optica

All-optical neural network with nonlinear activation functions: supplementary material

YING ZUO,[+] BOHAN LI,[+] YUJUN ZHAO,[+] YUE JIANG, YOU-CHIUAN CHEN, PENG CHEN, GYU-BOONG JO, JUNWEI LIU,[*] AND SHENGWANG DU[*]

Department of Physics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

**Corresponding authors: liuj@ust.hk , dusw@ust.hk*

Published 29 August 2019

This document provides supplementary information to "All-optical neural network with nonlinear activation functions," https://doi.org/10.1364/OPTICA.6.001132.

S1. Spatial light modulator (SLM)

Spatial light modulator (SLM) is a device that is used to spatially modulate amplitude, phase or polarization of light [S1]. Micro display pixels composed of liquid crystal molecules form the screen of SLM. By setting voltage distribution on screen, the orientation of liquid crystal of each pixel can be rotated with different angle. Resulting from birefringence, reflective or transparent light is spatially modulated. We constructed the linear operation and 2-layer all optical neural network (AONN) using two SLM (HOLOEYE PLUTO-2 and LETO) for the linear operations as shown in Fig. 2(a) and Fig. 4(a). Both SLMs are phase-only reflective liquid crystal on silicon modulators with full high-definition resolution (1920×1080) and 8-bit gray level patterns are addressed to regulate the voltage of each pixel.

S2. Gerchberg-Saxton algorithm and feedback iteration process for configuring the linear matrix elements

We take laser beam spots to represent a vector in the linear operation. A SLM is used to split the laser beams into different directions and the splitting beams in the same direction are summed after a lens. For a SLM placed on xy plane with Gaussian beam illumination, the complex amplitude of reflective light is $E_0(x,y)e^{i\phi(x,y)}$, where $E_0(x,y)$ is the amplitude of the incident light and ϕ is the phase change induced by the SLM. The phase modulation ϕ of 1920×1080 pixels can be controlled independently. In experiment, the SLM is placed at the back focal plane of lens to generate the separated beams. The output plane is at the front focal plane of the lens. In this configuration, the lens performs the Fourier transform and the output can be expressed as

$$E(x,y) = \mathcal{F}\{E_0(x,y)e^{i\phi(x,y)}\}. \quad \text{(S1)}$$

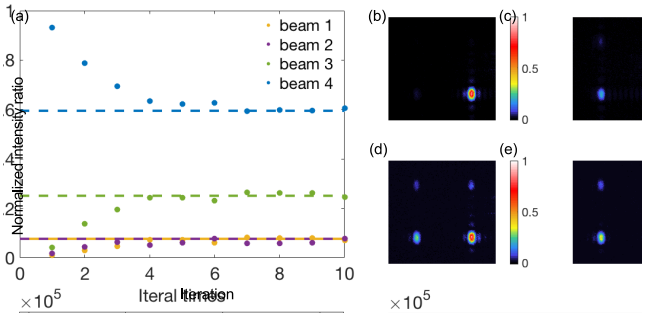

Fig. S1. Iteration process for a single neuron for configuring weight matrix element. (a) Iteral weight generation process for single input node based on gs algorithm to realize linear operation. (b)(c)(d)(e) Normalized beam intensity profile after 0, 2, 5 and 9 times iteration.

Our task is to find the proper $\phi(x,y)$ which satisfies the target profile $|E(x,y)|^2$ with predesigned weight W_{ij}. For the linear operation, we need divide the input beam into different spots in its Fourier plane. The initial phase profile setting $\phi(x,y)$ can take a superposition of different phase gratings following the grating equation. To finely tuning the phase profile to achieve the target outputs, we use the Gerchberg-Saxton (GS) algorithm [S2, S3, S4] to minimize the error. At the first step, the targeted output vector $\vec{a} = (a_1, a_2, \dots, a_n)$ is fed to GS algorithm to generate an initial encoded pattern for SLM. Then an iterating vector $\vec{c}_i = (c_{i1}, c_{i2}, \dots, c_{in})$ with $c_{i,j} = \left[\delta \frac{a_{i,j}}{b_{i-1,j}} + (1-\delta)\right] \times c_{i-1,j}$ is continued to feed the GS algorithm, where δ is a feedback parameter ranging from 0 to 1 and b_{i-1} is the measured output vector from the previous step. The iteration process stops when the error $\epsilon_j = |\frac{b_{i,j}-a_j}{\frac{1}{n}\sum a_j}|, j \in [0,n]$ is enough small. The output vector is encoded by the powers of four laser

beams for $n = 4$. We choose $\delta = 0.2$ and $\epsilon_j \leq 0.05$ in the iteration.

