Supporting Information

Partially coherent high-order optical neural networks via pseudo-nonlinear encoding

Bojie Dong[1], Xiaoxin Li[1,2*], Zixuan Wang[3], Qi Jia[1*], Bojian Shi[1], Shiyong Ren[1], Lin Huang[1], Hanyi Zhang[1], Zhengjun Wang[1], Rui Feng[1], Fangkui Sun[1], Yongyin Cao[1], Liangcai Cao[4*], Weiqiang Ding[1,5]*

[1]*Institute of Advanced Photonics, School of Physics, Harbin Institute of Technology, Harbin 150001, China*

[2] *Hangzhou International Innovation Institute, Beihang University, Hangzhou 310000, China*

[3]*Faculty of Computing, Harbin Institute of Technology, 150000, China*

[4]*Department of Precision Instrument, Tsinghua University, Beijing 100084, China*

[5] *Collaborative Innovation Center of Extreme Optics, Shanxi University, Taiyuan 030006, China*

[*]Correspondence: Xiaoxin Li (lixiaoxinbuaa@buaa.edu.cn), Qi Jia (jiaqi1350@hit.edu.cn), Liangcai Cao (clc@tsinghua.edu.cn), or Weiqiang Ding (wqding@hit.edu.cn)

Contents

S1. Simulation of quasi-monochromatic partially coherent light

Here we only focus on the spatial correlation of the optical field. Spatial coherence describes the correlation between the optical fields at two arbitrary spatial points at the same time. We use the complex random screen method [1–3] to generate the low-coherence Gaussian Schell-model beams[1–3]. For simplicity, we consider a quasi-monochromatic Gaussian Schell-model, and the coherence function only depends on the distance between the two points. The cross-spectral density matrix $\boldsymbol{\rho} = (\boldsymbol{E}\cdot\boldsymbol{E}^H)\circ\boldsymbol{\Gamma}$ and the matrix elements, which called density function, $\rho(r_n, r_m)$ is given by:

$$\begin{cases} \rho_{nm} = \rho(r_n, r_m) = \sqrt{I(r_n)}\sqrt{I(r_m)}\gamma(r_n - r_m) \\ \gamma_{nm} = \gamma(r_n - r_m) = \exp\left[-(r_n - r_m)^2 / (2l_c^2)\right] \end{cases}. \tag{1}$$

where r_n and r_m represent the positions of two points in space, $I(r_n) = E^*(r_n)\cdot E(r_n)$ and $I(r_m) = E^*(r_m)\cdot E(r_m)$ represent the intensities of the two points, $\gamma(r_n - r_m)$ is the degree of coherence, characterizing the degree of coherence between two points in space, and l_c represents the coherent length, $\boldsymbol{\Gamma}$ represents the coherence matrix which is related to coherence length l_c, '$\circ$' denotes the Hadamard product.

The working principle of the complex random screen method is to apply a random complex amplitude screen $\Psi(x, y)$ to the coherent light source field, and then propagate the coherent light field to the output plane. We repeat this process for independent random complex amplitude screens, and finally average the intensity patterns to obtain the transmission result of the incoherent light field. In the spatial frequency domain, the complex amplitude spectrum of the transmission screen can be expressed as:

$$\begin{cases} \Psi^F(u,v) = \mathcal{F}\left[\Psi(x,y)\right] = R(u,v)\cdot F(u,v) \\ R(u,v) = \dfrac{1}{\sqrt{\Delta u \Delta v}}\left[\dfrac{\mathrm{randn}(p,q) + j\cdot \mathrm{randn}(p,q)}{\sqrt{2}}\right] \\ S(u,v) = |F(u,v)|^2 \end{cases}. \tag{2}$$

$R(u, v)$ represents circular complex Gaussian noise where randn($\cdot$) represents a routine which generates a normally distributed, unit variance, random number. $F(u, v)$ represents the spectral function and $S(u, v)$ represents the power spectrum [2].

For the random complex amplitude screen $\Psi(x, y)$, its autocorrelation function R_{AA} has the following relationship with the degree of coherence γ:

$$\gamma(x', y') = \frac{< E_A(x + x', y + y')E_A^*(x, y) >}{< E_A(x, y)E_A^*(x, y) >} = \frac{R_{AA}(x', y')}{R_{AA}(0,0)}. \tag{3}$$

where the angle brackets indicate average operation. Under the normalization condition, autocorrelation value $R_{AA}(0, 0) = 1$ at the origin. The complex coherence function is numerically equivalent to the autocorrelation function, which means $R_{AA}(x, y) = \gamma(x, y)$. According to the Wiener-Khinchin theorem, the spatial autocorrelation function and the power spectral density form a Fourier transform pair for the Gaussian Schell model light source that satisfy the wide smoothness condition. Therefore, the power spectral density $S(u, v)$ can be directly calculated by the Fourier transform of the autocorrelation function $R_{AA}(x, y)$:

$$S(u,v)=\mathcal{F}[R(x,y)]=\mathcal{F}[\gamma(x,y)]=2\pi\delta_0^2\exp(-\pi^2 l_c^2(u^2+v^2)). \tag{4}$$

