arXiv:1808.01731v2 [cond-mat.dis-nn] 9 Aug 2018

Machine Learning Phase Transition: An Iterative Methodology

X. L. Zhao and L. B. Fu*
Graduate School of China Academy of Engineering Physics, China
(Dated: November 26, 2019)

Machine learning provides new techniques to investigate phase transitions in physics. Here, we propose an iterative methodology to find the critical temperature for the two-dimensional Ising model based on machine-learning techniques. Making use of dimension-reduction algorithm, we obtain the incipient phase boundaries and labels for some samples which would be used by a convolutional neural network to find the critical temperature in an iterative manner. During the finding process, the newly labelled samples would be put in the training set and the phase boundaries would be updated toward each other. Meanwhile, the average of the boundaries would converge to the theoretical value. We also compare the relation between the capability of recognition of the convolutional neural network and magnetic susceptibility near the critical point. This work not only offers a methodology to explore unexplored phase transitions for statistical models but also put forward the motivation to study the deep connections between statistical physical models and neural networks.

I. INTRODUCTION

Classifying phases for condensed matter models benefits the understanding about physical phenomena. However this usually be formidable due to the exponentially large Hilbert space. With the advancing of current computational power and the amelioration of algorithms, machine learning has provide novel avenues to accomplish such tasks of classification[1–11].

In machine learning, the 'learners' usually deal with samples represented by points in the feature space. Finding the hypersurfaces which divide the representative points into different groups may benefit the analysis, merely analytical efficiency may be reduced due to the redundant information. Fortunately, dealing with the characteristic information would be conducive to improve the efficiency. The linear dimension-reduction technique-principal component analysis (PCA), is frequently adopted in machine-learning tasks to reduce redundant information and compress data [12–14]. Inspired by the application of this technique in clustering tasks, it has been applied to identifying the phase transitions and the order parameters from the configurations of the classical Ising model [7–11].

Different to PCA, there are neural networks which need to be trained by samples with correct labels before they are adopted to recognize new samples. During the training, the parameters of a network are optimized with respect to a cost function to learn the key characteristic patterns of training samples. This tool has been demonstrated on investigating Ising model [1, 2], studying strongly correlated fermions [3, 15], and exploring the fermion-sign problem [16]. Topological states can also be learned by artificial neural networks. For example, by introducing quantum loop topography, a fully connected neural network can be trained to distinguish the Chern insulator and the fractional Chern insulator from trivial insulators [4]. Similarly, the phase boundary between the topological and trivial phases can be identified by feed-forward neural network [5]. Neural network can also be trained to learn the discrete version of the winding number formula from the set of Hamiltonians for topological band insulators [6]. There is study focuses on the mapping between variational renormalization group and restricted Boltzmann machines [17]. Quantum-inspired tensor networks has been applied to multi-class supervised learning tasks and demonstrated by using matrix product states to parameterize models [18]. An artificial neural network trained on entanglement spectra of individual states of a many-body quantum system can be used to determine the transition between a many-body localized and a thermalized region [19]. The highly accurate classification power of fully connected and convolutional neural networks are examined for the Ising model along with the application of convolutional neural network for the classification task of XY model [20]. Unsupervised and supervised machine learning tools have been sequentially applied to find and characterize the phase diagram for a liquid-gas transition [21]. These works indicate the validity of neural networks in exploring statistical models.

In this work, we propose an iterative scheme to find the critical temperature for statistical models accurately with high confidence. In this proposal, clustering algorithms provide the phase scenarios versus temperature just by checking the raw configurations. Taking phases as labels, the incipient labels and phase boundaries can be identified. Then taking the spin configurations above as incipient training set, we employ a convolution neural network (CNN) [22, 23] to recognize the samples with temperatures close to the labelled region. Once the samples were recognized by the CNN with high accuracy, they would be added into the training set to be used to train the CNN in the next recognition iteration. Meanwhile the phase boundaries would be updated by the temperatures of the newly labelled spin configurations; If the samples can not

* E-mail: lbfu@gscaep.ac.cn

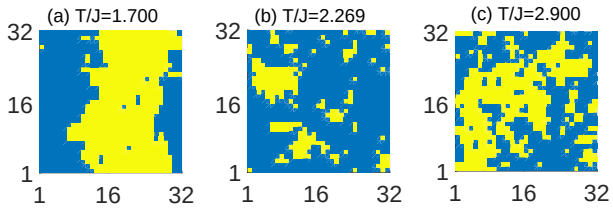

FIG. 1. Three examples of input spin configurations for the two-dimensional Ising model of size $N = 32 \times 32$ at different temperatures. Blue and yellow pixels denote up (↑) and down (↓) spins on the lattices.

be identified by the CNN, the samples with temperatures closer to the labelled region will be checked. Iterating the procedure of recognizing and updating, the region where the critical temperature locates in would shrink and the mean value of the phase boundaries would converge to the analytical value $T_c \approx 2.269$ (in units of coupling energy between nearest-neighbor spins) [24]. This analytical value offers the benchmark to check validation of our proposal. Monte Carlo technique is applied to providing the spin configurations at various temperatures, which obey the Boltzmann distribution at thermal equilibrium [25–27].

