical model to further adapt the systematic errors and accelerate the convergence, especially when the SEPNs haven't fully characterized the systematic errors during the optimization. These underlying mechanisms guarantee the effectiveness and convergence of the proposed DAT.

We validate the effectiveness of DAT by applying it for training large-scale diffractive PNNs (DPNNs) [18, 20] and interference-based PNNs (MPNNs) [12, 39] under various systematic errors. The network settings and training processes for two types of models are detailed in the Methods section. Two benchmark datasets, i.e., the Modified National Institute of Standards and Technology (MNIST) [26] and Fashion-MNIST (FMNIST) [42], were utilized for the performance evaluations. The results demonstrate the superior performance of DAT over the *in silico* training with direct deployment and the state-of-the-art *in situ* training method using physics-aware training (PAT) [28, 45].

2 Results

2.1 Training DPNN with DAT

We built two types of DPNN architectures, i.e., the DPNN-S and DPNN-M, as illustrated in Fig. 2a and Fig. 2b, respectively. DPNN-S in Fig. 2a was constructed using a single PNN block, where the block comprised the cascading of two phase modulation layers with transformation matrices $\mathbf{M}_{11}, \mathbf{M}_{12}$, followed by an opto-electronic intensity measurement layer at the output plane. The output layer of DPNN-S records the intensity $\mathbf{P}_1$ of output optical fields for the input $\mathbf{I}$. Specifically, the diffractive elements on a phase modulation layer are able to modulate the phase of input optical fields, and the secondary wave sources are generated via

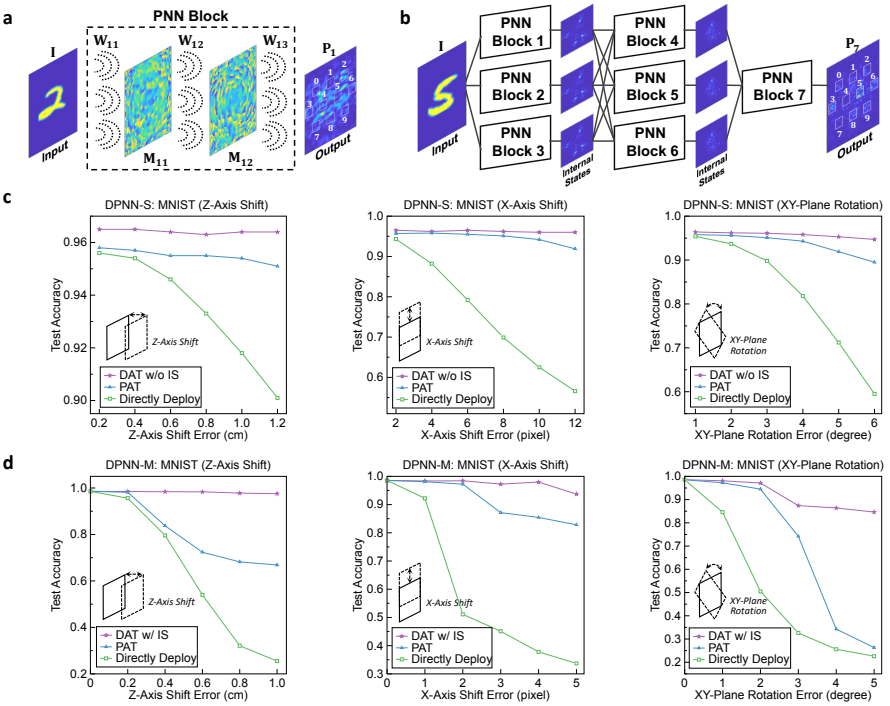


Fig. 2 Training DPNNs under three types of systematic errors for the MNIST classification. a, DPNN-S with a single block. The block consists of two phase modulation layers and one intensity measurement layer, where the optical diffraction for weighted matrices occurs between adjacent layers. **b**, DPNN-M with seven hierarchically interconnected blocks, where each block is the same as the counterpart in **a**. The evaluations of DAT performance with DPNN-S (**c**) and DPNN-M (**d**). The performances of DAT are compared with the PAT and direct deployment of *in silico* trained models under different amounts of systematic errors. The DPNN-S and DPNN-M are trained without and with the internal states (IS), respectively.

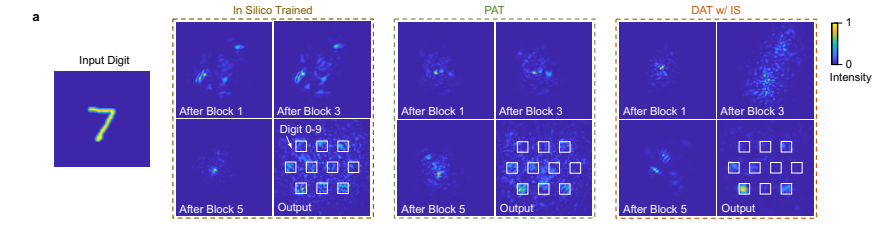
optical diffraction to interconnect to the next phase modulation layer or the output plane for intensity measurements. The forward propagation of a single PNN block has three free-space diffraction processes with three diffractive matrices $\mathbf{W}_{11}, \mathbf{W}_{12}$, and $\mathbf{W}_{13}$. Therefore, the mathematical forward model of DPNN-S can be defined as: $\mathbf{P}_1 = |\mathbf{W}_{13}\mathbf{M}_{12}\mathbf{W}_{12}\mathbf{M}_{11}\mathbf{W}_{11}\mathbf{I}|^2$. To further demonstrate the effectiveness of the proposed method on DPNNs with larger network scales, we constructed DPNN-M [20] that was designed with multiple PNN blocks to constitute multi-channel diffractive hidden layers with hierarchically interconnected structures. Each PNN block of DPNN-M is the same as the counterpart in DPNN-S, but their parameters are independent and not shared. DPNN-M has been demonstrated to achieve higher model performance yet inevitably accumulates more extensive systematic errors layer by layer with more complicated network structures.

For both DPNN-S and DPNN-M, the phase modulation coefficients are set as the learnable parameters and thus be optimized via end-to-end network training. In addition to recording intensity, the optoelectronic detectors on the output plane can be regarded as complex activation functions to accomplish the nonlinearity. The final

output intensity is used for approximating the desired target by minimizing the specific task loss L_t. We detail the DPNN settings and its training process with DAT in Methods. We further provide the pseudo-code of DAT in Supplementary Appendix A and Supplementary Algorithm S1, and elaborate on the standard training process in Supplementary Appendix B. For simplicity, all SEPN modules share the same network architecture for DPNN-S and DPNN-M (see Methods and Extended Data Fig. 1). Each SEPN was constructed as a complex-valued mini-UNet [33] to extract hierarchical features, which is much simpler and lighter than a standard UNet. The trainable parameters of a SEPN module and UNet are 26,800 and 7,765,442, respectively, with a parameter ratio of 0.345%.

We trained DPNN-S and DPNN-M with DAT for the MNIST and FMNIST classification tasks and compared their performance with PAT and direct deployment by considering four types of systematic errors in practical systems, i.e., *Z-Axis* shift error, *X-Axis* shift error, *XY-Plane* rotation error, and phase shift error (see Methods for detail description). The first three errors are geometric errors mainly due to the imprecision of alignments, which are included in a layer-by-layer manner, each with the same amount of errors. For example, a single pixel *X-Axis* shift error between successive layers in Fig. 2 results in the *X-Axis* shift of three pixels in DPNN-S with a single block and nine pixels in DPNN-M. The phase shift error, modeled with a normal distribution with zero mean and standard deviation σ, is mainly caused by the imperfection of phase modulation devices that leads to the deviation of phase modulations. The classification performances of DPNN models were evaluated under individual and joint systematic errors to validate the effectiveness of DAT in various scenarios with different systematic error configurations.

The MNIST classification results of DPNN-S and DPNN-M under the different amounts of individual geometric errors are shown in Fig. 2c and 2d, respectively. As the phase shift errors have a minor effect on the classification performance of DPNN, we only evaluate the phase shift error in joint systematic errors (See Fig. 3c). The MNIST classification accuracy of the baseline model for an error-free system is 96.0% for DPNN-S and 98.6% for DPNN-M. We implemented DAT without measuring internal states (DAT w/o IS) for DPNN-S and DAT with measuring internal states (DAT w/ IS) for DPNN-M. Here, DAT with internal states for updating SEPNs was conducted in a separable mode. For both DPNN-S and DPNN-M, the test accuracy decreases rapidly when directly deploying the *in silico* trained model to the physical system, where DPNN-M has a larger decrease in classification accuracy than DPNN-S due to the accumulation of more systematic errors. The PAT method [28, 45] can correct the errors to some extent but is not effective when the errors become severe and accumulate layer by layer, especially for the DPNN-M with a larger network scale. For example, one can see from Fig. 2d that PAT only improves the accuracy from 25.5% obtained by direct deployment to 66.9% when *Z-Axis* shift error is set to 1 cm and fails when *XY-Plane* shift error is set to 5 degrees as it only improves the accuracy from 22.6% to 26.3%. By contrast, DAT outperforms PAT and dramatically eliminates the performance degradation caused by different systematic errors, making the classification accuracies comparable with and even slightly higher, e.g., for the DPNN-S model with *Z-Axis* shift error, than the error-free systems. These results validate the effectiveness and robustness of DAT for training DPNN physical systems, especially demonstrating its powerful capacity to adapt to significant systematic errors from various sources in large-scale DPNN-M. Moreover, the results for FMNIST classification shown in Extended Data Fig. 2 justify the same conclusion.



c

DPNN Model	Error Configurations				MNIST Classification Accuracy				FMNIST Classification Accuracy			
	Z-Axis Shift (cm)	X-Axis Shift (pixel)	XY-Plane Rotation (degree)	Phase Shift (std)	Directly Deploy	PAT	DAT		Directly Deploy	PAT	DAT	
							w/o IS	w/ IS			w/o IS	w/ IS
DPNN -S	1.0	4	5	0.2	56.1%	89.1%	94.5%	/	31.2%	75.2%	80.2%	/
	1.0	6	4	0.3	55.6%	91.2%	94.2%	/	32.9%	75.8%	80.3%	/
	1.0	10	2	0.2	55.9%	91.8%	95.3%	/	33.6%	73.7%	82.2%	/
DPNN -M	0.2	1	1	0.1	65.4%	90.7%	/	96.6%	47.2%	77.1%	/	81.3%
	0.4	2	2	0.2	35.5%	50.4%	/	94.9%	14.9%	25.5%	/	78.7%
	0.6	2	1	0.3	37.5%	16.4%	/	92.3%	14.8%	30.0%	/	79.1%

d

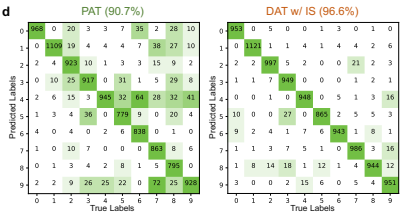


Fig. 3 Training DPNN under joint systematic errors for the MNIST and FMNIST classification. The performance of DAT is evaluated on the DPNN-S and DPNN-M architectures and compared with the PAT and direct deployment of *in silico* trained model under different joint systematic error configurations, as shown in Table **c**. The first configuration of DPNN-M listed in Table **c** was selected for the visualization of the network internal states, phase modulation layers, and confusion matrices on the MNIST classification in **a**, **b**, and **d**, respectively.