Thus, for the Gaussian Schell model, the random complex amplitude screen $\Psi(x, y)$, can be described as follows:

$$\begin{cases}\Psi(x,y)=\mathcal{F}^{-1}\left[R(u,v)\cdot\sqrt{S(u,v)}\right]\\ S(u,v)=2\pi l_c^2\exp(-\pi^2 l_c^2(u^2+v^2))\end{cases}. \tag{5}$$

S2. Detail principle of HONN

Complex amplitude input $\boldsymbol{E}=[E_1, E_2, \ldots, \quad E_N]^T$ and the optical output y is obtained by measuring the intensity of the light field. For the coherent diffraction, the entire process could be described as a linear transformation matrix $\boldsymbol{D}=\boldsymbol{H}_2\cdot\boldsymbol{T}\cdot\boldsymbol{H}_1$, where $\boldsymbol{T}$ is the diagonal modulation layer transfer matrix, and $\boldsymbol{H}$ is the transfer matrix of the diffraction process, of which matrix elements are given by the scalar diffraction theory:

$$h_{nm}^j=\frac{1}{2\pi}\frac{z_j}{r_{nm}}\left(\frac{1}{r_{nm}}-ik\right)\frac{\exp(ikr_{nm})}{r_{nm}}. \tag{6}$$

Here, z_j represents the diffraction distance, r_{nm} denotes the distance between two points, and k is the wave number of the light. Therefore, the optical output satisfies $\boldsymbol{y}=(\boldsymbol{D}\cdot\boldsymbol{E})^*\circ(\boldsymbol{D}\cdot\boldsymbol{E})$ where the '$\circ$' represents the Hadamard product.

Considering the spatial coherence, the density matrix of light field is $\boldsymbol{\rho}=(\boldsymbol{E}\cdot\boldsymbol{E}^H)\circ\boldsymbol{\Gamma}$, the output $\boldsymbol{y}$ is the diagonal elements of the matrix $\boldsymbol{O}=\boldsymbol{D}\cdot\boldsymbol{\rho}\cdot\boldsymbol{D}^H$. Each component are given below using the Einstein summation convention:

$$y_j=d_{jk}(\gamma_{kl}x_k x_l^*)d_{jl}^*, d_{jk}=h_{jn}^2 t_n h_{nk}^1. \tag{7}$$

Equation (7) can be rewritten as:

$$y_j=w_{jkl}(\gamma_{kl}x_k x_l^*), w_{jkl}=d_{jk}d_{jl}^*. \tag{8}$$

As $l_c \to 0$ then $\gamma_{kl}=\delta_{kl}$, implying that the light becomes completely incoherent and the modulation tensor w degenerates. The model then becomes linear:

$$y_j=w_{jkk}x_k x_k^*=w_{jkk}|x|_k^2=w_{jk}I_k, \tag{9}$$

and the modulation matrix w is:

$$w_{jk}=d_{jk}d_{jk}^*=h_{jm}^2 t_m h_{mk}^1 h_{jn}^{2*} t_n^* h_{nk}^{1*}. \tag{10}$$

While incoherent inputs lack phase information, they can still be modulated using a phase modulator.

As $l_c \to \infty$ then $\gamma_{kl}=1$, implying that the light becomes completely coherent and leading to a model in the form of a generalized second-order Laurent series:

$$y_j=w_{jkl}x_k x_l^*. \tag{11}$$

and the modulation tensor w is:

$$w_{jkl}=d_{jk}d_{jl}^*=h_{jm}^2 t_m h_{mk}^1 h_{jn}^{2*} t_n^* h_{nl}^{1*}. \tag{12}$$

For the system's nonlinearity, the modulation parameter t establishes a dependence on the input x, which means $t = t(x)$, and a higher-order nonlinearity can be achieved. In linear systems, repeat encoding could yield pseudo-nonlinearity. As amplitude modulation can lead to energy loss, we restrict our consideration to phase modulation. For simplicity, $t_n = \exp(jb_n + jax_n)$, with b_n is a trainable parameter. Equation (8) can be rewritten as:

$$d_{jk}d_{jl}^{*} = h_{jm}^{2}\exp(jt_m + jax_m)h_{mk}^{1}h_{jn}^{2*}\exp(-jt_n - jax_n)h_{nl}^{1*} \tag{13}$$

$$w_{jklmn} = h_{jm}^{2}\exp(jt_m)h_{mk}^{1}h_{jn}^{2*}\exp(-jt_n)h_{nl}^{1*} \tag{14}$$

$$y_j = w_{jklmn}(\gamma_{kl}x_k x_l)\exp(iax_n)\exp(-iax_m)$$

(15)

Equation (15) resulting formulation then constitutes a mixed series:

$$y_j = w_{jklmn}(\gamma_{kl}x_k x_l)\exp(iax_n)\exp(-iax_m) \tag{16}$$

By comparing the basic formulas of nonlinear optics $\boldsymbol{P} = \varepsilon_0\boldsymbol{\chi}^{(2)}:\boldsymbol{EE}$, we can further obtain the physical nonlinear formula under partially coherent light illumination as follows:

$$y_j = w_{jklmn}(\gamma_{kl}x_k^{*}x_l)(\gamma_{nm}x_n^{*}x_m)\,. \tag{17}$$

Pseudo nonlinearity, characterized by $\gamma_{nm} = 1$, retains more terms than physical nonlinearity. In contrast to physical nonlinearity—where parameters x_k, x_l, x_n and x_m are consistently encoded in phase or amplitude—pseudo nonlinearity achieves decoupling. Furthermore, it operates without an activation threshold.