Starting from samples with correct labels, the performance of this proposal mainly depends on the capability of recognition of the CNN. This capability can be indicated by the number of training iterations cost until the CNN be capable to identify the spin configurations with high accuracy. With the CNN adopted in this work, considering magnetic susceptibility is directly related to the spin configurations, we compare the capability of recognition with the power law of magnetic susceptibility in the vicinity of the critical temperature at last.

Following in Sec. II, we introduce the classical two-dimensional Ising model. In Sec. III, we use principal component analysis to separate the model into two phases and specify the region containing the critical temperature. In Sec. IV, we show the procedure that starts from the results of principal component analysis to train a convolution neural network to find the critical point in an iterative manner. In Sec. V, we compare the recognition capability of the CNN with the magnetic susceptibility near the critical temperature. Finally we conclude in Sec. VI.

II. TWO-DIMENSIONAL ISING MODEL

In this work, we consider the widely studied prototypical two-dimensional Ising model on square lattice [24, 28] with L sites on each dimension and the total number of spins $N = L \times L$. The Hamiltonian for this model with vanishing external magnetic field reads:

$$H = -J \sum_{\langle ij,kl \rangle} s_{ij} s_{kl}. \tag{1}$$

The coupling energy J between nearest-neighbor spins is set as the energy unit in this work. The dipole spins $s_{ij(kl)}$ take $+1 = \uparrow$ or $-1 = \downarrow$ on the lattice sites, and $\langle ij, kl \rangle$ denote the pairs of the nearest-neighbor sites in two orthogonal dimensions with periodic boundary conditions. This model undergoes a thermal phase transition at temperature $T_c = 2J/\ln(1+\sqrt{2}) \approx 2.269$ with Boltzmann constant $k_B = 1$ in this work [24]. This corresponds to an abrupt change of some statistical quantities or their derivations, like the magnetic susceptibility and specific heat at the critical temperature. Three input spin configurations generated by Monte Carlo method at different temperatures are displayed in Fig. 1. In this work, only such spin configurations are used to find the critical temperature. This model offers the benchmark to check the performance of the proposal to find the critical point.

III. DETERMINING THE INCIPIENT TRAINING SET

Firstly, the linear dimension-reduction technique-principal component analysis (PCA), is adopted to assign labels (low- or high-temperature-phase) to some spin configurations. Meanwhile the incipient phase boundaries which confine the critical temperature are identified. Secondly, starting from these results, the supervised machine learning tool-convolutional neural network (CNN), would be adopted to find the critical temperature in an iterative manner. During the iterating process, if samples were assigned labels by the CNN, they would be added into the training set, meanwhile the phase boundaries would be updated by the temperatures of the newly labelled samples; If not, samples with temperatures closer to the labelled region would be recognized. The details are given in the following.

A. Principal component analysis

Spin configurations can be represented by points in a high dimensional feature space with the spin values of the configuration sites as the coordinates. Thus it is desirable to find a simpler or more accessible representation which maintains as much information of the set of large spin configurations as possible. The linear dimension-reduction technique-principal component analysis [12–14], will be employed to extract salient fluctuation characteristic patterns for the samples and to compress them into a reduced space. The phase transition can be reflected by the distribution behavior of the representative points. This provides labels for the incipient training samples used by a CNN to find the critical temperature.

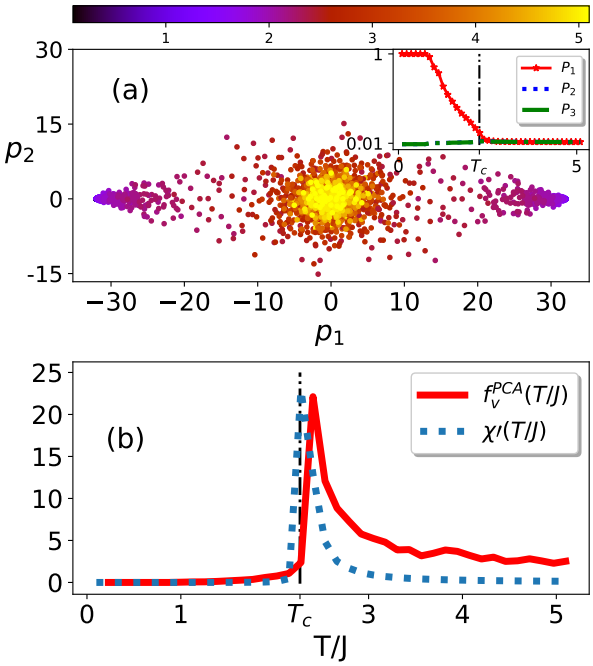

FIG. 2. (a) Principal component analysis for the spin configurations and their spin-flipped counterparts by projecting the samples onto the plane spanned by the leading two principal components. The temperature of the samples are mapped to the color of the representative dots. The inset in (a) displays the three leading normalized singular values ($P_1 > P_2 > P_3$) as a function of 20 temperature points distributed evenly in the region $T/J \in (0.1, 5.1)$. $T_c \approx 2.269$ is the analytical critical temperature. The points at each temperature is obtained by a vertical stack of 100 samples. (b) $f_v^{PCA}(T/J)$ corresponds to (a) and adjusted magnetic susceptibility $\chi\prime(T/J) = 260\chi(T/J)$ with $\chi(T/J) = \frac{J}{T}(\langle M^2\rangle - \langle M\rangle^2)$ where $M = \sum_{ij} s_{ij}/N$. The factor 260 is nothing but to make $\chi(T/J)$ be comparable to $f_v^{PCA}(T/J)$. Identical set of samples are shared in this figure.