We further evaluated the performance of DAT for training DPNN-S and DPNN-M under joint systematic errors, as illustrated in Fig. 3. The table in Fig. 3c lists the results of six joint systematic error configurations, where both DPNN-S and DPNN-M are assigned three configurations. We took the experimental MNIST classification accuracies of 63.9% with the direct deployment to a physical system in [20] as a reference to design these error configurations, with comparable or larger systematic errors as reflected in the accuracies of direct deployment in Fig. 3c. Notice that the FMNIST classification accuracy of the baseline model for an error-free system is 83.8% for DPNN-S and 85.8% for DPNN-M. In the joint systematic errors, DAT also achieves superior classification accuracies over PAT and restores the model performance, especially in the large-scale DPNN-M with more significant joint systematic error. PAT fails to train DPNN-M with the last joint systematic error configuration for MNIST classification, as the accuracies are even lower than the direct deployment of *in silico* trained model. By contrast, DAT successfully trains the DPNN-M, which significantly outperforms PAT and improves the classification accuracy by an average of 42.1% for MNIST and 35.5% for FMNIST. Besides, DAT has a larger improvement over the direct deployment method with an average accuracy of 48.5% than 32.1% in [20] for MNIST classification. We further showed the convergence plots

during the training stage in Extended Data Fig. 3, demonstrating that DAT was more robust than PAT, especially for DPNN-M.

Fig. 3a, 3b, and 3d illustrate the results for DPNN-M in the task of MNIST classification with the first joint systematic configuration in Fig. 3c. Fig. 3a visualizes the network internal states of the first, third, and fifth PNN blocks and final output with the example input digit '7' from the test set. The output intensities are distributed throughout the plane for *in silico* trained model and around detection regions for PAT, thus leading to an incorrect recognition category. By contrast, the intensities are concentrated in the correct detector region (the bottom left one) for DAT, and thus the example testing digit '7' can be correctly categorized. Fig. 3b illustrates the phase modulation layers $\mathbf{M}_{n1}$ and $\mathbf{M}_{n2}$ of the n-th PNN block for $n = 1, 4, 7$. The phase modulation layers obtained by PAT and DAT are dramatically different. Different from PAT, which generates phase modulation layers with a relatively flat distribution of values, DAT tends to generate a drastically uneven distribution to adapt to systematic errors, according to the contrast between the yellow (near π) and blue (near $-\pi$) areas. The confusion matrices in Fig. 3d summarize the classification results of 10,000 digits in the test set and further reveal the effectiveness of DAT as it concentrates the matched pairs of predicted labels and true labels on the main diagonal.

2.2 Training MPNN with DAT

As shown in Fig. 4a, the N-layer MPNN consists of N photonic meshes and $N-1$ optoelectronic units between adjacent photonic meshes for implementing nonlinear activation functions. Each photonic mesh is constructed with the array of MZIs formed as the rectangular grid [38]. Each MZI is a two-port optical component made of two $50:50$ beamsplitters $\mathbf{B}_1, \mathbf{B}_2$ and two tunable single-mode phase shifters with parameters ϕ, θ. In the n-th photonic mesh, the input optical field encoded in single-mode waveguides is multiplied with a unitary matrix $\hat{\mathbf{M}}_n$ realized by the n-th photonic mesh. The result is further processed with an optoelectronic unit with the function $f_{EO}(\cdot)$ for nonlinear processing, except for the final photonic mesh, to generate the output optical fields for the next layer. The $f_{EO}(\cdot)$ is the optoelectronic nonlinear activation function introduced in [39] (see Methods for the formulation). The output intensity at the last photonic mesh is measured by photodetectors and used for obtaining the inference result of a task. The mathematical forward model and the training process of DAT for the MPNN are elaborated in Methods. Similar to DPNNs, all SEPN modules for MPNNs share the same complex-valued mini-UNet architecture yet are lighter than the counterparts utilized in DPNN training. Specifically, we constructed each SEPN module with two different numbers of learnable parameters, i.e., 9,648 and 3,960 parameters, to evaluate the influence of the SEPN scale on the classification performance. Compared with the standard UNet with 7,765,442 parameters, the parameter ratios of the two SEPNs are 0.124% and 0.051%, respectively.

The input data are pre-processed to facilitate the on-chip implementation of MPNNs with a limited number of input ports (see Methods and Fig. 5a). Similar to [39], we extracted 64 Fourier coefficients in the center region of the Fourier-space representations as the input for MNIST and FMNIST classification. To match the input dimension, each photonic mesh consists of $64 \times 63/2 = 2016$ MZIs (see Methods and Supplementary Appendix C) that contains 4032 beamsplitters, 2016 phase shifters with parameters ϕ, and 2016 phase shifters with parameters θ. We built the

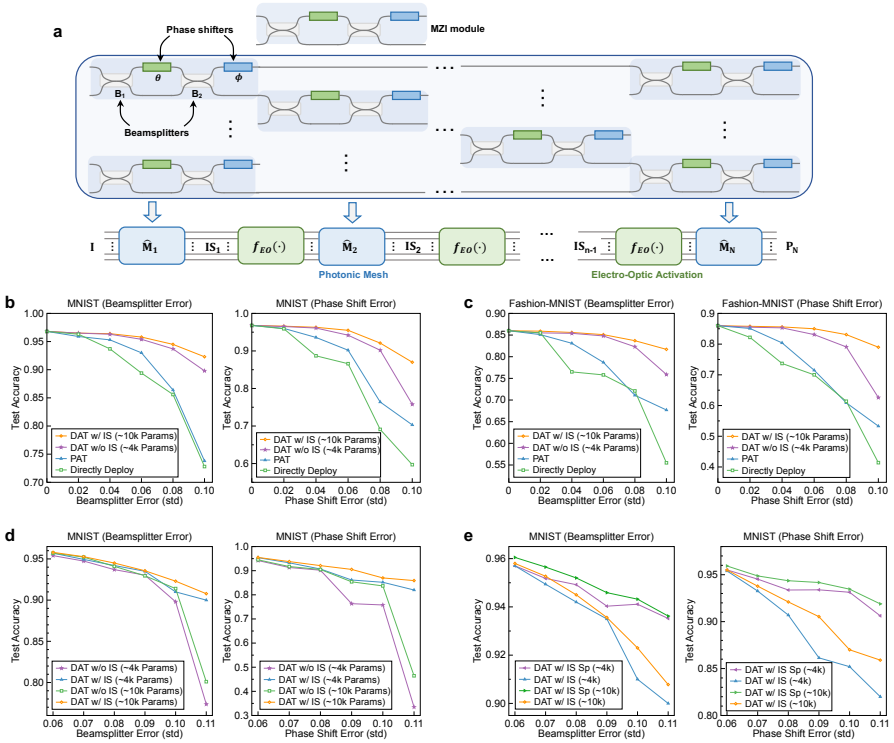
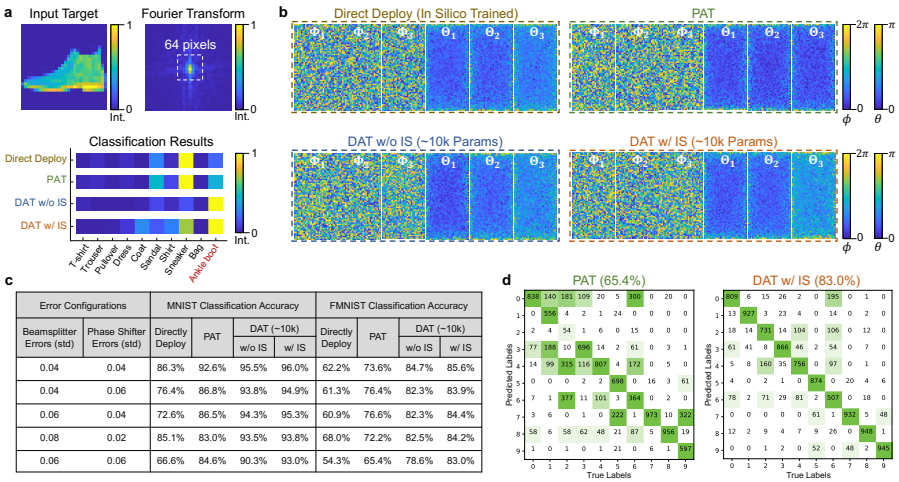


Fig. 4 Training MPNNs under two types of systematic errors for the MNIST and FMNIST classification. a, Schematic illustration of the N-layer MPNN consisting of N photonic meshes and $N-1$ optoelectronic activation units. Each photonic mesh comprises an array of MZIs with a rectangular grid (top subgraph), where each MZI consists of two beamsplitters and two phase shifters. The bottom subgraph describes the forward inference of the MPNN architecture, where IS_n denotes the location to obtain internal states. **b,c**, Comparing the performances of DAT with and without internal states (IS), PAT, and direct deployment of *in silico* trained model on training three-layer MPNNs under beamsplitter and phase shift errors for the MNIST and FMNIST classifications. **d**, Comparisons between DAT with and without internal states under different SEPN scales. **e**, Comparisons between separable and unitary training modes for optimizing SEPNs under different SEPN scales when adopting DAT with internal states.