The powers of four laser beams converge with the iteration as shown in Fig. S1(a). The targeted output vector is $\vec{a} = (0.31, 0.31, 1.00, 2.73)$, which are marked with dash lines in Fig. S1(a). It also shows the images of four laser beams with no iteration and with 2, 5 and 9 times of iteration in Fig. S1(b), (c), (d) and (e) respectively.

S3. Principle of linear power summation

In our AONN, a linear vector is represented by an array of powers of light spots. For a linear matrix operation, after the SLM, each input beam is divided into multiple beams along directions. The Fourier lens summates the beams along the same direction into one spot on its front focal plane. The size of the output spot is determined by the input beam size at the SLM and the focal length of the lens. The interference from multiple beams modifies the spatial profile of the output spot. The power of the output spot is the integral over the spot area, which, according to the energy conservation, is a linear summation of the total powers of the interfering beams. In the experiment, we analyzed the power distribution on the focal plane by cameras and chose the square detection area which is two times larger than the spot area.

S4. The two matrixes used for testing the linear operation

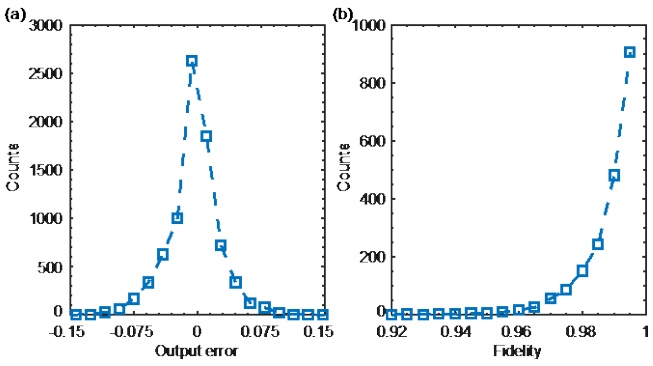

Fig. S2. Linear operation characterization of the random matrix. (a)Error distribution of the output vectors. (b)Fidelity distribution of the output vectors.

To test the error of linear operation, we measure two matrixes. The first matrix is a Hankel matrix with

$$A = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \\ 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1.0 \\ 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1.0 & 1.0 \end{pmatrix} / 3.7. \quad \text{(S2)}$$

The second matrix is a random matrix with

$$W_{random} = \begin{pmatrix} 0.04 & 0.34 & 0.30 & 0.02 & 0.30 & 0.26 & 0.24 & 0.29 \\ 0.40 & 0.01 & 0.01 & 0.22 & 0.09 & 0.26 & 0.05 & 0.38 \\ 0.41 & 0.26 & 0.26 & 0.34 & 0.05 & 0.06 & 0.21 & 0.02 \\ 0.08 & 0.33 & 0.19 & 0.35 & 0.29 & 0.26 & 0.26 & 0.24 \end{pmatrix} \quad \text{(S3)}$$

The error distribution and fidelity distribution are shown in Fig. S2. The input weight is the same as the test of Hankel matrix.

S5. Two-dimensional (2D) magneto-optical trap (MOT) of ^{85}Rb atoms and EIT

MOT is an apparatus that cools and traps neural atoms using lasers and magnetic fields [S5, S6]. The 2D MOT configuration is described in detailed in our previous work [S7]. The experiment is running periodically with a repetition rate of 10 Hz and a duty cycle of 12%. Fig. S3 shows the measurement timing. The experimental cycle starts from the MOT loading, where trapping and repumping lasers are on. The trapping laser is used to cool the atoms and the repumping laser is used to maintain atoms in the cooling cycle. The decouple laser is turned on for 50 μs after cooling cycle to depopulate atoms at the other ground state (F=2). During the 10 ms detection window, the probe and coupling lasers are turned on for 880 us, where the camera takes the transmitted probe light power from the EIT process with an exposure time of 2 ms. It is triggered by a pulse generated by a delay generator and starts to exposure after an electronic delay around 87.7 μs.

