Journal of the Physical Society of Japan **86**, 063001 (2017) **Letters**
https://doi.org/10.7566/JPSJ.86.063001

Detection of Phase Transition via Convolutional Neural Networks

Akinori Tanaka[1*] and Akio Tomiya[2†]

[1]*Interdisciplinary Mathematical and Computational Collaboration Team, RIKEN, Wako, Saitama 351-0198, Japan*
[2]*Key Laboratory of Quark & Lepton Physics (MOE) and Institute of Particle Physics, Central China Normal University, Wuhan 430079, China*

(Received March 25, 2017; accepted April 10, 2017; published online April 28, 2017)

A convolutional neural network (CNN) is designed to study correlation between the temperature and the spin configuration of the two-dimensional Ising model. Our CNN is able to find the characteristic feature of the phase transition without prior knowledge. Also a novel order parameter on the basis of the CNN is introduced to identify the location of the critical temperature; the result is found to be consistent with the exact value.

Studies of phase transition are connected to various areas among theoretical/experimental physics.[1–7] Calculating order parameters is one of the conventional ways to define phases and phase transitions. However, some phases like topological phases[8] do not have any clear order parameters. Even if there are certain theoretical order parameters like entanglement entropy,[9,10] they are difficult to measure in experiments.

Machine learning (ML) techniques are useful to resolve this undesirable situation. In fact, ML techniques have been already applied to various problems in theoretical physics: finding approximate potential surface,[11] a study of transition in glassy liquids,[12] solving mean-field equations[13] and quantum many-body systems,[14,15] a study of topological phases.[16]

Especially, ML techniques based on convolutional neural network (CNN) have been developing since the recent groundbreaking record[17] in ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012),[18] and it is applied to investigate phases of matters with great successes on classifications of phases in 2D systems[19–21] and 3D systems.[22,23] It is even possible to draw phase diagrams.[23]

In these previous works, however, one needs some informations of the *answers* for the problems a priori. For example, to classify phases of a system, the training process requires the values of critical temperatures or the location of phase boundaries. This fact prevents applications of the ML techniques to unknown systems so far.

The learning process without any answers is called *unsupervised learning*. Indeed, there are known results on detecting the phase transitions based on typical unsupervised learning architectures called autoencoder which is equivalent to principal component analysis[24] and its variant called variational autoencoder.[25] These architectures encode informations of given samples to lower dimensional vectors, and it is pointed out that such encoding process is similar to encoding physical state informations to order parameters of the systems. However, it is not evident whether the latent variables provide the critical temperature.

We propose a novel but simple prescription to estimate the critical temperature of the system via neural network (NN) based on ML techniques without a priori knowledge of the order parameter. Throughout this letter, we focus on the ferromagnetic 2D Ising model on a square lattice mainly based on the following three reasons. First, this system is one of the simplest solvable systems[26,27] which has a phase transition, and it is easy to check the validity of our method. Second, this system can be generalized to other classical spin systems like Potts model, XY model and Heisenberg model, so our method can be applied to these other systems straightforwardly. Third, this model is a good benchmark for new computational methods.[28,29]

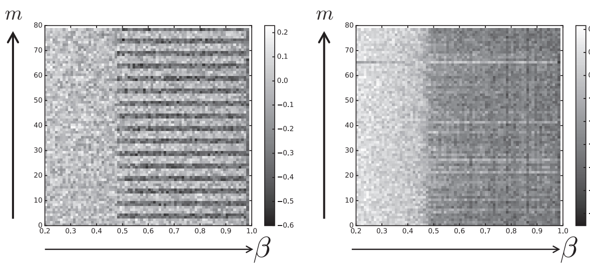

Fig. 1. Plots of weight matrix components in convolutional neural network (left) and fully connected neural network (right). Horizontal axis corresponds to the inverse temperature β. Vertical axis m corresponds to components connected to hidden nodes in NN.

Table I. Critical temperature of 2D Ising model on the square lattice extracted from CNN (3 from top). $L \to \infty$ stands for thermodynamic limit, and its value is exact one, $\beta_c^{\text{Exact}} = \frac{1}{2}\log(\sqrt{2}+1)$.

System size	β_c (CNN)	β_c (FC)
8×8	0.478915	0.462494
16×16	0.448562	0.433915
32×32	0.451887	0.415596
$L \to \infty$	$\beta_c^{\text{Exact}} \sim 0.440686$	

Using TensorFlow (r0.9),[30] we have implemented a neural network to study the correlation between spin configurations and discretized inverse temperatures. We find that NNs are able to capture features of phase transition, even for simpler fully-connected (FC) NN, in the weight W without any information of order parameters nor critical temperature. Figure 1 reminds us of the discovery of the "cat cell" in the literature[31] in which the model recognizes images of cats without having explicitly learned what a cat is.

We examine the boundary structure in Fig. 1 by defining order parameter, and estimate the inverse critical temperature by fitting distribution of the order parameter (Table I).