S3. Derivation of backpropagation process for the DONN

In the main text, we derive the forward propagation process of coherent DONN, which can be summarised as:

$$\begin{cases}\vec{E}_l = \hat{P}_l \cdot \hat{T}_l \cdot \vec{E}_{l-1} \\ \hat{T}_l = \exp(\mathrm{j}\hat{\Phi}_l)\end{cases}. \tag{18}$$

where $\boldsymbol{T}_l$ represents the transmittance function of the l^{th} diffraction layer, which is a diagonal matrix; $\boldsymbol{E}_l$ represents is the incident light field of the l^{th} diffraction layer, P represents the angular spectrum propagation operator, $\boldsymbol{\Phi}_l$ represents the phase of the neuron in the l^{th} diffraction layer. After being modulated by N diffraction layers, the light field is transmitted to the output plane, and its light intensity distribution is the result of coherent DONN:

$$\vec{I}_{\text{out}} = \vec{E}_N^{*} \circ \vec{E}_N = \left|(\prod_{i=1}^{N}\hat{P}_i \cdot \hat{T}_i)\cdot \vec{E}_0\right|^2. \tag{19}$$

In training, the crossentropy is chosen as the loss function for optimization, which can be expressed as:

$$L = L(\vec{I}_{\text{out}}) = \frac{1}{N_x N_y}\sum_{x,y} I_{\text{tar}}(x,y)\log[1/I_{\text{out}}(x,y)] \tag{20}$$

Based on the mathematical definition of Equation (19), the gradient of the loss function L with respect to the trainable parameters $\boldsymbol{\Phi}_l$ can be derived by the backpropagation algorithm as[4]:

$$\begin{cases} \dfrac{\partial L}{\partial \hat{\Phi}_l} = \dfrac{\partial L}{\partial \vec{E}_N}\dfrac{\partial \vec{E}_N}{\partial \hat{\Phi}_l} + \dfrac{\partial L}{\partial \vec{E}_N^*}\dfrac{\partial \vec{E}_N^*}{\partial \hat{\Phi}_l} = 2\,\mathrm{Re}(\dfrac{\partial L}{\partial \vec{I}_{out}} \circ \vec{E}_N^*)^T \dfrac{\partial \vec{E}_N}{\partial \hat{\Phi}_l} \\ \dfrac{\partial \vec{E}_N}{\partial \hat{\Phi}_l} = j(\prod_{i=l+1}^{N} \hat{P}_i \cdot \hat{T}_i)\,\mathrm{diag}[(\prod_{i=i}^{l} \hat{P}_i \cdot \hat{T}_i)\cdot \vec{E}_0] = j\cdot\{\mathrm{diag}[\prod_{i=i}^{l} \hat{P}_i \cdot \hat{T}_i)\cdot \vec{E}_0]\}^T \times (\prod_{i=l+1}^{N} \hat{P}_i \cdot \hat{T}_i)^T \end{cases}. \tag{21}$$

Equation (19) can be rewritten as backpropagation form:

$$\begin{cases} \dfrac{\partial L}{\partial \hat{\Phi}_l} = 2\,\mathrm{Re}[j \cdot \vec{P}_l^f \circ \vec{P}_l^b]^T \\ \vec{P}_l^f = (\prod_{i=1}^{l} \hat{P}_i \cdot \hat{T}_i)\cdot \vec{E}_0 \\ \vec{P}_l^b = (\prod_{i=l+1}^{N} \hat{T}_i \cdot \hat{P}_i^T)\cdot \vec{\delta}, \vec{\delta} = \dfrac{\partial L}{\partial \vec{I}_{out}} \circ \vec{E}_N^* \end{cases} \tag{22}$$

In the main text, we also give the forward propagation of partially coherent DONN, which can be summarized as:

$$\vec{I}_{\text{out}} = \frac{1}{M}\sum_{n=1}^{M}\left|(\prod_{i=1}^{N} \hat{P}_i \cdot \hat{T}_i)\cdot(\Psi_n \circ \vec{E}_0)\right|^2 = \frac{1}{M}\sum_{n=1}^{M} \vec{E}_n^* \circ \vec{E}_n \tag{23}$$

where $\boldsymbol{\Psi}_n$ is the random complex amplitude screen used in the n^{th} coherent simulation procedure and $\boldsymbol{E}_n$ is the n^{th} output optical field distribution obtained in M coherent simulations. Therefore, The loss function L of PC-DONN can be expressed as:

$$L = L(\vec{I}_{\text{out}}) = \frac{1}{N_x N_y}\sum_{x,y} I_{\text{tar}}(x,y)\log[M / \sum\nolimits_{n=1}^{n=M} |E_n(x,y)|^2] \tag{24}$$