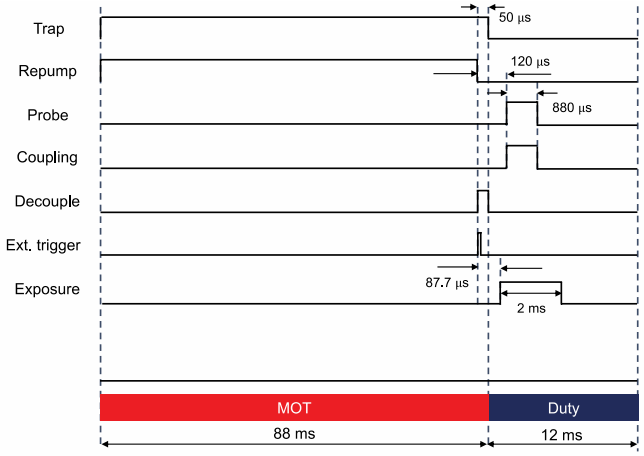

Fig. S3. Experiment timing circle of trap, repump, probe and coupling beams and camera exposure time.

For the EIT measurement and the nonlinear optical activation fucntions plotted in Fig. 3(d) and 3(e) in the main manuscript, the coupling beam input power is linearly scaled to 1 for 200 $\boldsymbol{\mu}$W. The probe beam laser power is the summation of 16-bit grey level of beam area reading from the camera (Hamamatsu C11440), which is proportional to the power of the probe beam and $\mathbf{1 \times 10^5}$ is scaled to 1.

S6. Two-layer AONN: Using a single SLM to achieve both the input vector and the first linear operation

As shown in Fig.2, for a general purpose, we use one SLM to obtain the input vector input vector $\vec{\mathrm{X}}$ and a second SLM (with the Fourier lens) for the linear operation W. We can rewrite matrix-vector multiplication operation into following equation:

$$\mathrm{W}\vec{\mathrm{X}} = \left(\mathrm{W}\ \mathrm{diag}(\vec{\mathrm{X}})\right)\vec{I} \quad \text{(S4)}$$

The elements of input vector are absorbed into the matrix, transforming $\mathrm{W} \rightarrow \mathrm{W}' = \mathrm{W}\ \mathrm{diag}(\vec{\mathrm{X}})$. Now we only need one SLM for generating matrix W′ for different input $\vec{\mathrm{X}}$. $\vec{I}$ is a fixed all-one vector that is generated by coupling laser beam in experiment. And for special input vector $\vec{\mathrm{X}}$ consisting binary element (0 or 1), we only need to convert any column of matrix W to all 0 to get W′ that is simple on experiment. Therefore, for the input and the first linear operation, we need only one SLM.

S7. Training of the two-layer AONN

On computer, we trained the two-layer neural network which consists of one input layer, one hidden layer and one output layer and $\vec{X}_1 = (x_{1,1}, \ldots, x_{1,16})^T$, $\vec{X}_2 = (x_{2,1}, \ldots, x_{2,4})^T$ and $X = (v_{3,1}, v_{3,2})^T$ are used to present the values of each layer namely. The relations of this vectors follow the equations below:

$$\vec{X}_2 = \phi(W_1\vec{X}_1) \quad \text{(S5)}$$

$$\vec{X}_3 = W_2\vec{X}_2 \quad \text{(S6)}$$

ϕ is the nonlinear activation function. $W_1 = \left[w_{ij}^1\right]_{4\times16}$ and $W_2 = \left[w_{ij}^2\right]_{2\times4}$ are the matrix to be determined by training.
Considering the experimental implementation of the neural network, the beam power of reflected beams and the power of incident light should be the same for each input node. The matrix must satisfy the condition that

$$\sum_{j=1}^{n} w_{ij} = c_i \tag{S7}$$

for any $1 \le i \le m$. Ideally, regardless of diffraction, $c_i = 1$. Also, due to the limited number of the pixels on the liquid crystal spatial light modulator (LC-SLM), we found that, in experiment, it was hard to adjust the weight accuracy if the power ratio of reflection beams from a single input node too large or too small. So, when training on computer, we restricted the ratio between the weights from the same neuron to ensure the practicability in experiments. Hence, during the training, the weight $W = \left[w_{ij}\right]_{m\times n}$ is required to fulfill

$$\frac{1}{M} \le \frac{w_{ij}}{w_{ij'}} \le M \tag{S8}$$

The conventional method to update weights in back propagation fails to meet conditions as Eqns. (S7) and (S8), and in the following, we derive the appropriate way to carry out weight updating. The other training conditions, including the optimizer, the loss function, and the nonlinear activation function (EIT in our case), will also be specified.