B. Results of principal component analysis

When PCA is applied to the stacks of Ising configurations at various temperatures, the most significant variations of the distribution of the samples versus temperature can be reflected by the representative points embedded in the reduced space. The distribution would indicate the phase transition of this model to some extent.

The principal components in PCA can be obtained by singular value decomposition (SVD). While SVD reads $V = UZW^T$, the singular values are the diagonal elements of Z which are usually arranged in descending order and normalized by dividing their summation [13, 14]. V is a vertical stack of n flattened spin configurations in the form of $\mathbf{S}_m = (s_{11}, ..., s_{ij}, ..., s_{LL})_m$ with means of columns subtracted at a temperature region. The principal components are leading columns of $VW = UZ$. Equivalently, principal components can also be obtained by analyzing the eigen-decomposition of the covariance matrix $V^TV/(n-1)$ [13, 14]. The phase transition of this model can be reflected by the clustering behavior of the representative points in the dimension-reduced space as in Fig. 2 (a).

As shown by the inset in Fig. 2 (a), while there is no more than one obvious dominant component (corresponding to the maximal singular value), two leading principal components are enough to reflect the behavior of distribution. Vanishing of the dominant principal component means there is no direction on which the samples distribute with obvious fluctuations than the others. The dominant component vanishes at the temperature slightly larger than T_c which may result from the information loss during this dimension-reduction treatment. We will see that this is coincident with the bias result of PCA to reflect the critical temperature bellow.

Being projected to the space spanned by the two principal components as in Fig. 2(a), a spin configuration and its spin-flipped counterpart (spins flip: $\uparrow\rightleftharpoons\downarrow$) distribute centrosymmetriccally about the origin when analyzed together by PCA. It can be assumed that the samples with temperatures $T/J < 1$ and those $T/J > 4.5$ can be divided into two different phases with high confidence. Those with temperature $T/J \in (1, 4.5)$ distribute with large fluctuations which coincides with that the characteristic patterns of the spin configurations are too similar to distinguish them into which phase easily. This inspires us to conjecture that the phase transition can also be reflected by calculating the variance of the distribution of the representative dots in Fig. 2 (a). Thus we make use of the quantity:

$$f_v^{PCA}(T/J) = \sqrt{V_1^{PCA}(T/J) + V_2^{PCA}(T/J)} \qquad (2)$$

to confirm the division of phases. $V_{1(2)}^{PCA}(T/J)$ denotes the variance for the first (second) coordinate of the representative dots at T/J. Corresponding to distribution of the dots, $f_v^{PCA}(T/J)$ is shown as a function of T/J in Fig. 2 (b). The peak of $f_v^{PCA}(T/J)$ divides the samples into two phases. Since magnetic susceptibility $\chi(T/J)$ relates directly to the spin configurations [29, 30], we conjecture that the behavior of $f_v^{PCA}(T/J)$ versus temperature may be correlated to $\chi(T/J)$. However the peak of $f_v^{PCA}(T/J)$ deviates from T_c, while the peak of $\chi'(T/J) = 260\chi(T/J)$ locates at this critical temperature. This may result from that some information is lost in PCA which coincides with the confluent behavior of the singular-value curves in the inset in Fig. 2 (a).

According to the results of PCA above, the critical temperature locates in the region $T/J \in (1, 4.5)$ but not identified exactly. $T/J = 1$ and $T/J = 4.5$ act as the incipient boundaries for the two phase, respectively. Thus the samples with temperature $T/J \in (0, 1)$ are assigned low-temperature-phase label, and those with $T/J > 4.5$ are assigned high-temperature-phase label. Taking these samples as incipient training set for a convolutional neural network, we show the iterative procedure to find the critical temperature bellow.