MPNN with $N = 3$ for MNIST and FMINST classification, where the MPNN settings and training process are detailed in Methods. We considered two kinds of systematic errors occurring in MZIs, i.e., beamsplitter error and phase shifter error [36, 40], caused by the imperfection of fabrications and inaccuracy of optical modulations. The beamsplitter error and phase shifter error are modeled with a normal distribution with zero mean and standard deviation of σ_{bs} and σ_{ps}, respectively. Besides, the errors are included in all devices and share the same strengths. For example, $\sigma_{ps} = 0.1$ means that the error corrupts all the 4032 phase shifters following a normal distribution with zero mean and standard deviation of 0.1.

We compare the DAT with PAT and direct deployment of *in silico* trained model for the MNIST and FMNIST classification under two types of systematic errors in Fig. 4b and 4c, respectively. The legends marked with '∼10k Params' or '∼4k Params'



Error Configurations		MNIST Classification Accuracy				FMNIST Classification Accuracy			
Beamsplitter Errors (std)	Phase Shifter Errors (std)	Directly Deploy	PAT	DAT (~10k)		Directly Deploy	PAT	DAT (~10k)	
				w/o IS	w/ IS			w/o IS	w/ IS
0.04	0.04	86.3%	92.6%	95.5%	96.0%	62.2%	73.6%	84.7%	85.6%
0.04	0.06	76.4%	86.8%	93.8%	94.9%	61.3%	76.4%	82.3%	83.9%
0.06	0.04	72.6%	86.5%	94.3%	95.3%	60.9%	76.6%	82.3%	84.4%
0.08	0.02	85.1%	83.0%	93.5%	93.8%	68.0%	72.2%	82.5%	84.2%
0.06	0.06	66.6%	84.6%	90.3%	93.0%	54.3%	65.4%	78.6%	83.0%

Fig. 5 Training MPNN under joint systematic errors for the MNIST and FMNIST classification. The performances of DAT with and without using internal states are evaluated on the MPNN architectures and compared with the PAT and direct deployment of *in silico* trained model under different joint systematic error configurations, as shown in Table **c**. The last configuration of MPNN listed in Table **c** was selected for the visualizations of an example result of fashion product 'ankle boot', phase shifter values, and confusion matrices on the FMNIST classification in **a**, **b**, and **d**, respectively.

denote implementing DAT with the learnable parameters of 9,648 or 3,960 for each SEPN. All the internal states of intensities measured at the output of photonic meshes are utilized for DAT with internal states (IS), and the performance of DAT with partially measured internal states is discussed in Extended Data Fig. 4. The classification accuracy of the baseline MPNN model in an error-free system is 96.8% for MNIST and 86.0% for FMNIST. As shown in Fig. 4b and 4c, DAT outperforms PAT and direct deployment even without measuring internal states and with the relatively small SEPN scale. By contrast, PAT confronts the difficulty of training, especially under large systematic errors. For example, the accuracy is 71.1% when using PAT for FMNIST classification under $\sigma_{bs} = 0.08$, while the accuracy for direct deployment is 72.1% with the same error. At the same time, DAT with and without internal states achieve classification accuracies of 83.7% and 82.3%, respectively, indicating the effectiveness of SEPNs for characterizing systematic errors during the DAT training. In the individual error configurations with the largest stds listed in Fig. 4b and 4c, DAT with internal states exceeds PAT by 16.0% (MNIST, $\sigma_{bs} = 0.1$), 8.2% (MNIST, $\sigma_{ps} = 0.1$), 9.2% (FMNIST, $\sigma_{bs} = 0.1$), 9.3% (FMNIST, $\sigma_{ps} = 0.1$), and the data for DAT with internal states increase to 18.5%, 14.0%, 16.7%, and 25.7%. The results demonstrate the superior performance and robustness of DAT for training MPNN with significant systematic errors.

We further evaluated the influence of the SEPN scale on the performance of MNIST classification under different amounts of systematic errors with the std range from 0.06 to 0.11 in Fig 4d. The accuracies are close with relatively small errors (from 0.06 to 0.08), while the gap becomes evident as the std increases (from 0.09 to 0.11). With the same SEPN scale, DAT with internal states outperforms DAT without internal states, especially under larger systematic errors. Although a large SEPN scale facilitates the classification, one can find that it is not evident for beamsplitter

error yet effective for phase shifter error. Compared with the SEPN with about 4k parameters, the larger one with about 10k parameters improves the test accuracy by 0.7% and 7.17% on average for beamsplitter and phase shifter error, respectively. Furthermore, Fig 4e compares the performances between unitary and separable mode (denoted by the term 'Sp' in the legends) of DAT with internal states for the MNIST classification. The separable mode surpasses the unitary mode significantly with the same SEPN scale, especially with increased std. Besides, DAT with the larger scale SEPNs improves the accuracy trivially for beamsplitter errors no matter which training mode is chosen, but it has a relatively significant improvement on performance for phase shifter errors.

The classification results of MPNN under five joint systematic error configurations with different error strengths are listed in Fig. 5c. Here, DAT was implemented with and without internal states and configured the SEPN with ∼10k parameters in a unitary optimization mode. DAT without internal states can adapt moderate systematic errors, e.g., it can improve the MNIST/FMNIST classification accuracy from 72.6%/60.9% of direct deployment to 94.3%/82.3% when $\sigma_{bs} = 0.06$ and $\sigma_{ps} = 0.04$. When corrupted by severe errors, measuring internal states can obviously improve the test accuracy, e.g., DAT with internal states exceeded DAT without internal states by 2.7%/4.4% for MNIST/FMNIST classification when $\sigma_{bs} = 0.06$ and $\sigma_{ps} = 0.06$. Meanwhile, DAT outperforms PAT by a large margin, especially under severe errors. Fig. 5a, 5b, and 5d visualize the results for FMNIST classification when $\sigma_{bs} = 0.06$ and $\sigma_{ps} = 0.06$. Fig. 3a depicts the visualization of the intensities of input and output of the example product 'ankle boot'. The input is the 64-pixel values in the center region of the Fourier-space representations, and the output is the intensities on 10 photodetectors corresponding to 10 categories. The classification results demonstrate that *in silico* trained model and PAT fail to classify the example to the true category (the last detector), whereas DAT suppresses the errors and successfully obtains the true classification result. Fig. 3b illustrates the phase shift values $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \mathbf{\Phi}_3$ consisting of all phase shifter values of ϕ ranging from 0 to 2π and $\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3$ consisting of all phase shifter values of θ within $(0, \pi)$. Fig. 5d further plots the confusion matrices representing the classification results of 10,000 products in the FMNIST test set, showing that DAT can effectively optimize the MPNN to extract the characteristics of some products that are hard to identify for PAT. For example, only 5.4% products of 'pullover' (category No.2) were correctly categorized for PAT, and the accuracies soared to 73.1% for DAT with internal states.

3 Discussion

In this work, we propose DAT for effectively training large-scale PNNs under significant systematic errors. The PNN numerical model, comprising the physical model and SEPNs, of the physical system, is optimized through dual backpropagation training in an end-to-end form that iteratively updates the parameters for each input training sample. Compared with the existing *in situ* training methods, e.g., using extensive system measurements with layer-by-layer training process [20] or additional hardware configurations for backward optical field propagation [29, 31], the DAT is a more general approach and cost-efficient for training large-scale analog PNN systems. It only requires forward inference to record output intensity with the optional internal states. For example, for the MPNN with L-dimensional input vector and N photonic meshes in Fig. 4a, the *in situ* training method in [29] requires $3NL(L-1)/2$

intensity measurements of phase shifters, i.e., three times of measures for each phase shifter, to generate of backward optical fields for each training sample in addition to the output intensity. In contrast, DAT with internal states only needs to measure N output intensities of photonic meshes, and DAT without internal states only requires the final network output without generating backward optical fields. Meanwhile, we have validated that DAT achieves more accurate gradient calculations than PAT for the more robust optimization of large-scale PNNs under various inherent systematic errors of varying strengths. More comparisons of the proposed DAT with the existing PNN training methods are provided in Supplementary Appendix D.

The underlying principle of DAT for high-precision *in situ* training is to transfer the additional hardware complexity to scalable algorithm complexity by introducing SEPNs during the numerical modeling. Intuitively, the parameters of SEPNs need to be proportional to the error strengths. Empirically, we found that the total parameters of the SEPN setting at the same scale with respect to the physical model can create enough fitting capacity for systematic errors. In this work, the parameter ratio between the SEPNs and physical model is approximately 1.0 for training the DPNN and MPNN with '∼4k' parameters. Besides, we found that increasing the SEPN scale with the parameter ratio to approximate 2.4 for training MPNN with '∼10k' parameters has considerably less influence on the performance than the training strategies, including whether measuring the internal states or not and using unitary or separable training mode. Furthermore, the learnable parameters of SEPNs with the complex-valued mini-UNet are significantly lighter than standard UNet (even 0.051% for the ratio of parameters) is enough to dramatically eliminate the performance degradation under various errors. For the connectivity, we incorporate SEPNs into the physical model with residual connections [32], which is demonstrated to be effective in independently modeling the inherent systematic errors. Suppose the SEPNs are connected directly from the input to the output for each PNN physical layer. In that case, the SEPN needs to share the responsibility for modeling the physical computing process, resulting in the requirement of large ESPN scale and inefficiency of learning.