The mathematical framework for the backpropagation process of partially coherent DONN is essentially the same as that for coherent DONN, with the difference being that when the system obtains the n^{th} output light field distribution $\boldsymbol{E}_n$ through coherent simulation, we need to calculate the partial derivative of the loss function L with respect to each output light field $\boldsymbol{E}_n$. And the total loss is below:

$$\begin{cases} \dfrac{\partial L}{\partial \hat{\Phi}_l} = 2\,\mathrm{Re}[j \cdot \vec{P}_l^f \circ \vec{P}_l^b]^T \\ \vec{P}_l^f = \dfrac{1}{M}\sum_{n=1}^{M}(\prod_{i=1}^{l} \hat{P}_i \cdot \hat{T}_i)\cdot(\vec{\Psi}_n \circ \vec{E}_0) \\ \vec{P}_l^b = \dfrac{1}{M}\sum_{n=1}^{M}(\prod_{i=1}^{l} \hat{P}_i \cdot \hat{T}_i)\cdot(\dfrac{\partial L}{\partial \vec{I}_{out}} \circ \vec{E}_N^*) \end{cases} \tag{25}$$

Figure S1 shows the partially coherent diffraction results of different coherence length

l_c.

S4. Nonlinear function fitting for HONN

For simplicity, consider a 2-input system in Figure S2(a) which is a simple example of Equation (15). For the 1st-order, $t_n = \exp(jb_n)$; as for the 2nd-order, $t_n = \exp(jb_n + jax_n)$. For the coherent system, the 1st-order output can be estimated as $y_1 \approx (x_1 + x_2)^2$ and the 2nd-order output can be estimated as $y_2 \approx (x_1 + x_2)^2 \cos^2 (x_1/2 - x_2/2)$. Partially coherent 1st-order system should be hard to get square output $y_3 \approx (x_1 + x_2)^2$. Thus, three models (1st-order coherence, 2nd-order coherence, and 1st-order partial coherence) were numerically trained via the angular spectrum method, adopting y_1, y_2, and y_3 as target functions.

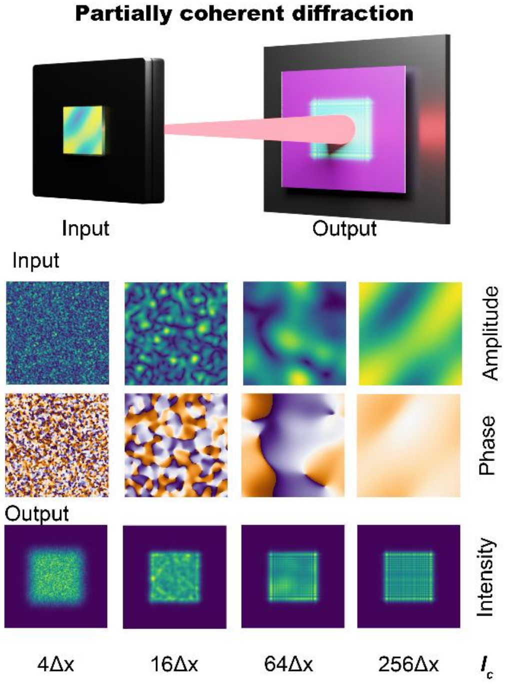

Figure S1. The diffraction of a 200×200 square light field at 30 mm under different degrees of correlation.

The 6 μm×6 μm light source contained 15×15 independent b_n parameters, with a diffraction distance of 0.02 mm. For partial coherence, the coherence length was 2 μm and M=30. Figure S2(b) plots the 2nd-order coherent system outputs y_{out} and the target nonlinear surface y_2. Figure S2(c) shows the 2D projection of the curve. Figure S2(d)-(f) illustrate the correlation between 1000 network outputs and three targets, yielding average errors of 0.0234 (y_1), 0.0369 (y_{22}), and 0.1877 (y_3). Error evaluation confirms the qualitative validity and physical rationality of the proposed theory.

S5. Image processing flowchart

The original MNIST image resolution is 28×28 pixels. We upsample it to 320×320 pixels. The image is then centered on a 400×400 pixel canvas, with zero-value padding (black background) around the surrounding areas, resulting in the final model input size (400×400 pixels). Figure S3 illustrates this processing workflow.

S6. Load input images under coherent illumination to SLM1

Figure S4 shows the image loaded onto SLM1 propagates to the input plane, forming the superimposed coherent field $\hat{\Psi}_m \cdot \vec{E}_{in}$, where the original coherent input $\vec{E}_{in}$ serves as the amplitude input. For the case illustrated, coherence length is fixed at $l_c = 64\Delta x$.