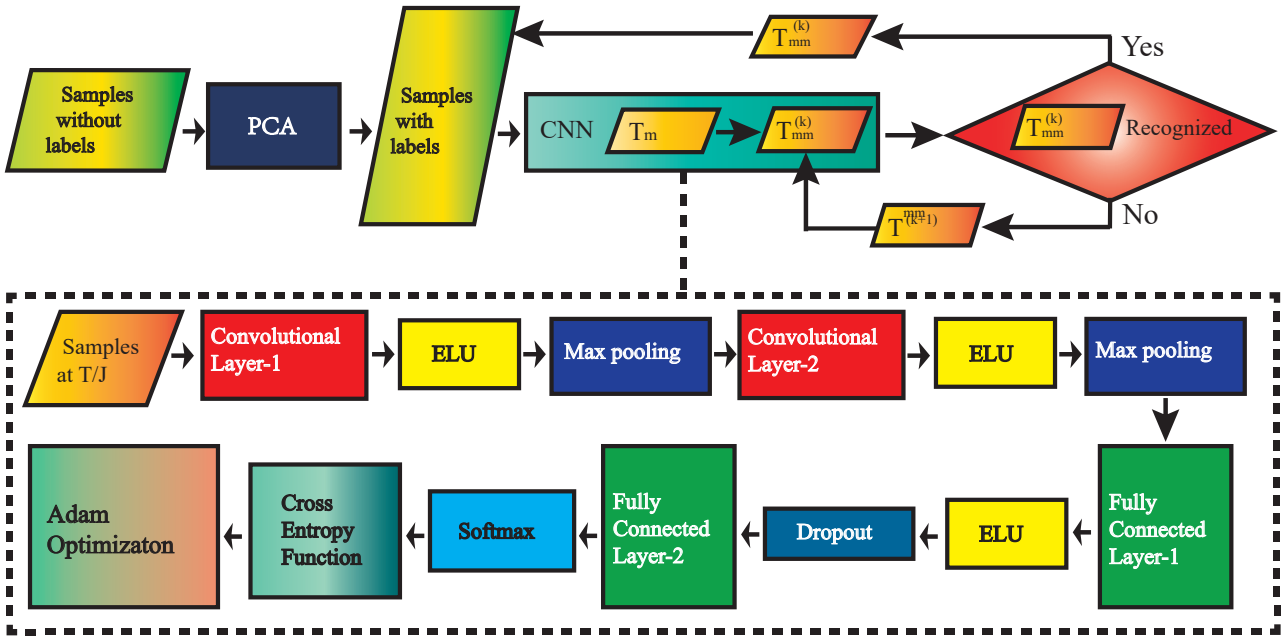

FIG. 3. The the schematics of the workflow to find the critical temperature and the convolutional neural network adopted in this work. PCA is employed to obtain the incipient phase boundaries and the labels for the training set used by the CNN; In the first convolutional layer of the CNN, the size of the convolutional kernels are set according to that of the training samples. The initial weights and biases are set with a small amount of values to prevent zero gradients. The number of input channels is based on the color channels of the input samples. Padding needs to be set both in convolutional operations and max-pooling operations. The kernel size of the second convolutional times the one in the first convolutional layer equals the number of spins on one dimension of the lattice in this work. The output of the second max-pooling operation should first be flattened to match the size of the fully connected layer following. The output of the second fully connected layer passes to the softmax operation to construct the cross entropy function. At last, the Adma algorithm is employed to minimize the cost function. We make the learning rate depend on the training iterations.

We should remark here that PCA plays the necessary but not critical role in terms of the finding result. It can be replaced by the other tools. For example, non-linear unsupervised learning methods can also play the role to determine the incipient phase boundaries [21, 31]. One may also apply clustering algorithms like K-Means [32], Density-based spatial clustering of applications with noise [33], Expectation-Maximization Clustering [34], Agglomerative Hierarchical Clustering [35], and so on to do the same work.

IV. ITERATIVE PROCEDURE TO FIND THE CRITICAL TEMPERATURE

Based on the results of PCA, the critical temperature locates in the region where the label is uncertain. The performance of the finding procedure mainly depends on the capability of learning of the CNN.

A. The convolutional neural network

CNN acts as a smart learner to extract the characteristic patterns of samples with labels of supervision [36, 37]. In this work, the spin configurations with similar characteristic patterns can be recognized as in one phase by the CNN with high accuracy (confidence). We elaborate the workflows to find the critical temperature and the learning process of the CNN in Fig. 3 with details in the caption. Since the CNN play the essential role in the finding procedure, we first show the workflow of one training iteration for the CNN: Samples at T/J → Convolutional operation-1 → ELU activation operation → Max pooling → Convolutional operation-2 → ELU activation operation → Max pooling → Fully connected operation-1 → ELU activation operation → Dropout operation → Fully connected operation-2 → Softmax operation → Constructing cross entropy function → Applying Adam algorithm to minimize cross entropy cost function. Then we give the brief instruction to the operations in the CNN.

As a prototype of neural networks to recognize images, the CNN takes advantage of the feature-map mechanism

to extract the characteristic patterns of input set [38]. The critical convolutional layer of CNN finishes the convolution operation by multiplying the corresponding pixels in a filter window of the input sample with the kernels which slide passing across the input sample until all the pixels are visited, and biases are generally added. The weights and biases of kernels are shared during the convolution operation which reduces the amount of parameters. To make the network be capable to extract more kinds of features, several filters are usually adopted. To extract more complex characteristics and improve the capability of recognition, deeper neural networks are recommendable candidates. These layers offers abstract inputs for next operations.