DAT has significant advantages in training PNNs under larger systematic errors compared with the state-of-the-art *in situ* training approaches, which facilitates the training of larger network scale and mitigates the system and fabrication precision, such as the translation stages for alignments in DPNNs and the on-chip fabrications of beamsplitter and phase shifter in MPNNs. Besides, we propose the unitary and separable mode of DAT for training PNNs with or without internal states to deal with different scenarios. Generally, the separable mode is more robust with higher performance, especially for large-scale PNNs, as it refines the optimization of SEPNs without apparently increasing the computational complexity. Furthermore, although the DAT only be examined on DPNN and MPNN in this work, it's a general *in situ* training paradigm that can be applied for universal PNN training or other types of AI systems with analog computing errors.

4 Methods

4.1 Preprocessing of benchmarks

The two benchmark datasets for classification, i.e., MNIST and FMNIST, consist of 70,000 grayscale 28×28 pixel images of 10 handwritten digits and fashion products of 10 classes, respectively. Specifically, the MNIST dataset contains digit categories from 0 to 9, while the FMNIST dataset contains product types of t-shirt, trousers,

pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boots. In addition, both datasets consist of a training set of 60,000 examples and a test set of 10,000 examples.

For the DAT training of DPNNs, the images were first upscaled to the size of 100×100 using bilinear interpolation to facilitate the physical fabrication and the network optimization. Then, to further satisfy the boundary condition of free-space propagation numerically implemented with the angular spectrum method [18, 22, 31], the images were further padded with zeros to the size of 200×200. Following previous works [18, 31], we adopted coherent illumination to encode the input information. Specifically, these preprocessed images were encoded into the amplitude of the complex optical fields with zero phases under a working wavelength of 1550 nm.

For the DAT training of MPNNs, we cropped the data to reduce the input size following the steps in [39]. Specifically, the images with an original size of 28×28 were first converted into two-dimensional Fourier-space representations, then 64 pixels in the center region of the representations, i.e., 8×8 Fourier coefficients closest to the center point, were extracted as input since the Fourier-space energy is mostly concentrated in the low-frequency domain located in the center region. This process is illustrated in Fig. 5a. After the date preprocessing, the 64-dimensional signals were input to the MPNNs through 64 input strip optical waveguides. The input information was compressed with the compression ratio of $64/28^2 \approx 8.16\%$ in comparison to the original data. Higher compression ratios can achieve better classification performance but require more input ports and raise costs.

4.2 DPNN settings

We built DPNN-S and DPNN-M as shown in Fig. 2a and 2b, and evaluated the DAT training performance with two models. Following previous works [20], the phase modulation layers can be implemented using programmable spatial light modulators (SLMs). In our DPNNs, the pixel size of SLMs was set to 17 μm under the working wavelength of 1550 nm during the training and testing. Each phase modulation layer was well configured by packing up 200×200 neurons, i.e., there are $40,000$ learnable parameters for each layer covering an area of 3.4 mm $\times$ 3.4 mm on the SLM. The total number of input nodes and neurons determined by the phase modulation layers is 0.12 million and 0.84 million for the DPNN-S and DPNN-M, respectively. The periphery of the phase modulation layer was zero-padded to guarantee the boundary condition of free-space diffraction during the numerical modeling with the angular spectrum method [18, 22, 31]. Besides, the distances between successive layers were fixed to 30 cm for the DPNN-S and 10 cm for the DPNN-M. Both distances for optical diffraction enable a fully connected structure according to the Huygens–Fresnel principle for calculating the maximum diffractive angle [18]. The sigmoid function was used to limit the phase modulation range of each element to $0 \sim 2\pi$ [18, 31], which facilitates DPNN training and makes full use of the full-range phase modulation of SLM. All the phase modulation parameters were randomly initialized before network training.

The output plane after each PNN block utilized an optoelectronic sensor to measure the intensity of the whole optical field and also served as a nonlinear activation function between adjacent PNN blocks to provide a powerful capacity for feature extraction. In the last PNN block, the output plane contains 10 detector regions corresponding to 10 classes of digits in MNIST or products in FMNIST. Each detector region covers 22×22 pixels with detector width 0.374 mm. The classification criterion is to find the detector region with the maximal intensity by optimizing the DPNNs with a cross-entropy loss function. During the training, the intensities of

the pixels in each detector region were summed up and normalized by the softmax function to generate a 10-dimensional vector. Then, the cross-entropy loss function was introduced as the task loss L_{t} to minimize the deviation between the generated vector and the ground truth $\mathbf{T}$.

4.3 MPNN settings

We built the MPNNs as shown in Fig. 4a with $N = 3$ as used in [36, 39, 40]. We chose Clements scheme [38] to construct MPNNs, which was discussed detailed in [36]. Each photonic mesh was connected with 64 input and output optical waveguides to match the input and output dimensions, where the 64 input optical waveguides were used to input the preprocessed data, and the other 64 output optical waveguides were used to transfer the internal states to the following optoelectronic unit except for the last photonic mesh. In the last mesh, we adopted a drop-mask to reduce the final output to 10 components to match the classification categories; thus, only 10 waveguide ports were utilized. The intensities of the 10 outputs were normalized by the softmax function and then compared with the one-hot encoding of the target vector. We utilized the cross-entropy loss function as the task loss L_{t} to minimize the deviation between the normalized output intensities and the correct one-hot vector and optimize the numerical model. In addition, each photonic mesh consisted of $64 \times 63/2 = 2016$ MZIs (see Supplementary Appendix C for more details) that contained 4032 beamsplitters, 2016 phase shifters with parameters ϕ, and 2016 phase shifters with parameters θ. At the start of the training process, each ϕ was initialized to a random value in $[0, 2\pi]$ following a uniform distribution, i.e., $\phi \sim U[0, 2\pi]$, and each θ was initialized following $\theta \sim U[0, \pi]$.

4.4 Training details of DPNN

Both the DPNN-S and DPNN-M were trained using a stochastic gradient descent algorithm, i.e., the adaptive moment estimation (Adam) optimizer [47] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, during the *in silico* training, PAT and DAT processes. With *in silico* training to obtain the baseline accuracy in an error-free system, the physical model of DPNN-S was optimized for 10 epochs with a batch size of 32 and an initial learning rate 0.01 decayed by 0.5 every epoch, while DPNN-M was trained for 50 epochs with a batch size of 128 and an initial learning rate 0.01 decayed by 0.5 every 10 epoch due to the large scale. Besides, DPNN-S were trained for both 5 epochs with PAT and DAT, while DPNN-M were trained for 50 and 10 epochs with PAT and DAT, respectively. Except for the optimization of the physical model, there is another gradient descent step to update SEPNs for each training sample during the DAT process. The same Adam optimizer was utilized with a constant learning rate 0.001 to minimize L_{s} and optimize all the SEPNs in a unitary or separable mode. All the experiments were performed on a desktop computer with Intel Xeon Gold 6226R CPU at 2.90GHz with 16 cores and an Nvidia GTX-3090Ti GPU of 24 GB graphics card memory.

4.5 Training details of MPNN

We adopted the same Adam optimizer as the experiments of DPNNs to train the MPNN with *in silico* training, PAT, and DAT. Specifically, the physical model of the MPNN as in Fig 4(a) with $N = 3$ was trained for 50 epochs with a batch size

of 32 and an initial learning rate 0.001 decayed by 0.5 every 10 epochs during the *in silico* training, PAT and DAT processes. The extra gradient descent step in DAT for optimizing SEPNs was implemented with an initial learning rate of 0.001 that was decayed by 0.5 every 20 epochs. In addition, during the first 20 epochs of DAT w/o IS, SEPNs were optimized to predict the systematic errors, but the connection with the physical model was cut off during the optimization of the physical model, which meant that SEPNs did not participate in the backpropagation process and the calculation of gradients. Since the SEPNs haven't fully characterized the systematic errors in the first few epochs, it may cause unstable optimization when involved in updating the physical model. After the first 20 epochs, SEPNs were roughly trained and then reconnected with the physical model to implement a standard optimization in the last 30 epochs.

4.6 Architecture of SEPNs

All SEPNs shared the same architecture as illustrated in Extended Data Fig. 1 in this work. Inspired by UNet [33], each SEPN was designed as a complex mini-UNet with hierarchically interconnected structures to extract multiscale features, while it was much simpler and lighter than UNet. To match the complex-valued computation of DPNNs and MPNNs, We adopted complex-valued weights similar to [41] to construct the SEPNs. As shown in Extended Data Fig. 1, each **c**omplex-valued **conv**olution layer (CConv) is set to a size of 5×5 in the DPNN and 3×3 in the MPNN, followed by a **c**omplex-valued **ReLU** (CReLU) except for the last convolution layer, where the CReLU is defined as: $\mathrm{CReLU}(\mathbf{x}) = \mathrm{ReLU}(\mathrm{Re}\{\mathbf{x}\}) + j \cdot \mathrm{ReLU}(\mathrm{Im}\{\mathbf{x}\})$. With input size of $H \times W$, successive CConvs with stride 2 (green blocks) are introduced to downscale the size to $\frac{H}{2} \times \frac{W}{2}$ and $\frac{H}{4} \times \frac{W}{4}$, while two **c**omplex-valued **t**ransposed **conv**olution layers (CTConvs) with stride 2 (yellow blocks) are utilized to upsample the size from $\frac{H}{4} \times \frac{W}{4}$ to $\frac{H}{2} \times \frac{W}{2}$, and from $\frac{H}{2} \times \frac{W}{2}$ to $H \times W$. Other CConvs plotted within blue blocks convolute input with stride 1 to maintain the scale yet may change the feature channel numbers.