S7.1 Gradient descent and the optimizer

Consider the loss function $L(F = XW)$ where $X = \left[x_{ij}\right]_{l\times m}$ is the n-th layer and $W = \left[w_{ij}\right]_{m\times n}$ is the weight. From Eqn. (1), since for any $1 \le i \le m$, $w_{in} = c_i - \sum_{j=1}^{j=n-1} w_{ij}$, for $j \ne n$, we have

$$\frac{\partial L}{\partial w_{ij}} = \sum_{i'j'} \left(\frac{\partial L}{\partial F_{i'j'}}\right)\left(\frac{\partial F_{i'j'}}{\partial w_{ij}}\right) = \sum_{i'=1}^{m} \frac{\partial L}{\partial F_{i'j}} x_{i'i} - \sum_{i'=1}^{m} \frac{\partial L}{\partial F_{i'j}} x_{i'N} \tag{3}$$

and for $j = n$,

$$\frac{\partial L}{\partial w_{in}} = -\sum_{j=1}^{n-1} \frac{\partial L}{\partial w_{ij}} \tag{4}$$

In training, we choose the momentum update as out optimizer since many others like Adam are incompatible with the gradient descent we here come up with. Also, note that different choices of n may result in different weights in the end if the local minima are found on the boundary of the domain. However, from our experience, normally different choices of n do not affect the training result since there are many sets of weights useful enough to classify the Ising model.
At last, we choose the cross-entropy loss as our loss function for two reasons. First, it is the most commonly-used training loss for classification. Second, the cross-entropy loss draws the relationship between the predicted probabilities and the values of the output layers based on the Boltzmann distribution, which is in accordance with the probabilities and the energies in statistical physics.

S7.2 Activation functions

For our three-layered neural network (one input layer, one hidden layer, and one output layer), there are 4 neurons in the hidden layer with different activation functions respectively. We train our neural network with the activation functions measured in the experiments in the first place. The source code can be given upon requested.

S7.3 *Training result of Ising model*

The 2D Ising model is described as

$$H(\sigma) = -J \sum_{<i,j>} \sigma_i \sigma_j \ , \tag{S9}$$

where $\langle i\ j\rangle$ is summation over pairs of nearest-neighbor sites, σ_i is spin on at site i, $\sigma_i \in \{+1, -1\}$, J is the interaction strength between nearest-neighbor spin pairs.
We use Monte Carlo (MC) method to generate raw train configurations of Ising model at 1J temperature and 100J temperature, labeled as order and disorder respectively. The spin variables of configurations need to be transformed from $\{+1, -1\}$ to $\{+1,0\}$ as input of AONN.
For 4×4 Ising model, we construct a two-layer AONN without bias and with 4 hidden neurons whose activation function are EIT nonlinear functions that are obtained by fitting the experiment data showed in Fig. 2(e). We have two matrixes W1 and W2 corresponding to first and second linear operations for two-layer AONNs. All parameters of two matrixes must be positive. We let the sum of each column $\sum_j w_j$ be fixed number. For element w_i on each column, we let it to satisfy $\frac{1}{M} \le \frac{w_i}{\sum_j w_j}$, M is set to 13 for W1 and 11 for W2.