After the convolution operation, Exponential Linear Unit (ELU) defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \exp(x) - 1 & \text{if } x \leq 0 \end{cases} \tag{3}$$

is employed as the activation function [39]. Compared to the prevalent activation function Rectified Linear Unit (ReLU)[40], ELU has negative values which pushes the mean of activations closer to zero. This decreases the bias shift effect which benefits to speeding up learning with lower computational complexity [39]. Besides, the novel activation function: Swish performs better than ReLU for a number of scenes, and may be intriguing to be investigated in comparison with ReLU and ELU [41]. Max pooling operation converts high resolution samples to lower resolution counterparts. This operation reduces the amount of computation, and maintains the most important information in a compact form. It not only makes feature detection more robust to scale and orientation changes but is also beneficial to avoid overfitting. We do not exclude the probability to apply other pooling versions such as mean pooling or L2-norm pooling to be capable to perform well [42].

Fully connected layer is adopted to purify the information for next classification operations. Dropout is adopted not only to reduce overfitting but also to make the network less sensitive to small variations of input samples, and improves the generalization of this network. This can be done by setting the values of some randomly chosen neurons in the network to be zero [43]. Softmax is used as a generalization of sigmoid function to represent a probability distribution. Finally, Adam algorithm is employed to minimize the cross entropy cost function with respect to the labels of the training set [44]. We employ TensorFlow to implement the CNN in this work [45]. The workflow to find the critical temperature and the results are give in the following.

B. Workflow to find the critical temperature

The sketch of the workflow to find the critical temperature is shown in Fig. 3. Starting from the results of PCA, the details are listed bellow.

Step-1: $T_{low\ boundary}$, $T_{high\ boundary}$ and $T_m = (T_{low\ boundary} + T_{high\ boundary})/2$ (in units of J) are used to denote the low-temperature-phase boundary, high-temperature-phase boundary, and the average for the boundaries, respectively. Taking the samples labelled by PCA as the training set, we train the CNN until it can distinguish the whole training set with accuracy more than a high threshold S_{1t} within the number of S_{1n} training iterations;

Step-2: If **Step-1** is fulfilled, we use the trained CNN to recognize the number of S_{2m} samples at temperature T_m. If T_m is judged to 'low-temperature phase (or high-temperature phase)' with accuracy larger than 0.5, we further check the number of S_{2mm} samples at temperature $T_{mm} = (T_{low\ boundary} + T_m)/2$ (or $T_{mm} = (T_{high\ boundary} + T_m)/2$); While **Step-1** is not fulfilled, we purge the added samples in the last iteration and roll T_{mm} back to the last value to continue the recognition process from **Step-1**;

Step-3: If the test samples at temperature T_{mm} are recognized with accuracy more than a high threshold S_{3t}, we label the samples at temperature T_{mm} 'low-temperature-phase label (or high-temperature-phase label)' and put newly labelled samples into the training set to be used firstly to train the CNN in the next recognition iteration. Meanwhile we update the $T_{low\ boundary} = T_{mm}$ (or $T_{high\ boundary} = T_{mm}$); Else, we update $T_{mm} = (T_{low\ boundary} + T_{mm})/2$ (or $T_{mm} = (T_{high\ boundary} + T_{mm})/2$) and carry out the procedure in this step until T_{mm} is assigned a label with the threshold of accuracy S_{3t}. Then go to **Step-1** to begin the next recognition iteration.

When $T_{low\ boundary}$ and $T_{high\ boundary}$ approach T_c, considering the discussion in Sec. III, the CNN may be unable to distinguish the samples easily at T_{mm} in **Step-3**. Fortunately, in **Step-1**, the threshold of recognition S_{1t} is set high to correct this mistake. When the T_{mm} is incorrectly classified in **Step-3**, the CNN in the next **Step-1** can not be trained to reach the recognition threshold S_{1t} within the number of S_{1n} training iterations. Large S_{1n} is necessary to train the CNN but too large value may lead to overfitting. The results for the finding procedure are shown by examples in Fig. 4.

C. Result of the finding procedure

Based on the results of PCA, samples with $T/J \in (0, 1)$ and $T/J \in (4.5, 5.5)$ are assigned 'low-temperature-phase label' and 'high-temperature-phase label' respectively to be used as incipient training samples of the CNN. Then according to the iterative procedure from **Step-1** to **Step-3**, the evolution of the phase boundaries as a function of the number of iterations from **Step-1** to **Step-3** are shown by examples in Fig. 4. It can be seen that with the increasing of the iterations, the gaps between the phase boundaries would shrink and the average values of