The total number of learnable parameters for a SEPN can be calculated as: $k^2(4F_1 + 2F_1^2 + 4F_1F_2 + 2F_2^2 + 2F_2F_3 + 2F_3^2)$, where F_1, F_2, F_3 denote the numbers of feature channels, and k represents the convolutional kernel size. In the experiments for DPNN-S and DPNN-M, we set $F_1 = 4, F_2 = 8, F_3 = 16$, and $k = 5$; thus, the parameter number is 26,800. As for the MPNN, we construct two SEPNs with different scales. The lighter one was configured with $F_1 = 4, F_2 = 6, F_3 = 8$ and $k = 3$ with a parameter number of 3,960, and F_1, F_2, F_3, k for the other were set to 4, 8, 16, 3 with a parameter number of 9,648. Compared with UNet [33] with a parameter number of 7,765,442, the SEPNs are lighter and can be efficiently optimized.

4.7 Description of systematic errors

We considered four types of errors that occurred in DPNN practical systems, including the phase shift error that causes biased phase modulations and the geometric errors containing *Z-Axis* shift error, *X-Axis* shift error, and *XY-Plane* rotation error. *Z-Axis* shift error denotes the propagation distance error of optical diffraction, *X-Axis* shift error denotes the upper shift error of the phase modulation layers and the output plane, and *XY-Plane* rotation error represents the rotation deviation of the phase modulation layers and the output plane, described using angles. In addition, we modeled the phase shift error values to be independently sampled from a

normal distribution with zero mean and std σ. We include the geometric errors in a layer-by-layer manner, each with the same amount of errors. For example, setting *Z-Axis* shift error to 1 cm in DPNN-S means that each of $\mathbf{W}_{11}, \mathbf{W}_{12}, \mathbf{W}_{13}$ deviates with an extra propagation distance 1 cm. Thus, the holistic emphZ-Axis shift error is 3 cm. Setting *XY-Plane* shift error to 1 degree in DPNN-M means that the two-phase modulation layers and the output plane rotate with 1 degree relative to the previous layer or plane for every PNN block. Thus, the holistic *XY-Plane* shift error accumulates to 9 degrees.

As for the MPNN, there are mainly two types of systematic errors, i.e., beamsplitter error and phase shifter error [36, 40]. Beamsplitter error is caused by imperfect beamsplitters with split ratio errors that change the behavior of the perfect 50 : 50 coupling regions, the formulation of which is described in Supplementary Eq. C5. The phase shifter error can affect the value of ϕ and θ, leading to uncertainties of phase modulation. Various error sources may result in the slight performance deviance of phase shifters in the photonic meshes, such as thermal crosstalk or environmental drift [36]. Following the same assumptions made for DPNN, the beamsplitter and phase shifter error values were independently sampled from a normal distribution with zero mean and std σ_{bs}/σ_{ps}, and the errors are included in all devices and share the same strength. For example, setting $\sigma_{ps} = 0.1$ means that all the 4,032 phase shifters are corrupted by such a phase shifter error for the MPNN with $N = 3$.

4.8 Separable mode for training SEPNs

For a N-layer PNN with input $\mathbf{I}$, we can obtain the observations $\{\mathbf{P}_n\}_{n=1}^N$ from the physical system by measuring internal states and final output, with which the counterparts $\{\bar{\mathbf{S}}_n\}_{n=1}^N$ from the numerical model can be extracted. Here, $\{\bar{\mathbf{S}}_n\}_{n=1}^N$ are not equal to the $\{\mathbf{S}_n\}_{n=1}^N$ obtained by unitary inference of the numerical model with the same initial input $\mathbf{I}$. The $\{\mathbf{S}_n\}_{n=1}^N$ obtained by unitary inference are not appropriate to be the extracted internal states and output to approximate the targets $\{\mathbf{P}_n\}_{n=1}^N$ in the separable training mode, as the inputs of the n-th group for obtaining $\mathbf{S}_n$ and $\mathbf{P}_n$ are mismatched. To address this issue, we propose to extract $\{\bar{\mathbf{S}}_n\}_{n=1}^N$ with a separable inference of the numerical model. Specifically, for the n-th group, the input is replaced by the corresponding practical intensity, then $\bar{\mathbf{S}}_n$ is generated by the inference of the group with the replaced input. This process is repeated N times to produce $\{\bar{\mathbf{S}}_n\}_{n=1}^N$. The extraction of $\{\bar{\mathbf{S}}_n\}_{n=1}^N$ for the DPNN-M is illustrated in Supplementary Appendix E and Supplementary Fig. S1. Nevertheless, $\{\mathbf{S}_n\}_{n=1}^N$ is still indispensable for the optimization of the physical model, although it is not used for the separable training of SEPNs. In addition, although the process is based on the assumption that all the internal states are measured, it can be easily extended to the scenario with partial intensity measurements.

For optimizing the SEPNs in a separable mode, the PNN numerical model can be separated into N groups, where the n-th group corresponds to the paired data $(\mathbf{P}_n, \bar{\mathbf{S}}_n)$ and governs SEPNs within the group. For example, the DPNN-M can be divided into seven groups where the n-th group governs three SEPNs within the n-th PNN block. We denote the parameters of SEPNs in the n-th group by Λ_n and the similarity loss function for the n-th group by $L_{\mathrm{s,n}}$, we have: $L_{\mathrm{s,n}}(\mathbf{P}_n, |\bar{\mathbf{S}}_n|^2) = l_{\mathrm{mse}}(\mathbf{P}_n, |\bar{\mathbf{S}}_n|^2) = \|\mathbf{P}_n - |\bar{\mathbf{S}}_n|^2\|_2^2$. For each training sample, the parameters of the PNN physical model are fixed, and the gradients of $L_{\mathrm{s,n}}$ with respect to Λ_n are calculated during the backpropagation to iteratively and separably optimize the SEPNs

within the n-th group for $n \in [1, N]$. The pseudo-code of training DPNN in the separable mode is provided in Supplementary Algorithm S2.

4.9 DAT with internal states for DPNN

We elaborate on DAT with internal states for training DPNN by cascading multiple blocks in Fig. 2a, termed DPNN-C, to show the principle, which could be easily extended to DPNN-S and DPNN-M. The standard backpropagation process for DPNN-C is established in Supplementary Appendix B. Extended Data Fig. 5 depicts the procedure of DAT with internal states for training the n-th PNN block, where SEPN_{nk} for $k = 1, 2, 3$ represents the SEPN attached to corresponding diffractive propagation layers, $\mathbf{W}'_{ni}$ and $\mathbf{W}_{ni}$ for $i = 1, 2, 3$ denote the ideal and practical diffractive weight matrices, $\mathbf{S}_n$ and $\mathbf{P}_n$ denote the simulated and practical output intensity of the n-the block, $\mathbf{M}'_{n1}, \mathbf{M}'_{n2}$ and $\mathbf{M}_{n1}, \mathbf{M}_{n2}$ represent the ideal and practical phase modulation matrices, respectively. Specifically, $\mathbf{M}'_{nk} = \text{diag}(e^{2\pi j \mathbf{\Phi}_{nk}})$ for $k = 1, 2$ denotes the diagonalization of the vectorized phase modulation layer with coefficient $\mathbf{\Phi}_{nk}$, and $\mathbf{M}_{nk} = \text{diag}(e^{2\pi j(\mathbf{\Phi}_{nk}+\epsilon_{nk})})$, where j denotes the imaginary unit and ϵ_{nk} denotes the phase shift error. Mathematically, the forward propagation of the n-th block for DPNN-C physical system with N blocks can be formulated as follows based on the Rayleigh-Sommerfeld diffraction principle,

$$\begin{aligned} \mathbf{U}_n &= \mathbf{W}_{n3}\mathbf{M}_{n2}\mathbf{W}_{n2}\mathbf{M}_{n1}\mathbf{W}_{n1}\mathbf{P}_{n-1}, \\ \mathbf{P}_n &= |\mathbf{U}_n|^2, \end{aligned} \tag{3}$$

and the corresponding forward numerical model with SEPNs can be formulated as:

$$\begin{aligned} \mathbf{U}'_{n1} &= \mathbf{M}'_{n1}\left[\mathcal{N}_{n1}(\mathbf{W}'_{n1}\mathbf{O}_{n-1}) + \mathbf{W}'_{n1}\mathbf{O}_{n-1}\right], \\ \mathbf{U}'_{n2} &= \mathbf{M}'_{n2}\left[\mathcal{N}_{n2}(\mathbf{W}'_{n2}\mathbf{U}'_{n1}) + \mathbf{W}'_{n2}\mathbf{U}'_{n1}\right], \\ \mathbf{S}_n &= \mathcal{N}_{n3}(\mathbf{W}'_{n3}\mathbf{U}'_{n2}) + \mathbf{W}'_{n3}\mathbf{U}'_{n3}, \\ \mathbf{O}_n &= |\mathbf{S}_n|^2, \end{aligned} \tag{4}$$

where $\mathbf{U}_n, \mathbf{U}'_{n1}, \mathbf{U}'_{n2}$ represent the vectorized complex optical fields; $\mathbf{O}_n$ denotes the intensity of $\mathbf{S}_n$; $\mathcal{N}_{n1}, \mathcal{N}_{n2}, \mathcal{N}_{n3}$ denote the functions expressed by the SEPNs; $\mathbf{P}_0 = \mathbf{O}_0 = \mathbf{I}$ denote the initial input. The formulation is based on the residual connections [32] for incorporating SEPNs into the physical model.

Four steps of DAT are repeated over all training samples to minimize the loss functions until convergence for obtaining the numerical model and physical parameters, i.e., the phase modulation matrices $\mathbf{M}_{ni}$ for $i = 1, 2, 3$, $1 \leq n \leq N$. We elaborate on the steps with one training sample as follows:

First, the optically encoded I is input to the physical system to perform the forward inference. In this pass, we obtain the internal states $\mathbf{P}_n$ for $n \in [1, N-1]$ and final output intensity $\mathbf{P}_N$. The blue dotted arrows in Extended Data Fig. 5 describe this step for the n-the PNN block, corresponding to Eq. (3).

Second, the same training sample I is digitally encoded and input to the numerical model to extract the internal states and final observation $\mathbf{S}_n$ for $n \in [1, N]$. The yellow dotted arrows in Extended Data Fig. 5 describe this step for the n-the PNN block, corresponding to Eq. (4).

