Phase transition encoded in neural network

Kouji Kashiwa,[1, *] Yuta Kikuchi,[2, †] and Akio Tomiya[2, ‡]

[1]*Fukuoka Institute of Technology, Wajiro, Fukuoka 811-0295, Japan*

[2]*RIKEN BNL Research center, Brookhaven National Laboratory, Upton, NY, 11973, USA*

Abstract

We discuss an aspect of neural networks for the purpose of phase transition detection. To this end, we first train the neural network by feeding Ising/Potts configurations with labels of temperature so that it can predict the temperature of input. We do not explicitly supervise whether the configurations are in ordered/disordered phase. Nevertheless, we can identify the critical temperature from the parameters (weights and biases) of trained neural network. We attempt to understand how temperature-supervised neural networks capture the information of phase transition by paying attention to what quantities they learn. Our detailed analyses reveal that they learn different physical quantities depending on how well they are trained. Main observation in this study is how the weights in the trained neural-network can have information of the phase transition in addition to temperature.

* kashiwa@fit.ac.jp

† yuta.kikuchi@riken.jp

‡ akio.tomiya@riken.jp

I. INTRODUCTION

Exploring phases of matters is one of the most important tasks to reveal infrared structure of the underlying microscopic physical system. Their phases are classified based on the symmetries that the theory possesses [1, 2]. In a theory with several distinct phases, phase transitions occur at their boundaries. Among them, the nature of second order phase transition is solely determined by number of dimensions, global symmetries of underlying theory independent of its microscopic details, i.e., classified by the universality class. In reality, however, analytic determination of various phases or detection of phase transitions based on the data of microscopic theory is generally a hard problem because we mostly find it difficult to exactly solve them or identify the corresponding infrared theory. Therefore, tremendous amount of works have been devoted to unravel them with numerical approaches. An obvious and major obstacle is that the larger the number of degrees of freedom grows the harder the numerical analyses become.

Machine learning has grown up rapidly in the field of computer science and made prominent successes in pattern recognition, image processing, etc. Recently, we have witnessed that the machine learning has also been applied in various branches of physics. Detection of phase transitions is one of the intriguing examples that machine learning may make new progresses, and several approaches have already been proposed and examined in simple models such as spin systems. In those works, the trainings are carried out with supervision [3–18] or without one [19–24]. Here, the input data is prepared independently of the neural network or its training process. For instance, the Monte Carlo simulation or experimental data based on the physical system of our interest provide the necessary input data.

One approach to detect a phase boundary of a physical system is a supervised binary classification, where a neural network is trained so that it can distinguish its ordered and disordered phases. Indeed, it reasonably detects the phase transition in several models from their raw data [3]. More recently, some implicit connections between latent variables and physical quantities have been extensively studied toward the ultimate goal of eliminating black-box nature of machine learning technique [3, 12, 15, 18]. It also succeeded in detecting non-standard phase transition such as topological phase transitions and BKT phase transition with the help of feature engineerings based on physical insights. [10, 12, 14, 16, 25, 26].

Another intriguing approach was proposed in [4] to detect the phase transition by specu-

lating that the information of order parameters are encoded in the weights of neural network as a consequence of training. They attempted to identify the critical temperature of the two dimensional Ising model based on supervised machine learning. A fully-connected as well as convolutional neural network are trained in such a way that it can correctly predict the target temperature of input spin configuration. It is surprising that they succeeded in extracting the phase transition temperature from its weight because they did not feed any direct information about phase transition during the training. It implies that the network *spontaneously captures the phase transition* and encodes it in the machine parameters along the process of supervised learning of temperature although the underlying mechanism of the outcome was remained unresolved.

The purpose of this article is to understand the mechanism of this phase transition detection. Understanding what this approach captures will be useful when we try to apply it to other unknown systems because it may not detect the phase transition correctly if it is triggered by different physical mechanisms such as quasi long-range order (BKT transition) or topology. Indeed, it turns out that the method captures different physical quantities, i.e., the features of input configurations, depending on how we train the neural network. In order to illustrate the idea, we pay attention to simplified neural network architectures embodying the essence of temperature prediction.

II. CRITICAL TEMPERATURE PREDICTION

We consider two dimensional Ising model, described by the Hamiltonian $H(\{\sigma_i\}) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$, where the coupling constant J is taken to be positive. $\sigma_i \in \{-1, 1\}$ represents a spin degree of freedom living on a site of a square lattice of size $L \times L$. We impose periodic boundary condition on the spin variables. The sum is taken over nearest neighboring sites. We redefine the Hamiltonian as

$$H(\{\sigma_i\}) = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{1}$$

by absorbing the coupling constant into the inverse temperature $K := J/k_\mathrm{B}T$ on the Boltzmann weight $e^{-KH}/\mathcal{Z}$, where $\mathcal{Z} = \sum_{\{\sigma_i\}} e^{-KH(\{\sigma_i\})}$ is the partition function.

We attempt to detect its critical temperature associated with the second-order phase transition. Nevertheless, our first step is to construct a *thermometer* by employing a supervised

feed-forward neural network. Then, we will examine the weights and biases of the trained neural networks, and attempt to identify the critical temperatures of two dimensional Ising model and 3-state Potts model.