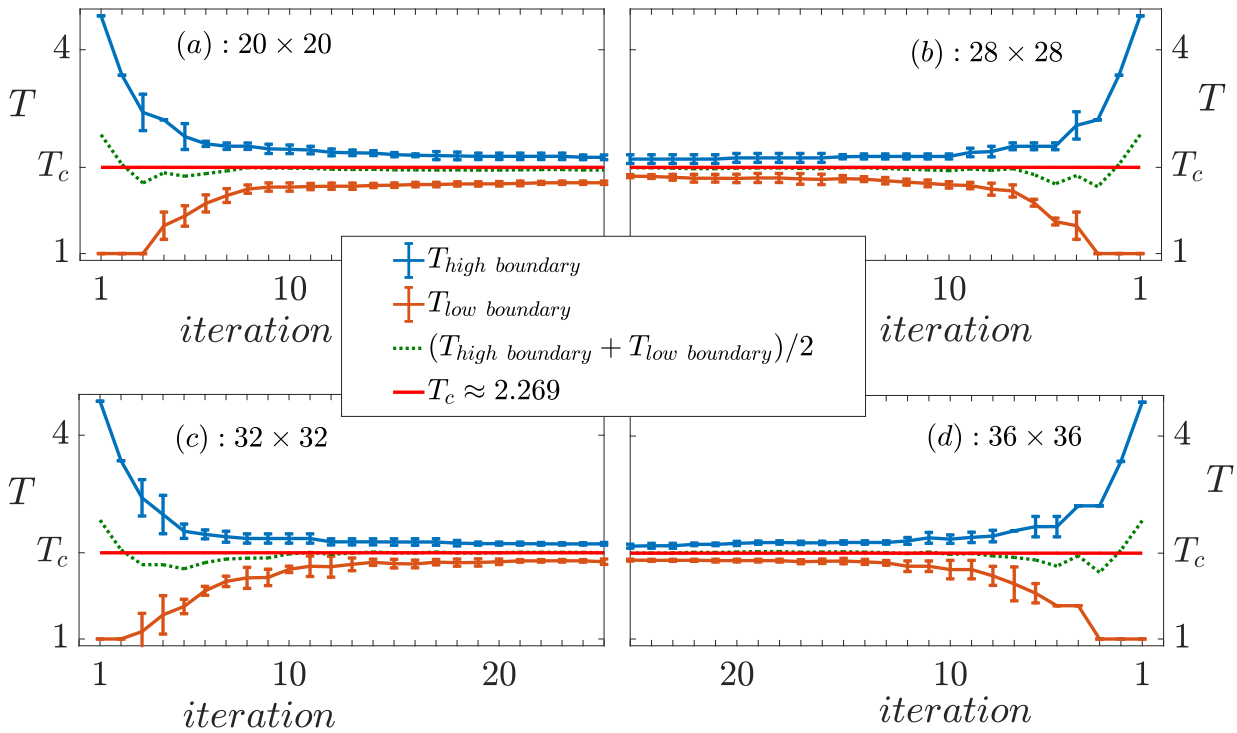

FIG. 4. The evolution of the errorbar graphs of the low- and high-temperature-phase boundaries versus the number of the iterations of recognition from **Step-1** to **Step-3**. The error bars are obtained by averaging over 6 independent rounds from **Step-1** to **Step-3**. 1000 incipient samples labelled by PCA are evenly distributed over 20 equidistant points in temperature region (0,1) and in (4.5,5), respectively. $T_{low\ boundary} = 1$, $T_{high\ boundary} = 4.5$ and $T_m = (T_{low\ boundary} + T_{high\ boundary})/2$ incipiently. During the training, batchsize=91 with learning rate varies as 0.001×0.99^k where k denotes the number of training iterations; S_{1t}=0.9, S_{1n}=0.9 in **Step-1**, S_{2m}=100, S_{2mm}=100 in **Step-2**, and S_{3t}=0.99 in **Step-3**. In the first convolutional layer, 32 convolutional kernels of size 4×4 are used, with strides=1 in two sliding directions. The initial weights are set with a small amount of random values with standard deviation of 0.1 and bias=0.1. The number of input channels is one since one single color channel is enough to distinguish the spin orientations. 'SAME' padding are chosen both in convolutional operations and max pooling. Max pooing takes 2×2 window, and strides equals 2 to do subsampling. 64 convolutional kernels are adopted in the second convolutional layer with the same setting of initial weights, bias, and padding to the first one. The initial weights and bias of the two fully connected layers are set same to those of convolutional layers above. There are 1024 nodes in both fully connected layers. 0.5 probability of zeroing is set for the dropout operation.

the phase boundaries would converge to the theoretical critical temperature $T_c \approx 2.269$ in these examples. The temperature gap between the boundaries evolves to a finite value for certain lattice size, and for larger lattice, the gaps become less at the same terminated iteration number. Namely, the larger spin configurations benefit the precision of the finding procedure.This also reflects the limitation of the CNN in learning the characteristic patterns of this two-dimensional Ising model. Further works may be valuable to study the relations between such limitations of recognition and neural networks.

V. THE CAPABILITY OF RECOGNITION OF THE CONVOLUTIONAL NEURAL NETWORK

From the work above, when the phase boundaries approach the critical temperature $T_c \approx 2.269$, the CNN seems blurred to distinguish which phase the samples belong to between the boundaries. It can be interpreted that the difference between the extracted characteristic patterns by the CNN is too tiny. For the same CNN, we conjecture that this capability of recognition may be linearly related to the magnetic susceptibility $\chi(T/J)$ which is related directly to the spin configurations. Thus we compare the capability of recognition of the CNN with $\chi(T/J)$ from analytical results [29, 30] versus tempera-