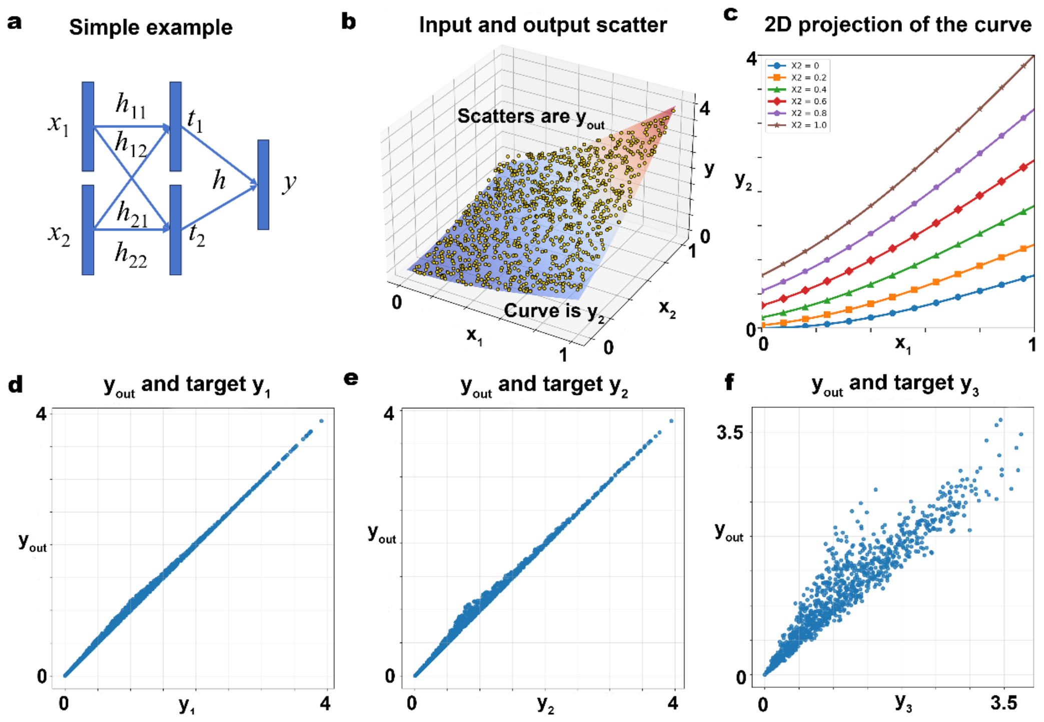

Figure S2. A simple model of HONN for nonlinear function fitting: (a) the 2-input model; (b) coherent 2st-order system output and target nonlinear curve; (c) 2D projection of the target nonlinear curve; (d) coherent 1st-order system output y_{out} and target output y_1; (e) coherent 2st-order system output y_{out} and target output y_2; (f) partially coherent 1st-order system output y_{out} and target output y_3.

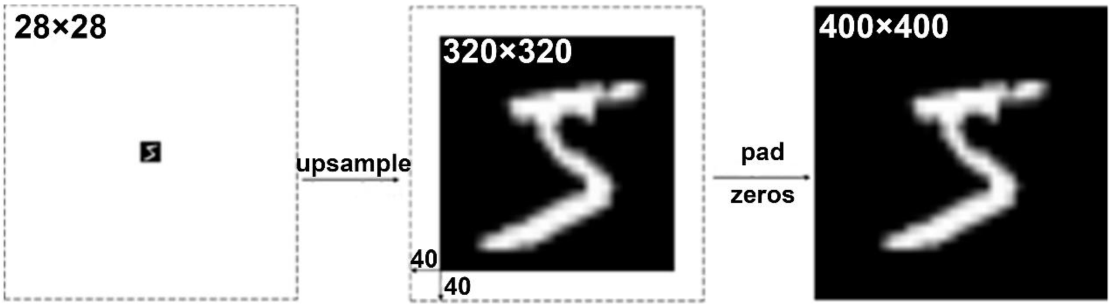

Figure S3. Image processing flowchart. Image processing flowchart: The input images are first enlarged to 320×320 pixels via upsampling, then padded to 400×400 pixels.

S7. Details of network training

Pytorch is a well-known neural network framework open sourced by Facebook (Meta), and is used to train our network model. The loss function is crossentropy and the optimizer is adam. Figure S5 shows the curves of loss function and accuracy with the number of training epochs.

S8. The choice of strategies

Equation (16) can lead to two specific encoding strategies: amplitude input phase modulation and phase input phase modulation. We compared the performance of these encoding strategies under different coherence lengths on the Kuzushiji-MNIST Japanese characters, Digit-MNIST handwritten digits, and Fashion-MNIST fashion items datasets. The input size of the MNIST dataset is 96×96, with a parameter quantity of 96×96×2, the modulation layer spacing z_1 = 2.6575 mm, and the diffraction distance to the detection plane z_2 = 6.39 mm. Figure S6 shows the test accuracy of these strategies under different coherence illumination in the case of second-order pseudo-nonlinearity, respectively for the Digit MNIST, Fashion MNIST, and Kuzushiji MNIST datasets. In conclusion, Strategy 1 is more effective than Strategy 2.

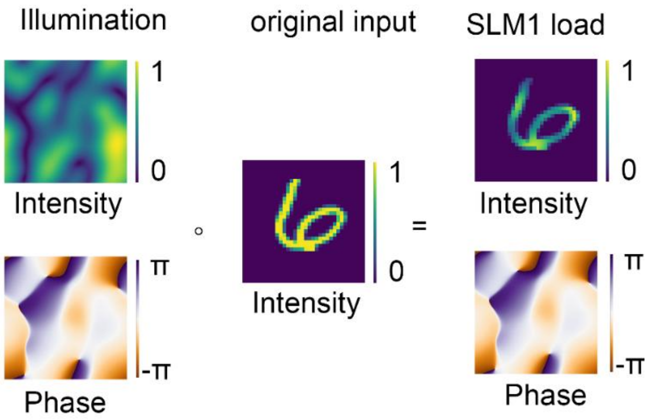

Figure S4. Load input images under coherent illumination to SLM1.