We generate 2000 Ising spin configurations at each temperature by employing the Metropolis-Hasting algorithm. They are fed to a neural network along with the 20 target temperatures. We implement two types of neural network architectures by using KERAS package with TENSORFLOW as the backend: fully-connected and convolutional neural networks.

The former consists of fully-connected hidden and output layers as follows:

$$
\begin{bmatrix}
\mathcal{I} = \left\{ \{\sigma_i\} \middle| \text{ Ising configs on } L \times L \text{ lattice.} \right\} \\
\downarrow \begin{cases} \text{Fully-connected (Dense) layer} \\ \text{Softmax activation} \end{cases} \\
x_a \in [0,1]^{N_\mathrm{h}} : \text{hidden units} \\
\downarrow \begin{cases} \text{Fully-connected (Dense) layer} \\ \text{Softmax activation} \end{cases} \\
y_K \in [0,1]^{N_\mathrm{o}} : \text{output}
\end{bmatrix}
\tag{2}
$$

Let us describe it in detail here. We denote input degrees of freedom by $\{\sigma_i\}$ ($i = 1, \ldots, L \times L$), which would be spins in case of Ising model. A hidden unit x_a ($a = 1, \ldots, N_\mathrm{h}$) is given by,

$$
x_a = \mathrm{softmax}(w_{ai}^{(1)}\sigma_i + b_a^{(1)}) := \frac{\mathrm{e}^{w_{ai}^{(1)}\sigma_i + b_a^{(1)}}}{\sum_a \mathrm{e}^{w_{ai}^{(1)}\sigma_i + b_a^{(1)}}},
\tag{3}
$$

where repeated indices are summed. $w_{ai}^{(1)}$ and $b_a^{(1)}$ are weights and biases of the first layer, respectively. In terms of weights $w_{\alpha a}^{(2)}$ and biases $b_\alpha^{(2)}$ of the second layer, variables y_K ($K = 1, \ldots, N_\mathrm{o}$) of output layer takes the same form as the hidden variables,

$$
y_K = \mathrm{softmax}(w_{Ka}^{(2)}x_a + b_K^{(2)}).
\tag{4}
$$

Based on the output $\{y_K\}$, the temperature of input configuration is determined via

$$
K^{\mathrm{output}} \equiv \arg\max_K(y_K),
\tag{5}
$$

namely, the temperature α with the highest "probability" y_K is picked as the output temperature. The training is implemented by tuning the weights and biases with the Adam

optimizer [27] and the error function,

$$E(y_K, \mathbf{1}_{K=K_i^{\mathrm{target}}}) = -\frac{1}{N_{\mathrm{o}}} \sum_i \mathbf{1}_{K=K_i^{\mathrm{target}}} \ln y_K \qquad (6)$$

which is the cross entropy between the target indicator function and output distribution function, where i denotes the label of input configurations. The indicator function is defined by

$$\mathbf{1}_{K=K_i^{\mathrm{target}}} = \begin{cases} 1 & (K = K_i^{\mathrm{target}}) \\ 0 & (\mathrm{otherwise}) \end{cases} \qquad (7)$$

The convolutional neural network consists of three convolutional layers followed by the final fully-connected layer (16). We shall discuss it in more detail in Sec. III B.
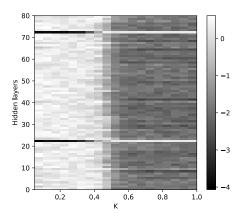
Summary of our procedure is as follows:

Step 1: Gather configurations via the standard Markov-chain Monte-Carlo method: We do not need the machine learning in this step. Data set $\{\sigma_i\}$ for each spin configuration at fixed K is stored.

Step 2: Train the neural-network as a thermometer: The input is the spin configuration and the output is the predicted temperature.[1]

Step 3: Analyze trained weights and biases appearing in the neural-network: We discuss how the machine parameters contain information of the phase transition.

A. 2D Ising model

We briefly discuss how the phase transition detection works. We first take a look at 2D Ising model, which was already studied in Ref. [4] with 100 target temperatures and its weights behave like an order parameter, i.e. spontaneous magnetization. Since our primary purpose is to understand its mechanism rather than to quantitatively estimate the critical temperature, we reduce the number of target temperatures to 20: $K = 0.05, 0.1, 0.15, \ldots, 1.0$. We perform the supervised training using neural network (2), with