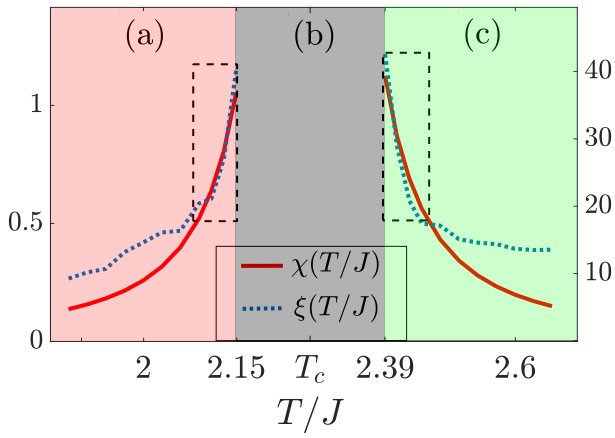

FIG. 5. The adjusted capability of recognition of the CNN (same to the one used in Fig. 4 (c)) versus temperature in the vicinity of the critical temperature on two sides. $\mathcal{R}(T/J)$ denotes the average of 160 training iterations cost until the CNN is capable to recognize 100 test samples with the accuracy more than 0.98, and $\xi(T/J)$ denotes the linear function of $\mathcal{R}(T/J)$. The temperature range $T/J \in (0.032, 4.569)$ is evenly discretized into 100 temperature nodes with 50 spin configurations on each node to offer the training set. The 100 test samples are randomly selected from 1000 samples at each temperature node which are not contained in the training set. As comparison, the results of the analytical results for magnetic susceptibility $\chi(\delta t)$ is also shown, here $\delta t = |T/J - T_c|$. In the vicinity of the critical temperature, the magnetic susceptibility $\chi(\delta t) \approx c_{0\pm}\delta t^{-7/4} + c_{1\pm}\delta t^{-3/4} + O(1)$ [29, 30]. Here the subscript $+(-)$ stands for that $T/J \to T_c$ from above (below), and $c_{0+} \approx 0.96258$, $c_{0-} \approx 0.02554$, $c_{1+} \approx 0.07499$ and $c_{1-} \approx 0.00199$. (a) comparison of $\xi(T/J) = 0.056\mathcal{R}(T/J)$ (blue dotted line) and the analytical results (red solid line) when $T/J < T_c$; (c) comparison of $\xi(T/J) = 2.10\mathcal{R}(T/J) - 5.00$ with the analytical results. (b) Cases frequently occur at this region that the CNN can not recognize the test samples with the high accuracy as in (a) and (b) within 260 training iterations.

ture in Fig. 5. The capability of recognition is indicated by the average of $\mathcal{R}(T/J)$ which is the number of training iterations cost until the CNN is able to recognize the samples with high accuracy at T/J. We adjust $\mathcal{R}(T/J)$ as $\xi(T/J) = a\mathcal{R}(T/J) + b$ near T_c, where a and b are the linearly regulative factor and bias, respectively. On both sides of T_c, when the boundaries approach the critical temperature as shown in Fig. 5 (b), a lot of cases occur that the CNN can not recognize the samples in this region with high accuracy. This limitation along with the asymptotic gaps for large iterations in Fig. 4 may both reflect the limitation of this CNN in recognizing the characteristic patterns of this Ising model. Only in temperature regions as shown in the dashed-edge rectangles in Fig. 5 (a) and (c), $\mathcal{R}(T/J)$ can be linearly adjusted to match $\chi(T/J)$ well.

The map between $\chi(T/J)$ and the performance of the CNN is elusive due to the complexity of the network. Complex relation between $\mathcal{R}(T/J)$ and $\chi(T/J)$ may exist but deep investigations are needed. Even $\mathcal{R}(T/J)$ is not adjusted to match magnetic susceptibility in the whole temperature range very well, this work brings the motivation to investigate scaling law near critical points by means of machine learning techniques, including various CNNs with different structures and parameters, auto encoder, restricted Boltzmann machine and so on [37]. Such investigations may help one get insight into physical systems.

VI. CONCLUSION

We have proposed an iterative methodology employing dimension-reduction algorithm-principal component analysis (PCA), to assign labels for incipient training samples which would be employed by the machine learning tool-convolutional neural network (CNN), to find the critical temperature for the classical two-dimensional Ising model. Starting from the results of PCA, we propose the scheme to pick the temperatures at which the samples be tried to be labelled by the trained CNN. Once the samples are labelled with high accuracy, they would be added into the training set and the corresponding phase boundaries would be updated by their temperatures. Along with the procedure of recognizing and updating, the boundaries of the phases would evolve toward each other, and the average of them would converge to the theoretical value. We find that with the same conditions, larger lattice benefits the finding performance. At last we show that in a temperature region near the critical temperature, the capability of recognition of the CNN can be linearly mapped to magnetic susceptibility well. This may inspire one to investigate scaling laws by machine learning, and the general connections between statistical models and neural networks. With the same methodology, PCA and the CNN can be replaced by other machine-learning tools with similar functions, and may also perform well.

ACKNOWLEDGEMENTS

The Xiaolong Zhao thanks Hongchao Zhang at Dalian University of Technology for helpful discussions. This work is supported by the National Natural Science Foundation of China (Grant No. 11725417, 11575027), NSAF (Grant No. U1730449), and Science Challenge Project (Grant No. TZ2018005).