Table S1. W1 for 4 by 4 Ising model
W1 is a 4 by 16 positive matrix used in first linear operation for AONN

W1 colum1~column8							
0.0069	0.0058	0.0087	0.0044	0.0040	0.0136	0.0125	0.0047
0.0226	0.0211	0.0235	0.0148	0.0234	0.0264	0.0332	0.0225
0.0117	0.0127	0.0113	0.0241	0.0210	0.0038	0.0118	0.0221
0.0034	0.0033	0.0096	0.0136	0.0040	0.0051	0.0066	0.0114
W1 colum9~column16							
0.0055	0.0176	0.0177	0.0055	0.0112	0.0075	0.0055	0.0193
0.0177	0.0363	0.0378	0.0259	0.0352	0.0299	0.0249	0.0401
0.0301	0.0051	0.0077	0.0232	0.0177	0.0199	0.0202	0.0060
0.0180	0.0074	0.0128	0.0163	0.0159	0.0134	0.0171	0.0121

Table S2. W2 for 4 by 4 Ising model
W2 is a 2 by 4 positive matrix used in second linear operation for AONN

0.2272	0.0228	0.2272	0.1076
0.0228	0.2272	0.0228	0.1424

On the conventional computer, we implement two-layer AONNs to perform supervised learning by feeding labeled train data. After training, we implement two-layer AONNs on experiment by using trained matrix parameters. The trained parameters of matrixes W1 and W2 for 4×4 Ising model list in Table S1and S2.

S8. Ising model related data processing and results

Considering system errors in experiment that causing translation and scale of final output, we normalize and correct our experiment output data.

S8.1 Normalization

First, we use equation below to normalize all experiment outputs:

$$[A_{norm}, B_{norm}] = \frac{[A, B] - [A_{all\ down}, B_{all\ down}]}{[A_{all\ up}, B_{all\ up}] - [A_{all\ down}, B_{all\ down}]} \quad (S10)$$

$[A, B]$ is experimental two-dimensional output for any input configurations, $[A_{all\ up}, B_{all\ up}]$ is experimental output of configuration with all spin up, and $[A_{all\ down}, B_{all\ down}]$ is experimental output of configuration with all spin down.

S8.2 Correction

After normalization, we use a 45-degree line: $A_{norm} = B_{norm} + \Delta$ as criterion line of ordered $(A_{norm} > B_{norm} + \Delta)$ or disordered $(A_{norm} < B_{norm} + \Delta)$. And Δ is the correction parameter.

Δ is the value that minimizes the root mean square error (RMSE) between phase probability plots from theory and experiment. We can learn it using result of few configurations and use it to experimental result of large configurations.

S8.3 Results for 4×4 Ising model

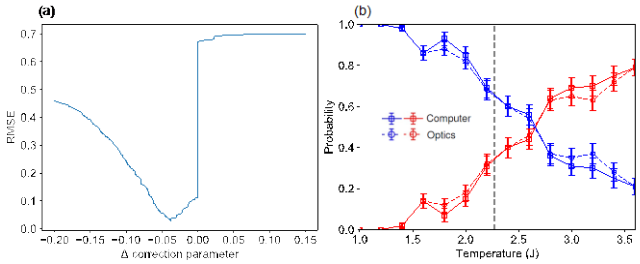

Fig. S4. For 4×4 Ising Model, we correct normalized experiment outputs using a correction parameter Δ minimize the difference of phase probability curves from experiment and theory. (a) RMSE between phase probability curves from experiment and theory. (b) Phase probability vs temperature for 100 MC configurations where correction parameter $\Delta = -0.0375$ in (A) is learned.

For 4×4 Ising Model, we generate 4000 MC configurations every 0.2J from 1.0J to 3.6J. We learn the correction parameter from result of 100 MC configurations per temperature as showed in Fig. S4. When $\Delta = -0.0375$, the RMSE between experiment and theory plot (100 MC configurations per temperature) reaches minimum. Using $A_{norm} = B_{norm} - 0.0375$ as criterion line for all normalized experiment outputs, we plot 40 graphs of phase probability vs temperature for 100 MC configurations and 1 graph of phase probability vs temperature for 4000 MC configurations in Fig. S5 and Fig. S6 respectively. In all graphs, the experiment plots have few differences with theory plots. It clearly shows that the correction parameter learned from few MC configurations can be applied to correct experiment output data of other MC configurations. From machine learning perspective, it is also a learning process. The few MC configurations are train data, other large MC configurations are test data.

S8.4 Necessity of the activation function for learning phase of Ising model

We have also tried to use linear classifier to distinguish order or disorder phases of Ising Model. We trained linear classifier using train data also generated by MC simulation at 1J and 100J temperature. After training, we use linear classifier to classify statistic configurations at other temperature and plot the phase probability vs temperature.