---

[1] Because of the overlap of the energy probability distributions between each temperature, there is the upper-bound of accuracy of the prediction even if we ideally train the neural-network. See Appendix A for detail.
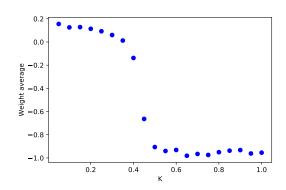
Figure 1. The weights in the fully-connected layer and their average after the training in case of 2D Ising model. The horizontal axes represent the temperatures K of input configurations. The vertical axis in the left panel corresponds to components connected to hidden units in the neural network. The vertical axis in the right panel shows average of weights for each K. The average value of weights significantly changes around the exact critical temperature $K_c^{\text{exact}} \simeq 0.4407$.
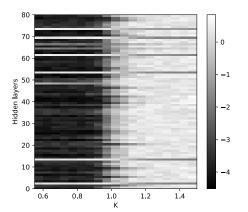
lattice size $L = 16$, the number of hidden units $N_h = 80$, and $N_o = 20$ corresponding to the 20 target temperatures. The critical temperature is exactly known to be $K_c^{\text{exact}} = \frac{1}{2} \ln(\sqrt{2}+1) \simeq 0.4407$ [28]. The weights of second layer after the training is shown in Fig. 1. In Ref. [4], the critical temperature was predicted by fitting the sum of the weights by $c_1 \tanh[c_2(K - K_c)] - c_3$ with free parameters c_1, c_2, c_3 for lattice sizes $L = 8, 16, 32$. Indeed, the average of the final weights appears to behave like an order parameter (right panel of Fig 1). We will discuss the detailed structure of the weights in the next section.

B. 2D 3-state Potts model

Before getting into the detail of learning mechanism of critical temperature of 2D Ising model, we take a look at another example, 2D 3-state Potts model. The Hamiltonian is given by

$$H(\{\Phi_i\}) = -\sum_{\langle i,j \rangle} \delta(\Phi_i, \Phi_j), \tag{8}$$

where Φ_i takes three values, a generalization of Ising spin σ_i. Hence, configurations $\{\Phi_i\}$ labeled by temperatures K are the inputs of neural network. The 2D 3-state Potts model is known to exhibit the second order phase transition at $K_c \simeq 1.0050$ because of the fluctuation
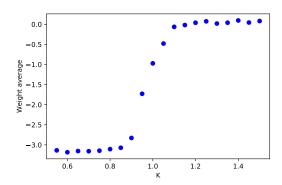
Figure 2. The weights in the fully-connected layer and their average after the training in case of 2D 3-state Potts model (left panel). The average value of weights significantly changes around the critical temperature $K_c \simeq 1.0050$ (right panel).

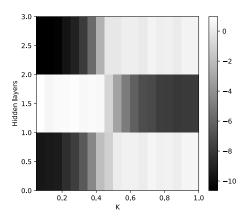unlike the simple Landau theory [29–31].

After a training with the same neural network architecture as that for 2D Ising model, we obtain the weights and their average shown in Fig. 2. We again find a drastic change in the weight structure around the critical temperature.

III. DISCUSSION

So far, we have observed that the change in value of weights in the second layer signals the phase transition, which implies that the information of critical temperature is somehow encoded in the trained neural network. In what follows, we carefully examine the trained fully-connected/convolutional neural networks and attempt to understand what (physical) quantity they extract as a feature of input configurations and how it is related to temperature prediction as well as the critical temperature detection [15, 17].

A. Magnetization encoded in neural network

Since the order parameter of phase transition in the 2D Ising model is the spontaneous magnetization, it sounds natural that it is encoded in the neural network after the training. To give a quantitative argument we construct a simplified model by examining the weights and biases of training neural network in case of 2D Ising model. First, we reduce the number of hidden unit in (2) from 80 to 3 for the sake of simplicity. We notice that it still captures
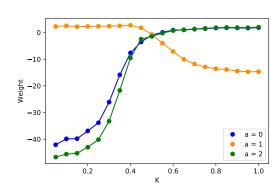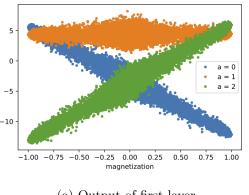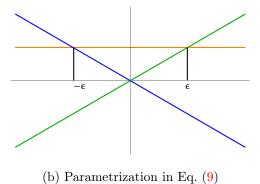
Figure 3. The fully-connected weights $w_{Ka}^{(2)}$ of trained neural networks with three hidden units ($N_{\mathrm{h}} = 3$). The horizontal axes represent temperatures K of input 2D Ising configurations. The structure change is still observed around critical temperature. One of the weights has an almost opposite temperature dependence, which also appeared in Fig. 9. See the main text for detailed discussion.

(a) Output of first layer

(b) Parametrization in Eq. (9)

Figure 4. (a) Correlations between the output of the first layer and magnetization of input configuration. The horizontal axis represents the magnetization per site of input Ising spin configuration and the vertical axis is $\tilde{x}_a = w_{ai}^{(1)}\sigma_i + b_a^{(1)}$. (b) The model parametrization, where each line corresponds to each row of Eq. (9).

the critical temperature as shown in Fig. 3.

Before modeling the second layer, we examine the characteristics of the first layer. Figure 4 shows the correlation between the output of the first layer $\tilde{x}_a := w_{ai}^{(1)}\sigma_i + b_a^{(1)}$ of Eq. (3) and magnetization density of the input Ising spin configuration. From this observation, we

model these output by three lines linear in magnetization $m(\{\sigma\})$ as shown in Fig. 4b:

$$\tilde{x} = \begin{pmatrix} \tilde{x}_0 \\ \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} -m \\ \epsilon \\ m \end{pmatrix}, \tag{9}$$

where $\epsilon > 0$ is a constant. Furthermore, as an activation function, we use the max function, that assigns 1 to the maximum entry and 0 to the rest, instead of softmax for our purpose; this replacement does not change our final result. $x_a = \max(\tilde{x}_a)$ yields the following vectors depending on magnetization of inputs m,

$$m < -\epsilon : \ x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad -\epsilon \leq m < \epsilon : \ x = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \epsilon \leq m : \ x = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{10}$$

The parameter ϵ may be interpreted as a threshold magnetization separating the ferromagnetic and paramagnetic phases [3].