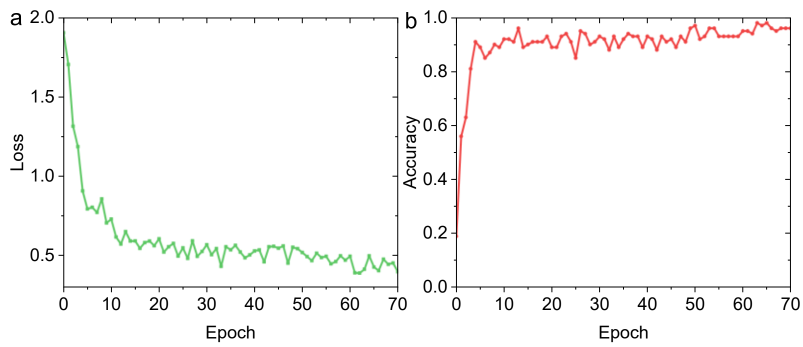

Figure S5. Loss and accuracy during network training.

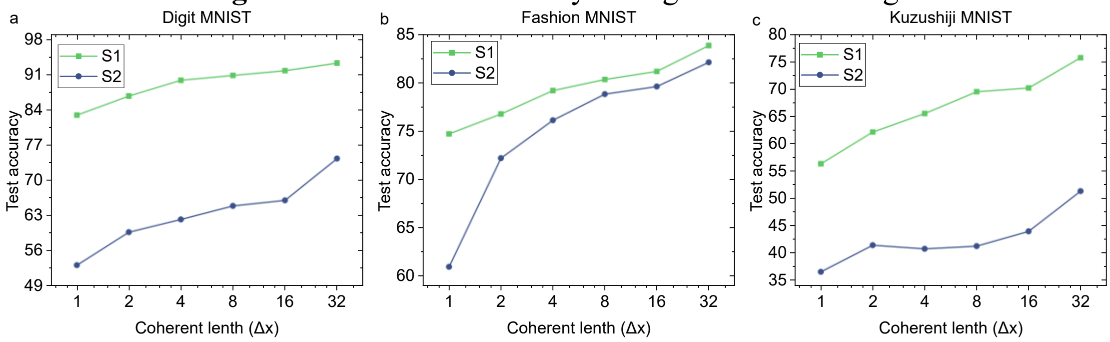

Figure S6. Test accuracy of the neural network choosing four strategies under different degrees of coherence illumination in the case of second-order pseudo nonlinearity, where Δx = 8 μm: (a) Digit MNIST results; (b) Fashion MNIST results; (c) Kuzushiji MNIST results.

S9. Details of network parameters for real world data set

Real world datasets in this paper include Real-world Affective Faces (RAF) facial expression dataset and the MedMNIST medical image. For the RAF dataset, four emotional categories including Surprise, Fear, Disgust, and Anger are adopted. The training set contains 2,993 samples (1,290, 281, 717, and 705 samples for each category, respectively), while the test set consists of 725 samples (333, 67, 169, and 156 samples correspondingly). All images are preprocessed into grayscale images with a resolution of 100 × 100 pixels. Regarding the MedMNIST dataset, four typical medical image modalities are utilized, namely Derma, Organ-CT, Pneumonia, and Retina. The training set comprises 25,770 samples (7,007, 12,975, 4,708, and 1,080 samples per category), and the test set includes 4,039 samples (1,003, 2,392, 524, and 120 samples for each modality). All samples are standardized to 128 × 128 grayscale images. All models were trained by crossentropy loss function.

Firstly, we take about the optical model. The RAF dataset is processed through three layers of 100×100 nonlinear encoding layers and then through a 200×200 linear optical fully connected layer to obtain the output. The coherence length of the input light field is $45\Delta x$. The Med MNIST dataset is processed through three layers of 128×128 nonlinear encoding layers and then through a 256×256 linear optical fully connected layer to obtain the output. The coherence length of the input light field is $50\Delta x$. The pixel size $\Delta x = 8$ μm, and the diffraction distances are $z_1 = z_2 = 3.5$ mm. Coherent model (CONN) has the then parameter, while the illumination is coherent. Each nonlinear parameter is $t = x + b$.