First, we explain the details of our NNs. If the reader is not familiar with machine learning based on NN, we suggest reading literatures reviewing it, for example, Refs. 32 and 33. Our NN model is designed to solve classification problems. It is constructed from a convolution layer and a fully connected

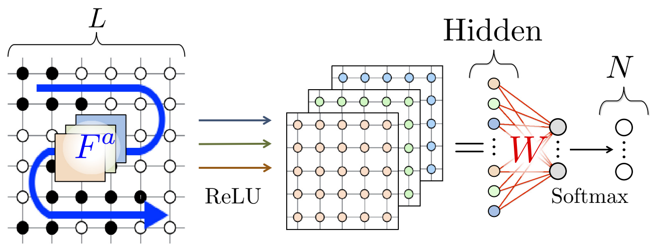

Fig. 2. (Color online) A schematic explanation for the network (1). In this figure, we take three filters F^a $(a = 1, 2, 3)$. An Ising spin configuration $\{\sigma_{xy}\}$ generated by MCMC is passed to three hidden lattices via the convolution process + ReLU activation. After making them flatten, they are connected to N nodes via fully-connected weight W. In the end, it returns $\beta_I^{\rm CNN}$ with the softmax activation.

layer, so it can be regarded as the simplest model of a CNN (Fig. 2). The whole definition is as follows.

$$\left[\begin{array}{l} \mathcal{I} = \{\{\sigma_{xy}\}|\text{Ising config on } L\times L \text{ lattice.}\} \\ \downarrow \left\{\begin{array}{l}\text{Convolution}_{[N_f^2\text{-filter, }(s,s)\text{-stride, }C\text{-channels}]} \\ \text{ReLU activation} \\ \text{Flatten}\end{array}\right. \\ \mathbb{R}^{\text{Hidden}=L^2/s^2\times C} \\ \downarrow \left\{\begin{array}{l}\text{Fully connected} \\ \text{Softmax}\end{array}\right. \\ [0,1]^N = \mathcal{O} \end{array}\right] \tag{1}$$

The first transformation in (1) is defined by *convolution* including training parameters F^a_{ij} called *filters*, rectified linear activation called *ReLU*, and flatten procedure:

$$\begin{aligned} &\{\sigma_{xy}\} \\ &\overset{\text{conv}}{\to} \sum_{i,j=1}^{N_f} \sigma_{(sX+i)(sY+j)} F^a_{ij} = \Sigma^a_{XY} \\ &\overset{\text{ReLU}}{\to} \max(0, \Sigma^a_{XY}) = u^a_{XY} \\ &\overset{\text{flatten}}{\to} \vec{u} = [u^1_{11}, u^2_{11}, \dots u^C_{11}, u^1_{21}, u^2_{21}, \dots, u^C_{21}, \dots] = [u_m]. \end{aligned} \tag{2}$$

The second transformation is defined by fully-connected layer including training parameters W^m_I called *weights*, and softmax activation:

$$\begin{aligned} &[u_m] \\ &\overset{\text{fully-connected}}{\to} \sum_{m=1}^{L^2/s^2\times C} W^m_I u_m = z_I \\ &\overset{\text{Softmax}}{\to} \frac{e^{z_I}}{\sum_{J=1}^{N} e^{z_J}} = \beta_I^{\rm CNN}. \end{aligned} \tag{3}$$

We classify configurations into N classes labeled by I in the final step. This N is related to the inverse temperature β through (7). Because of the following facts, $\beta_I^{\rm CNN} \in [0,1]$ and $\sum_I \beta_I^{\rm CNN} = 1$, we can interpret $\beta_I^{\rm CNN}$ as a probability for classifying given state $\{\sigma_{xy}\}$ to I-th class in a classification problem. In total, we have two types of parameters:

$$\begin{array}{cc} F^a_{ij} \text{ in convolution,} & W^m_I \text{ in fully connected layer.} \\ \left[\begin{array}{l} a = 1,\dots,C \\ i,j = 1,\dots,N_f \end{array}\right] & \left[\begin{array}{l} m = 1,\dots,L^2/s^2\times C \\ I = 1,\dots N \end{array}\right] \end{array} \tag{4}$$

In later experiments, these parameters will be updated. In the first trial, we take NN without convolution (12). In this case, the parameters F are not the filters in (4) but weights.

We need training set for optimizing the above parameters (4) in CNN. We call it $\mathcal{T}_L$ where L indicates the size of the square lattice. The definition is

$$\mathcal{T}_L = \left\{(\{\sigma^{(n)}_{xy}\}, \vec{\beta}_n) | \frac{1}{\beta_n} = T_{\min} + n\delta\right\}_{n=0,\dots,(N_{\rm conf}-1)}, \tag{5}$$

where $\{\sigma^{(n)}_{xy}\}$ is the generated configuration under the Ising Hamiltonian on the square lattice

$$H = -\sum_{x,y} \sigma_{xy}(\sigma_{(x+1),y} + \sigma_{x,(y+1)}), \tag{6}$$

and inverse temperature β_n using the Metropolis method.