Having understood the magnetization dependence of the three hidden units, we next analyze the second layer. The weights in the layer are given in Fig. 3. We divide the temperature into three pieces: low, critical, and high temperatures, respectively represented by the following vectors,

$$\text{Low K}: \ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \text{Critical K}: \ \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \text{High K}: \ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{11}$$

in N_o-dimensional output space. This procedure effectively reduces the output dimension N_o to 3. According to Fig. 3, we parametrize the weights in the following way,

$$w^{(2)} = \begin{pmatrix} -\Delta & 0 & -\Delta \\ -\delta & -\delta & -\delta \\ 0 & -\Delta & 0 \end{pmatrix}, \tag{12}$$

with $\Delta > \delta > 0$. Elements of $(K \times a)$-matrix $w^{(2)}$ are the weights $w_{Ka}^{(2)}$. We neglect the biases as they are much smaller than the weights. Precise parametrization is not necessary for the following discussion. Then, $y_K = \max(\tilde{y}_K) = \max(w_{Ka}^{(2)} x_a + b_K^{(2)})^2$ yields the following

---

[2] Remember that the max function here is defined to be a function assignning 1 to the maximum entry and 0 to the rest.

output,

$$m < -\epsilon : \ y_K = \max(w_{K0}^{(2)}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} , \tag{13}$$

$$-\epsilon \le m < \epsilon : \ y_K = \max(w_{K1}^{(2)}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \tag{14}$$

$$\epsilon \le m : \ y_K = \max(w_{K2}^{(2)}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} , \tag{15}$$

which are predicted as low, high, and low temperature, respectively. Since the configurations with $m < -\epsilon$ and $m > \epsilon$ are in the ordered phase, they are correctly predicted as low temperature phase. The configurations with $-\epsilon \le m < \epsilon$ is also correctly predicted as high temperature. However, it cannot detect the intermediate temperature, i.e., the critical temperature in this case. Furthermore, the different temperatures within the ordered/disordered phase are not distinguished by the trained neural network even if we introduce more than three target temperatures in Eq. (11) because upper branch of the weights (and biases) are almost temperature independent above and below K_c, respectively, as seen in Fig. 3. Therefore, the trained network is capable of distinguishing only high or low temperature.

One might think that this is due to the fact that we have single threshold parameter ϵ corresponding to the three hidden unit. Actually, we can increase the number of threshold parameters by introducing more hidden units. But, it does not lead to higher resolution of temperatures. In fact, even if we increase the number of hidden units, the weights in the second layer show only two patterns of temperature dependence: most of them behave like blue and green curves and the rest behave like the orange curve in Fig. 3. This is exactly what is observed in Fig. 1. Increasing the hidden units simply results in duplicating the second layer's weights that are already observed in Fig. 3, and consequently, the predicted temperature is either high or low no matter how many hidden units are introduced.

What we have learned from the above analyses is as follows: The network obtained the information of magnetization which manifests itself in the output of the first layer. However, it is hard for this network to discriminate each temperature except for the difference

between ordered and disordered phases based on the magnetization. From the viewpoint of machine learning parameters, this is due to the fact that the weights of the second layer are temperature independent except around the critical temperature.

B.   Energy and temperature prediction

We have found in the last subsection that the magnetization of 2D Ising model was build in the temperature-supervised fully-connected neural network, that in turn allows us to read off the critical temperature from its weight structure. While the critical temperature seems to be well detected, we have not discussed the temperature prediction itself. Interestingly, the accuracy of temperature learning can be theoretically computed to be 40.1% in case of 2D Ising model under the above setup, giving the upper bound for the accuracy of temperature prediction by machine learning [32] (see Appendix A for details). However, we note that the test accuracy of temperature prediction by fully-connected neural network is 16.8%, which is not even close to theoretically predicted accuracy 40.1%. We could train better from the viewpoint of temperature prediction although we did not need for the phase transition detection by extracting magnetization as we have already seen. What happens if we design the neural network architecture in such a way that the temperature prediction accuracy improves?