Secondly, we take about the optoelectronic model (PCOENN). The RAF dataset is 100×100 size. Optical part has 3 layers of 100×100 nonlinear encoder with coherence length of the input light field is $50\Delta x$ and $z_1 = z_2 = 3.5$ mm. The electric part is an ensemble network with 8 parallel fully connected network (FCNN). Each FCNN is shown in algorithm 1:

Algorithm 1: FCNN of PCOENN for RAF

```
nn.Linear(1000, 512); # first layer：1000 → 512
nn.GELU();
nn.Linear(512, 128); # second layer：512 → 128
nn.GELU();
nn.Linear(128, 4); # third layer：128 → 4
```

The Med MNIST dataset is $64 \cdot 64$. The optical part has 3 layers of $64 \cdot 64$ nonlinear encoder with coherence length of the input light field is $25 \cdot \Delta x$ and $z_1 = z_2 = 1.8$ mm, The electric part is an FCNN, which is shown in algorithm 2:

Algorithm 2: FCNN of PCOENN for Med MNIST

```
nn.Linear(4096, 512); # first layer：4096 → 512
nn.GELU();
nn.Linear(512, 256); # second layer：512 → 256
nn.GELU();
nn.Linear(256, 4); # third layer：256 → 4
```

Thirdly, we take about the electric CNN (ECNN) model. The RAF dataset is $100 \cdot 100$ size and the network is shown in algorithm 3:

Algorithm 3: ECNN for RAF

```
// Conv1: 1 input channel, 16 output channels, kernel=3, padding=1
nn.Conv2d(in_channels=1, out_channels=16, kernel_size=3, padding=1);
nn.GELU();
nn.MaxPool2d(kernel_size=2);
// Conv2: 16 input channels, 32 output channels
nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, padding=1);
nn.GELU();
nn.MaxPool2d(kernel_size=2);
// Conv3: 32 input channels, 64 output channels
nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=1);
nn.GELU();
nn.AvgPool2d(kernel_size=1);
// Fully connected part
nn.Linear(64, 32);
nn.GELU();
nn.Linear(32, 4);
```

The Med MNIST dataset is 64 · 64 and the network is shown in algorithm 4:

Algorithm 4: ECNN for Med MNIST

```
// Conv1: 3 input channels, 16 output channels
nn.Conv2d(in_channels=3, out_channels=16, kernel_size=3, padding=1);
nn.ReLU();
nn.MaxPool2d(kernel_size=2);
// Conv2: 16 input channels, 32 output channels
nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, padding=1);
nn.ReLU();
nn.MaxPool2d(kernel_size=2);
// Conv3: 32 input channels, 64 output channels
nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=1);
nn.ReLU();
nn.AvgPool2d(kernel_size=1);
// Fully connected part
nn.Linear(64, 32);
nn.ReLU();
nn.Linear(32, 4);
```

Fourthly, we take about the ensemble FCNN (EFCNN). The RAF dataset is 100×100 size and the model has 12 FCNNs which is the same as the optoelectronic model shown in algorithm 1. The Med MNIST dataset is 64×64 and the model has 4 FCNNs which is same as the optoelectronic model shown in algorithm 2.

Finally, the parameter count and computational cost could be easily calculated via the above information. Computational cost is measured in MACs (multiply-accumulate operations). 1 MAC is $a = a + (b \times c)$ and 1 MAC = 2 FLOPs (floating point operations). The result is in the table S1.

Table S1. Parameter count, computational cost and accuracy of 5 models in 2 set

	Parameter count		Computational cost		Accuracy	
Model	RAF	MED	RAF	MED	RAF	MED
CONN	0.08M	0.131M	0.08M	0.131M	66.62	92.62
HONN	0.08M	0.131M	0.08M	0.131M	72.68	97.17
HOENN	41.537 M	2.25M	41.48M	4.591M	75.86	98.39
ECNN	0.0255M	0.0258M	24.48M	11.2 M	69.38	98.04
EFCNN	62.246 M	8.936M	124.44 M	17.84M	71.72	96.96

S10. Detail results of Digit MNIST and Fashion MNIST

This part shows the detail experiment and simulation results of Digit MNIST and Fashion MNIST.

Table S2 Simulation and experiment results under different coherent length l_c of digits

l_c	Sim (1st)	Sim (3rd)	Exp (1st)	Exp (3rd)	Diff (1st)	Diff (1st)
8	79.25	83.225	70.25	67.75	9	9.475
32	89.34	92.075	83.75	87.75	5.59	4.325
64	93.865	94.17	87.25	88.25	6.615	5.92
∞	94.83	95.55	88.75	90.25	6.08	5.3

Table S3 Simulation and experiment results under different pseudo-nonlinear order of digits

Order	Sim (l_c 32)	Sim (l_c∞)	Exp (l_c 32)	Exp (l_c∞)	Diff (l_c 32)	Diff（l_c∞）
1	89.34	94.83	83.75	88.75	5.59	6.08
2	89.225	95.19	84.75	91	4.475	4.19
3	92.075	95.55	87.75	90.25	4.325	5.3
4	93.07	96.53	85.5	91.75	7.57	4.78

Table S4 Simulation and experiment results under different coherent length l_c of fashions

l_c	Sim (1st)	Sim (3rd)	Exp (1st)	Exp (3rd)	Diff (1st)	Diff (1st)
8	70.512	72.22	60.25	61.5	10.262	10.72
32	77.244	79.584	66.75	72.25	10.494	7.334
64	79.648	81.868	71.75	75.25	7.898	6.618
∞	81.07	83.89	75.25	78.75	5.82	5.14

Table S5 Simulation and experiment results under different pseudo nonlinear order of fashions

Order	Sim (l_c 32)	Sim (l_c∞)	Exp (l_c 32)	Exp (l_c∞)	Diff (l_c 32)	Diff（l_c∞）
1	77.244	81.07	66.75	75.25	10.494	5.82
2	77.444	82.15	66	76.75	11.444	5.4
3	79.584	83.89	72.25	78.75	7.334	5.14
4	81.056	85.04	76	80.25	5.056	4.79

S11. Detail results of Digit MNIST and Fashion MNIST in different test coherence length

This part shows the detail results of Digit MNIST and Fashion MNIST for coherence robustness.