From Fig. S7, we can see the phase probability curves predicted by linear classifier fluctuate around 50% along temperature that do not match the actual phase transition property of Ising model. It is clear that linear classifier cannot properly learn phase of Ising model even for small 4×4 size. It proves that two-layer AONNs implemented on experiment must take advantage of the nonlinear function to learn phase of Ising model.

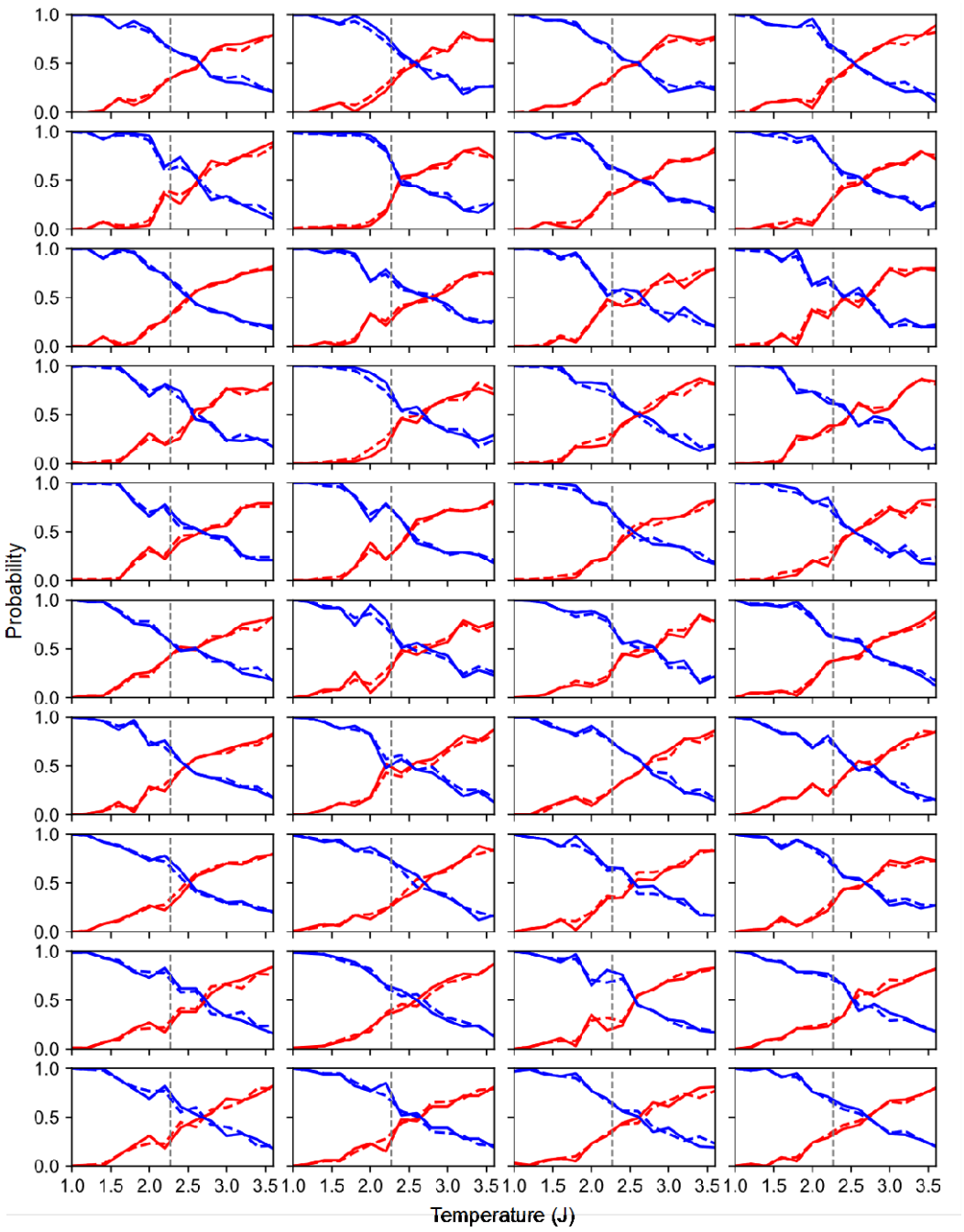

Fig. S5. For 4×4 Ising Model, we generate 4000 MC configurations every 0.2J from 1.0J to 3.6J. We divide 4000 MC configurations into 40 parts in sampling order. And we draw 40 graphs of phase probability vs temperature for 100 MC configurations. Graphs are arranged from left to right and from top to down in sampling order. Solid line is computer results and dashed line is experiment results.