To answer the question, we use a convolutional neural network shown below, enabling us

to achieve higher accuracy in two dimensional image recognition.

$$\left[\begin{array}{l} \mathcal{I} = \left\{ \{\sigma_i\} \,\middle|\, \text{Ising configs on } L \times L \text{ lattice.} \right\} \\[4pt] \downarrow \left\{ \begin{array}{l} \text{Convolution}_{[(s_1,s_1)\text{-filter}, \, (s_1,s_1)\text{-stride}, \, C_1\text{-channels}]} \\ \text{ReLU activation} \\ \text{Convolution}_{[(s_2,s_2)\text{-filter}, \, (s_2,s_2)\text{-stride}, \, C_2\text{-channels}]} \\ \text{ReLU activation} \\ \text{Convolution}_{[(s_3,s_3)\text{-filter}, \, (s_3,s_3)\text{-stride}, \, C_3\text{-channels}]} \\ \text{ReLU activation} \\ \text{Flatten} \end{array} \right. \\[4pt] x_a \in \mathbb{R}^{N_{\mathrm{h}} = \, L^2/(s_1 s_2 s_3)^2 \times C_3} : \text{hidden units} \\[4pt] \downarrow \left\{ \begin{array}{l} \text{Fully connected layer} \\ \text{Softmax activation} \end{array} \right. \\[4pt] y_\alpha \in [0,1]^{N_{\mathrm{t}}} : \text{output} \end{array} \right] \qquad (16)$$

The network has three convolutional layers (Conv2d layers from KERAS) with square filters with size (s_i, s_i), strides (s_i, s_i) and the number of channels C_i, each of which is followed by ReLU activation function. Then, the output is passed to fully-connected layer, whose input and output are of out interest and analyzed in detail. The output of the second last layer, denoted by x_a, is given by
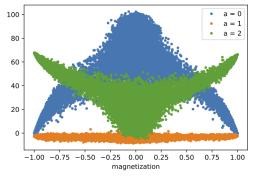
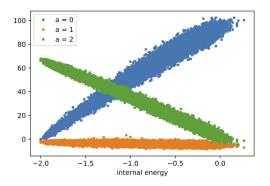$$x_a = \text{ReLU}(\tilde{x}_a) = \text{Max}(\tilde{x}_a, 0), \qquad (17)$$

takes a value in $[0, \infty)$. $\tilde{x}_a$ is an output of the previous layer before passed to the ReLU activation. Then, the output x_a plays a role of input of the fully-connected layer to give a prediction of temperature via (4) and (5).

Following the last subsection, we put three hidden units playing a role of input of the fully-connected layer and attempt to understand what the neural network learns as a result of the training. To this end, the parameters are set as follows:

$$L = 16, \quad (s_1, s_2, s_3) = (2, 2, 4), \quad (C_1, C_2, C_3) = (64, 32, 3), \qquad (18)$$

leading to the number of hidden units $N_{\mathrm{h}} = 3$. With this setup we achieved 35.9% of accuracy on a test data set. It is twice higher than fully-connected layers, although still not very close to the bound partially due to the lack of statistics and very small number of hidden units.

(a) Correlation with magnetization.   (b) Correlation with internal energy.

Figure 5. Correlations between physical quantities and the weights of fully-connected layer in the trained neural network with five hidden units, each of which corresponds to vertical axis of the left and right panel. The horizontal axes in (a) and (b) show magnetization and internal energy of input 2D Ising configurations, respectively.

Figures 5 shows the correlation of the output of first layer with magnetization (Fig. 5a) and internal energy (Fig. 5b), respectively. We clearly see the transition in what the neural network has learned. The first-layer outputs are now proportional to the internal energy rather than the magnetization of input configurations. Also, we notice that the weights of the second layer obtain mild dependence on temperature, implying the information of the critical temperature is blurred (Fig. 7); as we shall see momentarily, the oscillating orange line is irrelevant in the temperature prediction.

We now proceed to a simplified parametrization of the last two layers. Here, $(w_{ai}^{(1)}, b_a^{(1)})$ and $(w_{Ka}^{(2)}, b_K^{(2)})$ stand for weights and biases in the second last and last layer, respectively. As we have already mentioned and checked in Fig. 5b, the output of the first layer is proportional to energy $E(\{\sigma_i\})$ of input configuration $\{\sigma_i\}$. After passing to the ReLU activation,

$$x_a = \text{ReLU}(w_{ai}^{(1)}\sigma_i + b_a^{(1)}), \quad x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} E + 2\epsilon \\ 0 \\ -\phi E \end{pmatrix}, \tag{19}$$

where ϵ and ϕ are positive constants. Domain of E is restricted so that x_a does not take a negative value.

Having observed that the input of the fully-connected layer is proportional to E, let us consider what would be an optimal estimation of the temperature of an input configura-

13

tion $\{\sigma_i\}$ [32]. The probability $P(\{\sigma_i\}; K)$ that the configuration $\{\sigma_i\}$ appears at temperature K is given by

$$P(\{\sigma_i\}; K) = \frac{\mathrm{e}^{-KE(\{\sigma_i\})}}{\mathcal{Z}(K)}. \tag{20}$$

Therefore, the likelihood that $\{\sigma_i\}$ is generated at temperature K is represented by the following "probability",

$$y_K^{\mathrm{theory}} = \frac{P(\{\sigma_i\}; K)}{\sum_{K'} P(\{\sigma_i\}; K')} = \mathrm{softmax}(-KE + F), \tag{21}$$

where $F = -\ln \mathcal{Z}(K)$ is the free energy and K' is summed over the target temperatures. Then, we obtain the estimated temperature,

$$K^{\mathrm{output}} = \arg\max_{K} \left[\mathrm{softmax}(-KE + F))\right]. \tag{22}$$

We remark here that the free energy is a function of temperature K and holds the information of phase transition although it does not exhibit genuine singularity in finite systems.