Table S6 Test accuracy of Digit MNIST under 1st order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	79.2	24	15.6	17.4
8	80.6	31	17.6	19.2
16	80	39.8	29.2	30
32	78.4	82.2	52.4	50
48	73.4	93.6	78.6	67
64	73.8	95.2	92	82.6
96	71.6	96	95.3	91.6
128	74	95.4	96.4	94.2
256	72.8	95.2	96.2	95.6
∞	72.6	95.4	95	95.6

Table S7 Test accuracy of Digit MNIST under 2nd order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	80	26.4	18.6	17.6
8	81.2	31.8	22	22
16	78.2	40.2	29.2	28.6
32	72.8	80.2	55.2	49.8
48	71.6	92.6	78.6	70.6
64	70.2	95.6	89.6	86
96	68.4	95.6	92.6	92.8
128	68.2	94.4	93	94.2
256	68.4	94.4	93.4	96.2
∞	67.8	93.4	94.4	96.2

Table S8 Test accuracy of Digit MNIST under 3rd order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	80.6	45.8	32.2	34.2
8	83.6	58.8	38.2	50.6
16	81.6	74.8	49.6	59
32	82.2	90	73.6	74
48	81.8	94	87.4	86.4
64	81.2	95.2	94.6	91.2
96	80.4	95.4	94.4	94.8
128	80.2	95.8	95.6	95.4
256	81.8	95.2	96	95.5
∞	81.6	94.8	95.2	95.6

Table S9 Test accuracy of Digit MNIST under 4th order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	83.8	56.6	49.4	56.4
8	84.6	69.8	66	72
16	82.8	83.6	77.8	81.2

32	81.2	92.8	90.4	86.4
48	79	94.6	94.6	91.8
64	78.4	96.4	96.4	93.6
96	79.4	95	96.4	95.8
128	79	96	97	96.8
256	78	96	96.8	97
∞	77.8	96	96.8	97.2

Table S10 Test accuracy of Fashion MNIST under 1st order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	63.2	35.8	22.2	13.8
8	67	42.4	26	14
16	65.6	52.8	31.4	14.6
32	63.2	75.6	48.6	18.6
48	60.4	76.2	69.6	20.6
64	59.2	75.4	78.4	28.8
96	59.4	73.6	79.8	44.4
128	59.2	72	80.2	62.8
256	59.8	71.6	80	78.6
∞	59.8	71.4	78.8	79

Table S11 Test accuracy of Fashion MNIST under 2nd order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	64.6	36.6	24.8	16.2
8	69	39.6	24.8	19.2
16	66.8	52.2	28.8	22
32	63	75.2	54.4	29.4
48	60.6	73.6	74	34.2
64	59.8	73.8	77.8	42.4
96	61	73.4	77.6	65.8
128	59	73	76.6	76.2
256	59	71.6	76	80.6
∞	59.6	71.33	75.8	81.8

Table S12 Test accuracy of Fashion MNIST under 3rd order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	66.6	37.6	23	26.2
8	67.4	38.6	26.4	32.8
16	65.6	53.6	31	38.2
32	63.4	74.6	53.4	41.6
48	60.8	78.2	72	50.6
64	60	76.6	79.2	60
96	58.8	75.6	81	70.4

128	58.6	76.8	81	79.4
256	58.6	76.2	80.9	82
∞	58.8	76.6	80.8	82.2

Table S13 Test accuracy of Fashion MNIST under 4th order

$l_{c\text{-}test}$	$l_{c\text{-}train} = 8$	$l_{c\text{-}train} = 32$	$l_{c\text{-}train} = 64$	$l_{c\text{-}train} = \infty$
4	64.4	38.8	26	28
8	65	45.6	30.8	38.8
16	66.8	57.4	36.2	44.2
32	65.8	74	53.6	55.6
48	62.6	77.6	71.8	63.8
64	60.8	78.6	78.2	74.2
96	61.6	79	79.6	81.2
128	61.2	77.4	79.4	82.6
256	62.2	77.6	80.2	84
∞	63.2	78	80	84.66

Reference

1. Basu, S., Hyde, M. W., Xiao, X., Voelz, D. G. & Korotkova, O. Computational approaches for generating electromagnetic Gaussian Schell-model sources. *Opt. Express* **22**, 31691 (2014).
2. Voelz, D., Xiao, X. & Korotkova, O. Numerical modeling of Schell-model beams with arbitrary far-field patterns. *Opt. Lett.* **40**, 352 (2015).
3. Ma, P. *et al.* Numerical Approach for Studying the Evolution of the Degrees of Coherence of Partially Coherent Beams Propagation through an ABCD Optical System. *Appl. Sci.* **9**, 2084 (2019).
4. Zhou, T. *et al.* In situ optical backpropagation training of diffractive optical neural networks. *Photonics Res.* **8**, 940 (2020).