Third, we optimize the SEPNs' parameters Λ by minimizing the similarity loss function in the unitary mode as Eq. (1) or separable mode described in Methods 4.8 and Supplementary Appendix E. By the way, the similarity loss function simplifies to $L_s(\mathbf{P}, |\mathbf{S}|^2) = \|\mathbf{P}_N - |\mathbf{S}_N|^2\|_2^2$ for DAT if without internal states. The gradients of

the similarity loss function with respect to Λ are calculated via the backpropagation to optimize the SEPNs, while the parameters of the physical model are fixed. The red dotted arrows in Extended Data Fig. 5 describe this step for the n-the PNN block.

Fourth, we implement state fusion to obtain new internal states and the final observation for generating the new gradients of the task loss with respect to the phase modulation matrices. Specifically, for any $n \in [1, N]$, $\mathbf{P}_n$ can contribute to the replacement of $|\mathbf{S}_n|^2$ with $\mathbf{P}_n$, and the fusion of $\mathbf{S}_n$ and $\mathbf{P}_n$ using the fusion function $F_n(\mathbf{P}_n, \mathbf{S}_n) = \sqrt{\mathbf{P}_n}\exp(j\mathbf{\Phi}_{\mathbf{S}_n})\}_{n=1}^{N}$. The new internal states and observation are utilized to calculate the gradients to optimize the physical parameters of the PNN numerical model by minimizing the task loss L_t as Eq. (2). Therefore, the gradients of L_t with respect to $\mathbf{\Phi}_{nk}$ for $k = 1, 2$ are derived as:

$$\frac{\partial L_\mathrm{t}}{\partial \mathbf{\Phi}_{n1}} = 2\mathrm{Re}\{\frac{\partial L_\mathrm{t}}{\partial \mathbf{S}_n}\frac{\partial \mathbf{S}_n}{\partial \mathbf{M}'_{n1}}\frac{\partial \mathbf{M}'_{n1}}{\partial \mathbf{\Phi}_{n1}}\}, \ \frac{\partial L_\mathrm{t}}{\partial \mathbf{\Phi}_{n2}} = 2\mathrm{Re}\{\frac{\partial L_\mathrm{t}}{\partial \mathbf{S}_n}\frac{\partial \mathbf{S}_n}{\partial \mathbf{M}'_{n2}}\frac{\partial \mathbf{M}'_{n2}}{\partial \mathbf{\Phi}_{n2}}\}, \tag{5}$$

where

$$\frac{\partial L_\mathrm{t}}{\partial \mathbf{S}_n} = \begin{cases} \frac{\partial L_\mathrm{t}}{\partial \mathbf{O}_N} \odot (\mathbf{S}_N)^*, & n = N, \\ 2\mathrm{Re}\{\frac{\partial L_\mathrm{t}}{\partial \mathbf{S}_{n+1}}\frac{\partial \mathbf{S}_{n+1}}{\partial \mathbf{O}_n}\} \odot (\mathbf{S}_n)^*, & 1 \le n \le N-1; \end{cases} \tag{6}$$

$\odot$ is element-wise multiplication; $*$ is conjugation of the optical field; $\mathrm{Re}\{\cdot\}$ means reserving the real part of the optical field; the form of $\partial L_\mathrm{t}/\partial \mathbf{O}_N$ is related to the task loss function. We omit the detailed formulations of $\frac{\partial \mathbf{S}_n}{\partial \mathbf{\Phi}_{nk}}$ and $\frac{\partial \mathbf{S}_{n+1}}{\partial \mathbf{O}_n}$ as the forms are complicated when adopting SEPNs. Nevertheless, they can be easily deduced with reference to the standard propagation of DPNN-C in Supplementary Appendix B. Then, one gradient descent step is performed to optimize the phase modulation coefficients using the calculated gradients while the parameters of SEPNs are fixed. The green dotted arrows in Extended Data Fig. 5 describe this step for the n-the PNN block.

4.10 DAT without internal states for MPNN

We elaborate on DAT without internal states for the MPNN illustrated in Fig. 4a. The standard backpropagation was established in Supplementary Appendix C. Extended Data Fig. 6 illustrates the procedure of DAT without internal states for training the MPNN, where $\mathbf{I} \in \mathbb{C}^L$ denotes the input complex optical filed, $\hat{\mathbf{M}}'_n$ and $\hat{\mathbf{M}}_n$ represent the ideal and practical transformation matrix of the n-th photonic mesh that consisting of $L(L-1)/2$ embedded MZIs, and $\mathbf{Z}'_n$ and $\mathbf{Z}_n$ for $1 \le n \le N$ represent the simulated and practical output of the n-th photonic mesh, respectively. Mathematically, the forward propagation of the MPNN physical system can be described as:

$$\begin{aligned} \mathbf{Z}_1 &= \hat{\mathbf{M}}_1\mathbf{I}, \\ \mathbf{Z}_n &= \hat{\mathbf{M}}_n f_{EO}(\mathbf{Z}_{n-1}), \ 2 \le n \le N, \end{aligned} \tag{7}$$

where $f_{EO}(\cdot)$ is the optoelectronic nonlinear activation function introduced in [39] that can be formulated as $f_{EO}(\mathbf{Z}) = j\sqrt{1-\alpha}\exp[-j(\beta|\mathbf{Z}|^2+\gamma)]\cos(\beta|\mathbf{Z}|^2+\gamma)\mathbf{Z}$, and α, β, γ are constants related to configurations of the optoelectronic unit. The corresponding forward numerical model of the MPNN can be formulated as:

$$\begin{aligned} \mathbf{Z}'_1 &= \mathcal{N}_1(\hat{\mathbf{M}}'_1\mathbf{I}) + \hat{\mathbf{M}}'_1\mathbf{I}, \\ \mathbf{Z}'_n &= \mathcal{N}_n(\hat{\mathbf{M}}'_n f_{EO}(\mathbf{Z}'_{n-1})) + \hat{\mathbf{M}}'_n f_{EO}(\mathbf{Z}'_{n-1}), \ 2 \le n \le N, \end{aligned} \tag{8}$$

where $\mathcal{N}_n$ denotes the function expressed by SEPN_n incorporated into the physical model with residual connections [32].

Four steps for DAT without internal states are repeated over all training samples to minimize the loss functions and optimize the physical model, i.e., the phase coefficients $\mathbf{\Theta}_n$ and $\mathbf{\Phi}_n$ of the n-th photonic mesh for $1 \leq n \leq N$, where each $\mathbf{\Theta}_n$ or $\mathbf{\Phi}_n$ contain $L(L-1)/2$ coefficients. We detail the steps for one training sample as follows:

First, $\mathbf{I}$ is optically encoded and inputted to the physical system through waveguide ports to implement a forward inference, with which we measure the final output intensity $\mathbf{P}_N = |\mathbf{Z}_N|^2$. The blue dotted arrows in Extended Data Fig. 6 describe this step.

Second, $\mathbf{I}$ is digitally encoded and input to the numerical model for inference. To be consistent with $\mathbf{P}_N$, we only extract the complex optical field $\mathbf{S}_N = \mathbf{Z}'_N$. The yellow dotted arrows in Extended Data Fig. 6 describe this step.

Third, we simultaneously optimize all the SEPNs by minimizing the similarity loss function $L_\mathrm{s}(\mathbf{P}, |\mathbf{S}|^2) = \|\mathbf{P}_N - |\mathbf{S}_N|^2\|_2^2$. The gradients of L_s with respect to the parameters of SEPNs are calculated during the backpropagation to optimize the SEPNs in the unitary mode for one step, while the parameters of the physical model are fixed. The red dotted arrows in Extended Data Fig. 6 describe this step.

Fourth, we fuse $\mathbf{P}_N$ and $\mathbf{S}_N$ to obtain $\sqrt{\mathbf{P}_N}\exp(j\mathbf{\Phi}_{\mathbf{S}_N})$ to replace $\mathbf{S}_N$, and directly replace the simulated intensity $|\mathbf{S}_N|^2$ by $\mathbf{P}_N$. The new states and the replaced output are utilized to calculate the gradients of the task loss L_t as Eq. (2) with respect to the phase coefficients $\mathbf{\Theta}, \mathbf{\Phi}$ of all MZIs via backpropagation through the numerical model. Specifically, we have for $n \in [1, N]$ that

$$\frac{\partial L_\mathrm{t}}{\partial \mathbf{\Theta}_n} = 2\mathrm{Re}\left\{\left(\frac{\partial L_\mathrm{t}}{\partial |\mathbf{Z}'_N|^2} \odot (\mathbf{Z}'_N)^*\right)^T \frac{\partial \mathbf{Z}'_N}{\partial \hat{\mathbf{M}}'_n} \frac{\partial \hat{\mathbf{M}}'_n}{\partial \mathbf{\Theta}_n}\right\}, \tag{9}$$

where $\odot$, $*$ and $\mathrm{Re}\{\cdot\}$ are defined in Eq (6); the forms of $\partial L_\mathrm{t}/\partial|\mathbf{Z}'_N|^2$ and $\partial\hat{\mathbf{M}}'_n/\partial\mathbf{\Theta}_n$ are related to the task loss function and the scheme to construct MPNN, respectively. Similarly, the gradient of L_t with respect to $\mathbf{\Phi}_n$ can be easily deduced. The detail form of $\frac{\partial \mathbf{Z}'_N}{\partial \hat{\mathbf{M}}'_n}$ is omitted due to the complexity of its formulation with SEPNs but can be deduced with reference to the standard propagation of the MPNN in Supplementary Appendix C. Then, one gradient descent step is implemented to optimize the phase coefficients, while the parameters of SEPNs are fixed. The green dotted arrows in Extended Data Fig. 6 describe this step.

Data availability All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials.

Code availability The code for the construction of MPNN is available at https://github.com/solgaardlab/neurophox and the code for training MPNN with DAT will be released soon.