Based on the above consideration, we guess fully-connected weights of our neural network so that the resultant output behaves like (21). Then, we compare with the actual parameters of the trained network. To this end, we parametrize the weights as follows:

$$w_{K0}^{(2)} = -\phi G(K) - pK + q, \quad w_{K2}^{(2)} = -G(K) + rK + s, \tag{23}$$

with constant parameters p, q, r, s and a common nonlinear function $G(K)$. The orange curve in Fig. 7, $w_{K1}^{(2)}$, is not relevant to our consideration because $x_1 = \mathrm{ReLU}(\tilde{x}_1) = 0$. We use an observation from the simulation to neglect the bias, where it is much smaller than the value of weights.

Then, $y_K = \mathrm{softmax}(w_{Ka}^{(2)} x_a + b_K^{(2)})$ with Eqs. (19) and (23) yields the following output;

$$y_K = \mathrm{softmax}(-(p+r)KE - 2\epsilon(\phi G(K) + pK)), \tag{24}$$

where we dropped K-independent terms because they do not affect the outcome of the max function. The expression indeed takes the form of y_K^{theory} (21) if $-2\epsilon(\phi G(K)+pK) = (p+r)F$ is satisfied. We plot the theoretically predicted weights in Fig. 6:

$$w_{K0}^{(2)} = \frac{p+r}{2\epsilon} F(K) + q, \quad w_{K2}^{(2)} = \frac{p+r}{2\epsilon\phi} F(K) + \left(\frac{p}{\phi} + r\right)K + s. \tag{25}$$
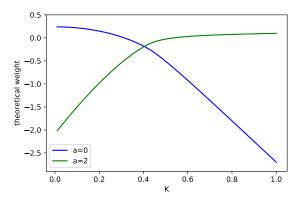
Figure 6. Blue and green curves correspond to $w_{K0}^{(2)}$ and $w_{K2}^{(2)}$ in (23), respectively. The horizontal axes represent temperatures K of the input 2D Ising configurations. See the main text for detail.
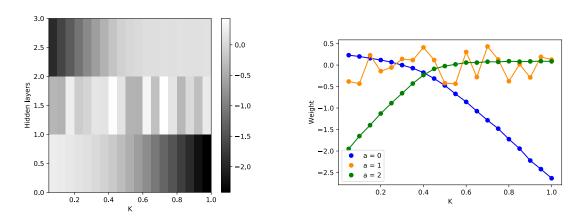

Figure 7. The fully-connected weights of trained convolutional neural networks with three hidden units ($N_{\mathrm{h}} = 3$). The horizontal axes represent temperatures K of the input 2D Ising configurations. The value of the first and last weights gradually change as temperature increases in contrast to the previous case. See the main text for detailed discussion.

$F(K)$ is the free energy analytically calculated in 2D Ising model. It agrees very well with the actual weights obtained from the trained neural network shown as blue and green curves in Fig. 7. Based on these considerations, we conclude that the information of the phase transition is again encoded in the fully-connected weight because $G(K)$ in Eq. (23) is directly related to the free energy. Particularly, the critical temperature is obtained by detecting the enhancement of the second derivative of weights with respect to temperature.

It is noted that the model or the weight parametrization demonstrated above is one of many possibilities yielding the same temperature prediction. For example, the quadratic K-

15

dependence could arise from the biases on the last layer instead of the weights [32]. In that case, the information of phase transition or free energy should be encoded in the bias. How the physical information is stored in the machine parameters depends on the architecture of neural network including the number of hidden units or activation functions.

Having gained the insight on the internal structure of the neural networks, we argue on why the convolutional neural network works better than the one composed only of fully-connected layers in predicting temperatures from physical viewpoint. As we have observed in this section, the latter captures the magnetization in the intermediate layer, while the former extracts the internal energy, based on which it tries to infer the temperature. However, the temperature-dependence of magnetization in the Ising model is small except around the critical temperature, compared with that of internal energy. Thus, it is plausible that the neural network extracts information of the internal energy for more accurate temperature prediction. Indeed, our convolutional neural network successfully learns the internal energy to achieve higher accuracy. The better performance by the convolutional neural network should be readily understood. It is due to the fact that the convolutional layers identify the feature of input spin configurations by exploiting their spatial structure such as spatial correlation of Ising spins. The fully-connected layers, however, are blind to the structure.

Finally, we mention that the discussion given for 2D Ising model also holds for the 3-state Potts model.

IV. CONCLUSION

We revisited the phase transition detection of the 2D Ising model based on temperature-supervised machine learning to clarify the underlying mechanism. ...

We first demonstrated that the fully-connected neural network shows the drastic change in its weight structure of the second layer as a result of training on 2D Ising model as well as 3-state Potts model. Closer look at the neural network with 3 hidden units revealed that it actually captures the magnetization in the first layer. The phase transition detection is, however, a consequence of low prediction accuracy of temperature on input configurations except around the critical temperature.