[1] J. Carrasquilla and R. G. Melko, Machine Learning Phases of Matter, Nat. Phys. **13**, 431 (2017).

[2] E. P. L. van Nieuwenburg, Y. H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**,

435 (2017).
[3] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine learning phases of strongly correlated fermions[J]. Physical Review X **7**, 031038 (2017).
[4] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. **118**, 216401 (2017).
[5] Y. Zhang, R. G. Melko, and E.-A. Kim, Machine learning Z_2 quantum spin liquids with quasiparticle statistics, Phys. Rev. B **96**, 245119 (2017).
[6] P. Zhang, H. Shen, H. Zhai, Machine Learning Topological Invariants with Neural Networks, Phys. Rev. Lett. **120**, 066401 (2018).
[7] N. C. Costa, W. Hu, Z. J. Bai, R. T. Scalettar, and R. R. P. Singh, Principal component analysis for fermionic critical points,Phys. Rev. B **96**, 195138 (2017).
[8] L. Wang, Discovering Phase Transitions with Unsupervised Learning, Phys. Rev. B **94**, 195105 (2016).
[9] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).
[10] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).
[11] C. Wang and H. Zhai, Machine learning of frustrated classical spin models. I. Principal component analysis,Phys. Rev. B **96**, 144432 (2017).
[12] K. Pearson, On lines and planes of closest fit to systems of points in space, Philos. Mag. **2**, 559 (1901).
[13] I. Jolliffe, Principal Component Analysis (Wiley, 2002).
[14] C. M. Bishop, Neural Networks for Pattern Recognition, 1st ed. (Clarendon, 1996).
[15] L. Li, T. E. Baker, S. R. White, and K. Burke, Pure density functional for strong correlation and the thermodynamic limit from machine learning, Phys. Rev. B **94**, 245129 (2016).
[16] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).
[17] P. Mehta and D. J. Schwab, , ,and, An exact mapping between the variational renormalization group and deep learning, arXiv: 1410. 3831 (2014).
[18] E. M. Stoudenmire and D. J. Schwab, Supervised Learning with Quantum-Inspired Tensor Networks, Adv. Neural Inf. Process. Syst. **29**, 4799 (2016).
[19] F. Schindler, N. Regnault, and T. Neupert, Probing many-body localization with neural networks, Phys. Rev. B **95**, 245134 (2017).
[20] P. Suchsland and S. Wessel, Parameter diagnostics of phases and phase transition learning by neural networks, Phys. Rev. B **97**, 174435 (2018).
[21] C. Casert, T. Vieijra, J. Nys, J. Ryckebusch, Interpretable Machine Learning for Inferring the Phase Boundaries in a Non-equilibrium System, arxiv: 1807. 02468 (2018).
[22] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).
[23] M. A. Nielsen, Neural Networks and Deep Learning (Determination Press, 2015)
[24] L. Onsager, Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition, Phys. Rev. **65**, 117 (1944).
[25] N. Metropolis, S. Ulam, The Monte Carlo Method, J. Am. Stat. Assoc. **44**, 335 (1949) .
[26] K. Binder, Applications of Monte Carlo methods to statistical physics, Rep. Prog. Phys. **60**, 487 (1997).
[27] K. Binder and D. W. Heermann, Monte Carlo Simulation in Statistical Physics, (Springer-Verlag, 1988).
[28] E. Ising, Beitrag zur Theorie des Ferromagnetismus,,Zeits, f. Physik **31**, 253 (1925).
[29] E. Barouch, B. M. McCoy, and T. T. Wu, Zero-Field Susceptibility of the Two-Dimensional Ising Model near T_c, Phys. Rev. Lett. **31**, 1409 (1973).
[30] T. T. Wu, B. M. McCoy, C. A. Tracy, and E. Barouch, Spin-spin correlation functions for the two-dimensional Ising model: Exact theory in the scaling region, Phys. Rev. B **13**, 316 (1976).
[31] L. McInnes and J. Healy, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, arXiv: 1802. 03426 (2018).
[32] C. Ding and X. He, K-means clustering via principal component analysis, LBNL-52983. Intl Conf. Machine Learning (2004).
[33] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD'96), 226-231 (1996).
[34] T. K. Moon. The expectation-maximization algorithm. IEEE Signal Process Mag **13**, 47 (1996).
[35] M. Steinbach, G. Karypis, and V. Kumar, A comparison of document clustering techniques, KDD Workshop on Text Mining 400 (2000).
[36] Y. LeCun, L. Bottou, Y. Bengio, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, **86**, 2278 (1998).
[37] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016).
[38] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, 1097-1105 (2012).
[39] D. A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus) arXiv: 1511. 07289 (2015).
[40] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th international conference on machine learning (ICML-10), 807-814 (2010).
[41] P. Ramachandran, B. Zoph, Q. V. Le, , ,and, Swish: a Self-Gated Activation Function, arXiv: 1710. 05941 (2017).
[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv: 1207. 0580 (2012).
[43] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research **15** 1929 (2014).
[44] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv: 1412. 6980 (2014).
[45] Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (2015); http://tensorflow.org.