References

[1] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

[2] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations, (2015)

[3] Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: A comprehensive review. Neural Computation **29**(9), 2352–2449 (2017)

[4] Capper, D., Jones, D.T., Sill, M., Hovestadt, V., Schrimpf, D., Sturm, D., Koelsche, C., Sahm, F., Chavez, L., Reuss, D.E., *et al.*: Dna methylation-based classification of central nervous system tumours. Nature **555**(7697), 469–474 (2018)

[5] Torlai, G., Mazzola, G., Carrasquilla, J., Troyer, M., Melko, R., Carleo, G.: Neural-network quantum state tomography. Nature Physics **14**(5), 447–450 (2018)

[6] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., Dean, J.: Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350 (2021)

[7] Shainline, J.M., Buckley, S.M., Mirin, R.P., Nam, S.W.: Superconducting optoelectronic circuits for neuromorphic computing. Physical Review Applied **7**(3), 034013 (2017)

[8] Tait, A.N., De Lima, T.F., Nahmias, M.A., Shastri, B.J., Prucnal, P.R.: Continuous calibration of microring weights for analog optical networks. IEEE Photonics Technology Letters **28**(8), 887–890 (2016)

[9] Shi, B., Calabretta, N., Stabile, R.: Deep neural network through an inp soa-based photonic integrated cross-connect. IEEE Journal of Selected Topics in Quantum Electronics **26**(1), 1–11 (2019)

[10] Feldmann, J., Youngblood, N., Wright, C.D., Bhaskaran, H., Pernice, W.H.: All-optical spiking neurosynaptic networks with self-learning capabilities. Nature **569**(7755), 208–214 (2019)

[11] Chakraborty, I., Saha, G., Roy, K.: Photonic in-memory computing primitive for spiking neural networks using phase-change materials. Physical Review Applied **11**(1), 014063 (2019)

[12] Shen, Y., Harris, N.C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., *et al.*: Deep learning with coherent nanophotonic circuits. Nature Photonics **11**(7), 441–446 (2017)

[13] Gholipour, B., Bastock, P., Craig, C., Khan, K., Hewak, D., Soci, C.: Amorphous metal-sulphide microfibers enable photonic synapses for brain-like computing. Advanced Optical Materials **3**(5), 635–641 (2015)

[14] Xu, X., Tan, M., Corcoran, B., Wu, J., Boes, A., Nguyen, T.G., Chu,

S.T., Little, B.E., Hicks, D.G., Morandotti, R., *et al.*: 11 tops photonic convolutional accelerator for optical neural networks. Nature **589**(7840), 44–51 (2021)

[15] Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A.S., *et al.*: Parallel convolutional processing using an integrated photonic tensor core. Nature **589**(7840), 52–58 (2021)

[16] Hughes, T.W., England, R.J., Fan, S.: Reconfigurable photonic circuit for controlled power delivery to laser-driven accelerators on a chip. Physical Review Applied **11**(6), 064014 (2019)

[17] Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M., Englund, D.: Large-scale optical neural networks based on photoelectric multiplication. Physical Review X **9**(2), 021032 (2019)

[18] Lin, X., Rivenson, Y., Yardimci, N.T., Veli, M., Luo, Y., Jarrahi, M., Ozcan, A.: All-optical machine learning using diffractive deep neural networks. Science **361**(6406), 1004–1008 (2018)

[19] Yan, T., Wu, J., Zhou, T., Xie, H., Xu, F., Fan, J., Fang, L., Lin, X., Dai, Q.: Fourier-space diffractive deep neural network. Physical Review Letters **123**(2), 023901 (2019)

[20] Zhou, T., Lin, X., Wu, J., Chen, Y., Xie, H., Li, Y., Fan, J., Wu, H., Fang, L., Dai, Q.: Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. Nature Photonics **15**(5), 367–373 (2021)

[21] Hughes, T.W., Williamson, I.A., Minkov, M., Fan, S.: Wave physics as an analog recurrent neural network. Science Advances **5**(12), 6946 (2019)

[22] Bueno, J., Maktoobi, S., Froehly, L., Fischer, I., Jacquot, M., Larger, L., Brunner, D.: Reinforcement learning in a large-scale photonic recurrent neural network. Optica **5**(6), 756–760 (2018)

[23] Van der Sande, G., Brunner, D., Soriano, M.C.: Advances in photonic reservoir computing. Nanophotonics **6**(3), 561–576 (2017)

[24] Larger, L., Baylón-Fuentes, A., Martinenghi, R., Udaltsov, V.S., Chembo, Y.K., Jacquot, M.: High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. Physical Review X **7**(1), 011015 (2017)

[25] Brunner, D., Penkovsky, B., Marquez, B.A., Jacquot, M., Fischer, I., Larger, L.: Tutorial: Photonic neural networks in delay systems. Journal

of Applied Physics **124**(15), 152004 (2018)

[26] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[27] Zuo, Y., Li, B., Zhao, Y., Jiang, Y., Chen, Y.-C., Chen, P., Jo, G.-B., Liu, J., Du, S.: All-optical neural network with nonlinear activation functions. Optica **6**(9), 1132–1137 (2019)

[28] Wright, L.G., Onodera, T., Stein, M.M., Wang, T., Schachter, D.T., Hu, Z., McMahon, P.L.: Deep physical neural networks trained with backpropagation. Nature **601**(7894), 549–555 (2022)

[29] Hughes, T.W., Minkov, M., Shi, Y., Fan, S.: Training of photonic neural networks through in situ backpropagation and gradient measurement. Optica **5**(7), 864–871 (2018)

[30] Hughes, T., Veronis, G., Wootton, K.P., England, R.J., Fan, S.: Method for computationally efficient design of dielectric laser accelerator structures. Optics Express **25**(13), 15414–15427 (2017)

[31] Zhou, T., Fang, L., Yan, T., Wu, J., Li, Y., Fan, J., Wu, H., Lin, X., Dai, Q.: In situ optical backpropagation training of diffractive optical neural networks. Photonics Research **8**(6), 940–953 (2020)

[32] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[33] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241 (2015). Springer

[34] Yamaguchi, I.: Phase-shifting digital holography. Digital Holography and Three-Dimensional Display, 145–171 (2006)

[35] Shokraneh, F., Nezami, M.S., Liboiron-Ladouceur, O.: Theoretical and experimental analysis of a 4× 4 reconfigurable mzi-based linear optical processor. Journal of Lightwave Technology **38**(6), 1258–1267 (2020)

[36] Pai, S., Bartlett, B., Solgaard, O., Miller, D.A.: Matrix optimization on universal unitary photonic devices. Physical Review Applied **11**(6), 064044 (2019)

[37] Reck, M., Zeilinger, A., Bernstein, H.J., Bertani, P.: Experimental realization of any discrete unitary operator. Physical Review Letters **73**(1), 58 (1994)

[38] Clements, W.R., Humphreys, P.C., Metcalf, B.J., Kolthammer, W.S., Walmsley, I.A.: Optimal design for universal multiport interferometers. Optica **3**(12), 1460–1465 (2016)

[39] Williamson, I.A., Hughes, T.W., Minkov, M., Bartlett, B., Pai, S., Fan, S.: Reprogrammable electro-optic nonlinear activation functions for optical neural networks. IEEE Journal of Selected Topics in Quantum Electronics **26**(1), 1–12 (2019)

[40] Pai, S., Williamson, I.A.D., Hughes, T.W., Minkov, M., Solgaard, O., Fan, S., Miller, D.A.B.: Parallel programming of an arbitrary feedforward photonic network. IEEE Journal of Selected Topics in Quantum Electronics **26**(5), 1–13 (2020)

[41] Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y., Pal, C.J.: Deep complex networks. In: International Conference on Learning Representations (2018)

[42] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

[43] Chang, J., Sitzmann, V., Dun, X., Heidrich, W., Wetzstein, G.: Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. Scientific Reports **8**(1), 1–10 (2018)

[44] Miscuglio, M., Hu, Z., Li, S., George, J.K., Capanna, R., Dalir, H., Bardet, P.M., Gupta, P., Sorger, V.J.: Massively parallel amplitude-only fourier neural network. Optica **7**(12), 1812–1819 (2020)

[45] Spall, J., Guo, X., Lvovsky, A.I.: Hybrid training of optical neural networks. Optica **9**(7), 803–811 (2022)

[46] Xu, X., Ren, G., Feleppa, T., Liu, X., Boes, A., Mitchell, A., Lowery, A.J.: Self-calibrating programmable photonic integrated circuits. Nature Photonics **16**(8), 595–602 (2022)

[47] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations, (2015)

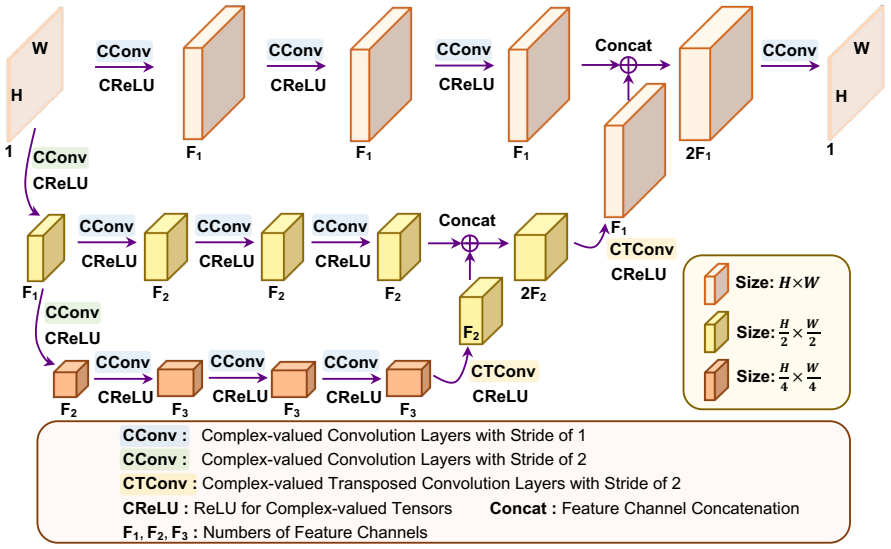
[48] Filipovich, M.J., Guo, Z., Al-Qadasi, M., Marquez, B.A., Morison, H.D.,