On the other hand, employing the convolutional neural network, we succeeded in im-

proving the temperature-prediction accuracy. It turned out that the trained convolutional network captures the internal energy of input configurations instead of magnetization. Also, the weights in the last fully-connected layer do not show any drastic change in their value in contrast to the former case and this aspect is understood from the viewpoint of optimal temperature prediction. In this case, the weights are proportional to free energy, and hence, the physical information including the critical temperature is again encoded in them.

Interestingly, the trained neural network extract different physical information depending on how they are trained. A "bad" neural network in terms of temperature prediction tries to capture the magnetization of input spin configurations, which happens to be convenient for detecting critical temperature as it is the order parameter. Improvement of the network architecture allow us to construct a "good (better)" temperature predictor. But the information of phase transition is encoded in the network more implicitly.

Appendix A: Temperature prediction and its accuracy

We think about how temperature prediction of Ising spin configuration works [32]. We start with preparing Ising spin configurations generated by Markov-chain Monte-Carlo algorithm at target temperatures. The configurations at a fixed temperature K are distributed over energy with mean $\langle E \rangle$[3] and variance $\langle E^2 - \langle E \rangle^2 \rangle$, each of which is given by

$$\langle E \rangle = \frac{\sum_{\{\sigma_i\}} H(\{\sigma_i\}) e^{-KH(\{\sigma_i\})}}{\sum_{\{\sigma_i\}} e^{-KH(\{\sigma_i\})}} = -\frac{\partial}{\partial K} \ln \mathcal{Z}, \tag{A1}$$

$$\langle E^2 - \langle E \rangle^2 \rangle = -\frac{\partial}{\partial K} \langle E \rangle = \frac{\partial^2}{\partial K^2} \ln \mathcal{Z}. \tag{A2}$$

Given a spin configuration, we consider the optimal prediction of temperature. Energy can be calculated for each configuration. It is noted that, since the standard deviation of

---

[3] $\langle E \rangle$ stands for a thermal expectation value of the Hamiltonian's eigenvalues.
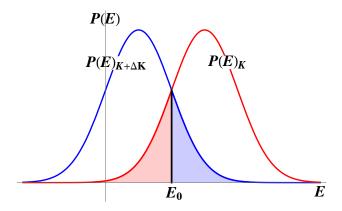
Figure 8. Energy probability distributions at two nearby temperatures K and $K+\Delta K$, which have overlap shown by red and blue shaded areas. The red distribution is generated at a temperature K. The configurations are, however, misclassified as those at temperature $K + \Delta K$ by maximum-like method if they are in the red shaded area because $P(E)_{K+\Delta K} > P(E)_K$ below E_0. The same argument holds for configurations generated in the blue shaded area.

energy density E/L^2 is proportional to L^{-1}, the energy probability distribution does not admit width in infinite system. In such case, provided a certain configuration, we should be able to correctly predict the temperature at which it is generated by calculating its energy density. However, the distribution at each temperature has finite width if we consider finite systems. In that case, temperature prediction does not necessarily give the correct answer because there are overlaps between energy distributions (Fig. 8). The best we can do is to guess the temperature of the configuration by maximally likelihood estimate, namely, the temperature that is most likely to yield the configuration's energy is the optimally predicted temperature. Consequently, there is an upper bound in the accuracy of prediction from the overlap of the energy probability distributions in finite system.

Let us take a look at two dimensional Ising model in more detail. Since it is exactly solved [28], we have an explicit expression of the free energy density,

$$f = -\frac{\ln \mathcal{Z}}{N} = -\frac{\ln 2}{2} - \ln[\cosh(2K)] - \frac{1}{2\pi} \int_0^\pi d\theta \ln \left[1 + \sqrt{1 - \left(\frac{2\sinh(2K)}{\cosh^2(2K)} \cos \theta \right)^2} \right],$$
(A3)

with the number of sites N. We do not incorporating the finite size effect. Then, we obtain the energy probability distributions approximated by the Gaussian distributions with mean $\langle E \rangle$ and variance $\langle (E - \langle E \rangle)^2 \rangle$ at the temperatures of our interest. We show energy
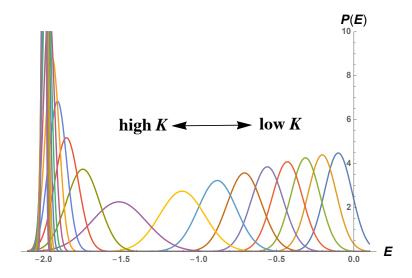
Figure 9. Energy probability distributions in two dimensional Ising model at temperatures $K = 0.05, 0.1, 0.15, \ldots, 1.0$. The vertical axis is energy per site and number of sites is taken to be 16×16.

distributions at 20 different temperatures $K = 0.05, 0.1, 0.15, \ldots, 1.0$ in Fig. 9. We see that the overlaps between distributions are significant while they are reduced around the critical temperature $K_{\mathrm{c}}^{\mathrm{exact}} \sim 0.4407$. The accuracy obtained by maximum-likelihood estimate is as low as 40.1%, implying that a thermometer constructed by machine learning can achieve an accuracy 40.1% at most. Nevertheless, it turns out that the trained neural network can work as a phase transition detector.

[1] L. D. Landau, *On the theory of phase transitions. I.*, *Zh. Eksp. Teor. Fiz.* **11** (1937) 19.