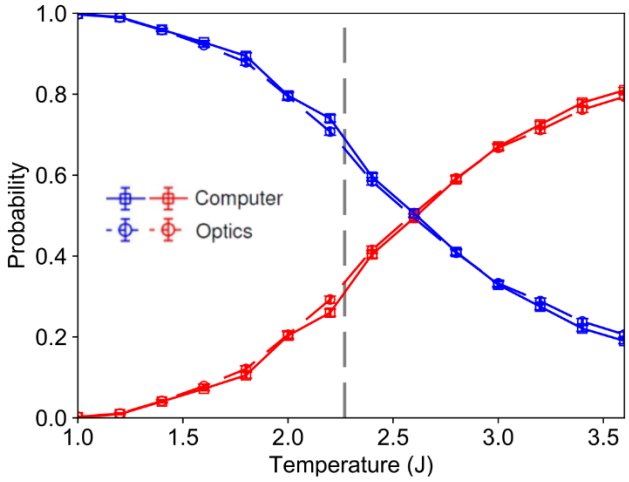

Fig. S6. For 4×4 Ising Model, we generate 4000 MC configurations every 0.2J from 1.0J to 3.6J. We draw graph of phase probability vs temperature for 4000 MC configurations.

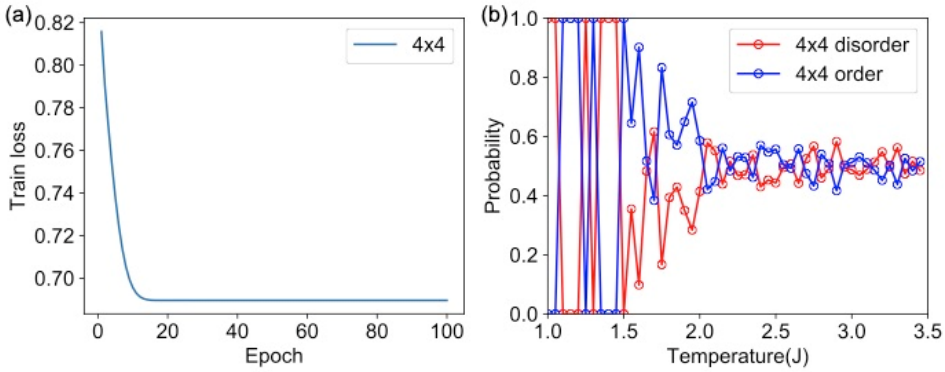

Fig. S7 For 4×4 Ising Model, we use a linear classifier to learn phase of Ising model. (A) Train loss vs epoch of linear classifier. (B) Phase probability vs temperature predicted by linear classifier.

References

S1. K. Lu and B. E. A. Saleh, "Theory and design of the liquid crystal TV as an optical spatial phase modulator," Opt. Eng. **29**, 240 (1990).

S2. G.-Z. Yang, B.-Z. Dong, B.-Y. Gu, J.-Y. Zhuang, and O. K. Ersoy, "Gerchberg-Saxton and Yang-Gu algorithms for phase retrieval in a nonunitary transform system: a comparison," Appl. Opt. **33**, 209-218 (1994).

S3. L. R. Di, F. Ianni, and G. Ruocco. "Computer generation of optimal holograms for optical trap arrays," Opt. Express **15**, 1913-1922 (2007).

S4. F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys, "Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries," Phys. Rev. X **4**, 021034 (2014).

S5. E. L. Raab, M. Prentiss, A. Cable, S. Chu, and D. E. Pritchard, "Trapping of neutral sodium atoms with radiation pressure," Phys. Rev. Lett. **59**, 2631 (1987).

S6. H. J. Metcalf and P. van der Straten. *Laser Cooling and Trapping*. Springer-Verlag New York (1999).

S7. S. Zhang, J. F. Chen, C. Liu, S. Zhou, M. M. T. Loy, G. K. L. Wong, and S. Du, "A dark-line two-dimensional magneto-optical trap of 85Rb atoms with high optical depth," Rev. Sci. Instrum. **83**, 073102 (2012).