Sorger, V.J., Prucnal, P.R., Shekhar, S., Shastri, B.J.: Monolithic silicon photonic architecture for training deep neural networks with direct feedback alignment. arXiv preprint arXiv:2111.06862 (2021)

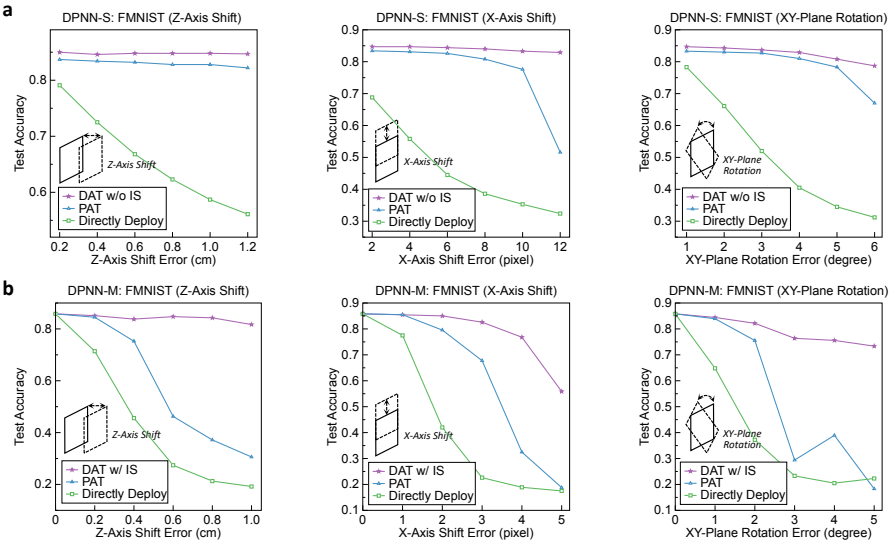
Acknowledgements This work is supported by the National Key Research and Development Program of China (No. 2021ZD0109902) and the National Natural Science Foundation of China (No. 62275139 and No. 61932022).

Author contributions X.L. and H.X. initiated and supervised the project. X. L., Z.Z., and Z.D. conceived the research. X. L. designed the methods. Z.Z. and Z.D. implemented the algorithm and conducted experiments. Z.Z., Z.D., H.C., R.Y., S.G., and H.Z. processed the data. X.L., Z.Z., Z.D., H.C., and R.Y. analyzed and interpreted the results. All authors prepared the manuscript and discussed the research.

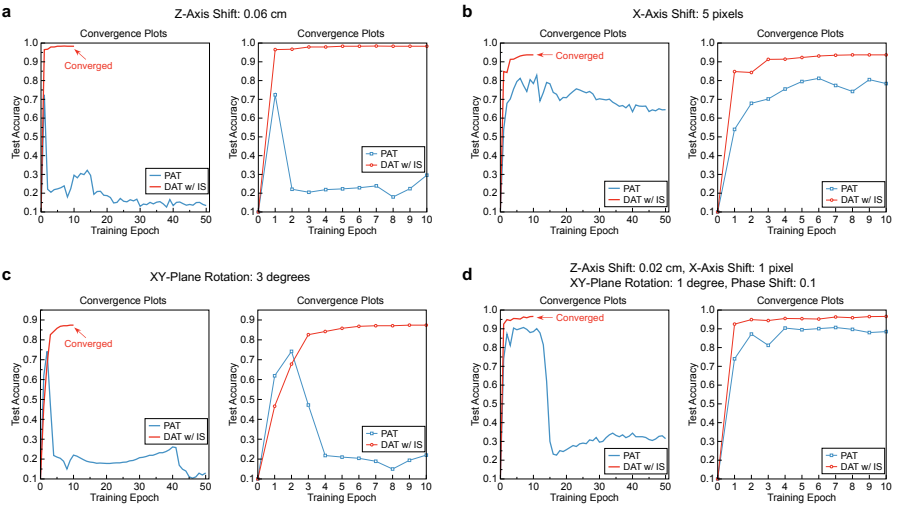
Competing interests The authors declare no competing interests.



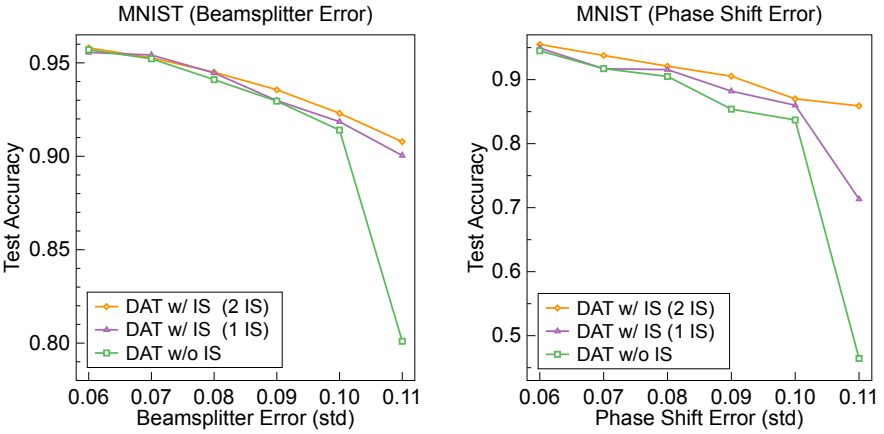
Extended Data Fig. 1 Architecture of the SEPN constructed with a complex-valued mini-UNet. See Methods for the detailed description.



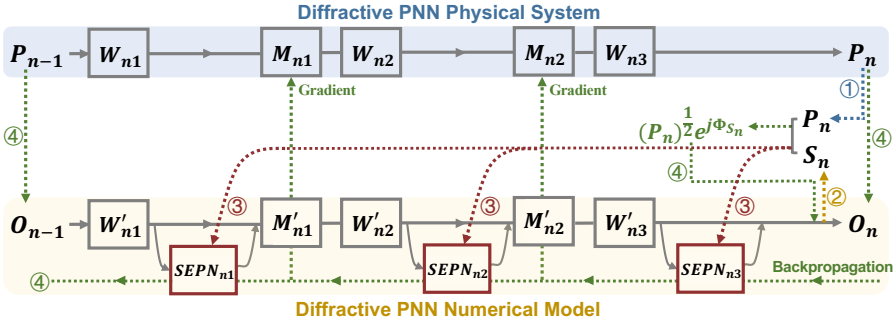
Extended Data Fig. 2 Training DPNNs under three types of systematic errors for the FMNIST classification. The performances of DAT for DPNN-S (**a**) and DPNN-M (**b**) are compared with the PAT and direct deployment of *in silico* trained models under different amounts of systematic errors. The DPNN-S and DPNN-M are trained without and with the internal states (IS), respectively.



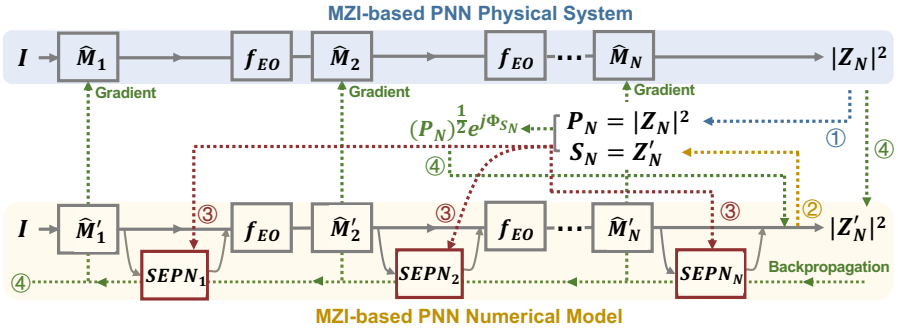
Extended Data Fig. 3 Convergence plots of the DPNN-M evaluated on the MNIST blind-test dataset during the training process. Each subfigure consists of convergence plots with 50 (total training epochs for PAT, left) and 10 (total training epochs for DAT, right) epochs, where **a,b,c** represent the training process under the individual errors and **d** under the joint errors, with the error configurations shown above the subfigures. DAT outperforms PAT with a more robust training process for the optimization of the DPNN-M.



Extended Data Fig. 4 Comparisons of DAT performances with all, partial, and without internal states for the 3-layer MPNN in the task of MNIST classification. The DAT methods are implemented with each SEPN parameter of 9,648 in the unitary mode. The performance of DAT with all internal states $\mathbf{P}_1, \mathbf{P}_2$ (2 IS), one internal state $\mathbf{P}_2$ (1 IS), and without internal states are evaluated. The classification accuracy improves with more measurements of internal states, especially under severe systematic errors.



Extended Data Fig. 5 Procedure of DAT with internal states for DPNN in the n-th PNN block. The flow charts with blue and yellow backgrounds denote the forward inferences in the physical system and the numerical model, respectively. Four steps of DAT with internal states, labeled using dotted arrows with four different colors, are repeated over all training samples to minimize the loss functions until convergence for obtaining the numerical model and physical parameters for the system, i.e., the phase modulation matrices $\mathbf{M}_{ni}$ for $i = 1, 2, 3$, $1 \leq n \leq N$. See Methods for the detailed description.



Extended Data Fig. 6 Procedure of DAT without internal states for the MPNN. The flow charts with blue and yellow backgrounds denote the forward inferences in the physical system and the numerical model, respectively. Four steps of DAT without internal states, labeled using dotted arrows with four different colors, are repeated over all training samples to minimize the loss functions and optimize the physical model, i.e., the phase coefficients $\mathbf{\Theta}_n$ and $\mathbf{\Phi}_n$ of the n-th photonic mesh for $1 \leq n \leq N$. See Methods for the detailed description.