[2] V. L. Ginzburg and L. D. Landau, *On the theory of superconductivity*, *Zh. eksp. teor. Fiz* **20** (1950) 35.

[3] J. Carrasquilla and R. G. Melko, *Machine learning phases of matter*, *Nature Physics* **13** (2017) 431.

[4] A. Tanaka and A. Tomiya, *Detection of phase transition via convolutional neural network*, *J. Phys. Soc. Jap.* **86** (2017) 063001.

[5] T. Ohtsuki and T. Ohtsuki, *Deep learning the quantum phase transitions in random two-dimensional electron systems*, *Journal of the Physical Society of Japan* **85** (2016) 123706.

[6] T. Mano and T. Ohtsuki, *Phase Diagrams of Three-Dimensional Anderson and Quantum*

Percolation Models Using Deep Three-Dimensional Convolutional Neural Network, *Journal of the Physical Society of Japan* **86** (2017) 113704.

[7] F. Schindler, N. Regnault and T. Neupert, *Probing many-body localization with neural networks*, *Physical Review B* **95** (2017) 245134.

[8] P. Broecker, J. Carrasquilla, R. G. Melko and S. Trebst, *Machine learning quantum phases of matter beyond the fermion sign problem*, *Scientific reports* **7** (2017) 8823.

[9] K. Ch'ng, J. Carrasquilla, R. G. Melko and E. Khatami, *Machine learning phases of strongly correlated fermions*, *Physical Review X* **7** (2017) 031038.

[10] Y. Zhang and E.-A. Kim, *Quantum loop topography for machine learning*, *Physical review letters* **118** (2017) 216401.

[11] P. Ponte and R. G. Melko, *Kernel methods for interpretable machine learning of order parameters*, *Physical Review B* **96** (2017) 205146.

[12] Y. Zhang, R. G. Melko and E.-A. Kim, *Machine learning $\mathbb{Z}_2$ quantum spin liquids with quasiparticle statistics*, *Physical Review B* **96** (2017) 245119.

[13] S. Arai, M. Ohzeki and K. Tanaka, *Deep Neural Network Detects Quantum Phase Transition*, *Journal of the Physical Society of Japan* **87** (2018) 033001.

[14] M. J. S. Beach, A. Golubeva and R. G. Melko, *Machine learning vortices at the Kosterlitz-Thouless transition*, *Phys. Rev. B* **97** (Jan, 2018) 045207.

[15] S. J. Wetzel and M. Scherzer, *Machine Learning of Explicit Order Parameters: From the Ising Model to SU(2) Lattice Gauge Theory*, *Phys. Rev.* **B96** (2017) 184410.

[16] M. Richter-Laskowska, H. Khan, N. Trivedi and M. Maśka, *A machine learning approach to the Berezinskii-Kosterlitz-Thouless transition in classical and quantum models*, `arXiv:1809.09927`.

[17] P. Suchsland and S. Wessel, *Parameter diagnostics of phases and phase transition learning by neural networks*, *Phys. Rev. B* **97** (2018) 174435.

[18] D. Kim and D.-H. Kim, *Smallest neural network to learn the ising criticality*, *Phys. Rev. E* **98** (Aug, 2018) 022138.

[19] L. Wang, *Discovering phase transitions with unsupervised learning*, *Physical Review B* **94** (2016) 195105.

[20] A. Morningstar and R. G. Melko, *Deep learning the Ising model near criticality*, `arXiv:1708.04622`.

[21] E. P. Van Nieuwenburg, Y.-H. Liu and S. D. Huber, *Learning phase transitions by confusion*, *Nature Physics* **13** (2017) 435.

[22] W. Hu, R. R. Singh and R. T. Scalettar, *Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination*, *Physical Review E* **95** (2017) 062122.

[23] S. J. Wetzel, *Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders*, *Physical Review E* **96** (2017) 022140.

[24] S. Iso, S. Shiba and S. Yokoo, *Scale-invariant Feature Extraction of Neural Network and Renormalization Group Flow*, *Phys. Rev.* **E97** (2018) 053304.

[25] P. Zhang, H. Shen and H. Zhai, *Machine learning topological invariants with neural networks*, *Phys. Rev. Lett.* **120** (2018) 066401.

[26] N. Sun, J. Yi, P. Zhang, H. Shen and H. Zhai, *Deep learning topological invariants of band insulators*, *Phys. Rev. B* **98** (2018) 085402.

[27] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, `arXiv:1412.6980`.

[28] L. Onsager, *Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition*, *Phys. Rev.* **65** (1944) 117–149.

[29] F. Y. Wu, *The potts model*, *Rev. Mod. Phys.* **54** (1982) 235–268.

[30] R. J. Baxter, *Potts model at the critical temperature*, *Journal of Physics C: Solid State Physics* **6** (1973) L445.

[31] S. Alexander and D. J. Amit, *When the Landau criterion fails qualitatively*, *Journal of Physics A: Mathematical and General* **8** (1975) 1988.

[32] K. Aoki, T. Fujita and T. Kobayashi, *What does deep learning of statistical system learn?*, *Journal of the Japanese Society for Artificial Intelligence* **33** (2018) .