$T_{\min\,(\max)}$ is the minimum (maximum) temperature for the target Ising system. The temperature resolution is defined by $\delta = (T_{\max} - T_{\min})/N_{\rm conf}$ where $N_{\rm conf}$ is the number of samples. $\vec{\beta}$ is the discretized inverse temperature defined by

$$\text{cl}_N : \beta \to \vec{\beta} = \begin{cases} (1,0,\dots,0,0) & \text{for } \beta < 0 \\ (0,1,\dots,0,0) & \text{for } \beta \in \left[0, \frac{1}{N-2}\right) \\ \cdots & \\ (0,0,\dots,1,0) & \text{for } \beta \in \left[\frac{N-3}{N-2}, 1\right) \\ (0,0,\dots,0,1) & \text{for } 1 \le \beta \end{cases}. \tag{7}$$

This is called the one-hot representation, and enables us to implement the inverse temperature β into the NN directly. We use index I or J to represent component of the vector $\vec{\beta}$ as already used in.

Now let us denote our CNN, explained in (1) as $F^{(F,W)}_{\rm CNN}$. We need *error function* that measures the difference between the output of the CNN and the correct discretized inverse temperature. In our case, the task is classification, so we take cross entropy as the error function:

$$E(\vec{\beta}^{\rm CNN}, \vec{\beta}) = -\sum_{I=1}^{N} \beta_I \log \beta_I^{\rm CNN}, \tag{8}$$

where $\vec{\beta}^{\rm CNN} = F^{(F,W)}_{\rm CNN}(\{\sigma_{xy}\})$ and $(\{\sigma_{xy}\}, \vec{\beta}) \in \mathcal{T}_L$.

Roughly speaking, the parameters $w = (F^a_{ij}, W^m_I)$ are updated via $w \leftarrow w - \epsilon\nabla_w E$ with small parameter ϵ. More precisely speaking, we adopt a sophisticated version of this method called *Adam*[34] implemented in TensorFlow (r0.9)[30] to achieve better convergence.

Our neural network, $F^{(F,W)}_{\rm CNN}$, learns the optimal parameters $F = \{F^a_{ij}\}$ and $W = \{W^m_I\}$ in (4) through iterating the optimization of the cross entropy (8) between the answer $\vec{\beta}$ and the output $\vec{\beta}_{\rm CNN}$ constructed from stochastically chosen data $(\{\sigma_{xy}\}, \vec{\beta}) \in \mathcal{T}_L$.

```
Algorithm
Require: CNN $F^{(F,W)}_{\rm CNN}$ (1); an Ising dataset $\mathcal{T}_L$ (5)
for Num_of_iterations do
    Choose $(\{\sigma_{xy}\}, \vec{\beta}) \in \mathcal{T}_L$ randomly
    $\vec{\beta}^{\rm CNN} = F^{(F,W)}_{\rm CNN}(\{\sigma_{xy}\})$                    (9)
    loss = $E(\vec{\beta}^{\rm CNN}, \vec{\beta})$ by (8)
    Update $F, W$ via AdamOptimizer (loss)
end for
```

As we show later, the weight matrix W inheres well approximated critical temperature after 10,000 iterations in (9).

Here, we prepare $\mathcal{T}_L$ (5) by using the Metropolis method with the parameters

$$T_{\min} = 0.1, \quad T_{\max} = 5.0, \quad N_{\text{conf}} = 10^4. \tag{10}$$

The max and min values for T mean that $0.2 < \beta < 10$, where β is the inverse temperature. Note that the known value form the phase transition is $T_c \sim 2.27$ or $\beta_c \sim 0.44$. This means that configurations in our training data $\mathcal{T}_L$ extend from the ordered phase to the disordered phase. In all, we prepare three types of training set,

$$\mathcal{T}_{L=8}, \quad \mathcal{T}_{L=16}, \quad \mathcal{T}_{L=32}. \tag{11}$$

We apply negative magnetic field weakly to realize the unique ground state at zero temperature. As a result, almost all configurations at low temperature ($\beta \gg \beta_c \sim 0.44$) are ordered to $\{\sigma_{xy}\} = \{-1, -1, \dots, -1\}$.

Before showing the CNN result, let us try a somewhat primitive experiment: training on NN without the convolution layer, i.e., a fully connected NN.

$$\left[\begin{array}{l} \mathcal{I} = \{\sigma_{xy} | \text{Ising config on } L \times L \text{ lattice.}\} \\ \downarrow \left\{\begin{array}{l} \text{Flatten} \\ \text{Fully connected } F \\ \text{Softmax} \end{array}\right. \\ [0,1]^{\text{Hidden}=80} \\ \downarrow \left\{\begin{array}{l} \text{Fully connected } W \\ \text{Softmax} \end{array}\right. \\ [0,1]^{N=100} = \mathcal{O} \end{array}\right] \tag{12}$$

We retain the error function and optimizer, i.e., the cross entropy (8) and AdamOptimizer (10^{-4}). We align the heat map for the weight W trained by using $\mathcal{T}_{L=8}$ in right side of Fig. 1. After 10,000 iterations, NN detected the phase transition. So this NN is sufficient for detecting ising phase transition, but we cannot answer *why* this NN captures it. To answer it, we turn to our main target: CNN below.

Next, we take $N = 100$, $N_f = 3$ and $C = 1$ and use the $\mathcal{T}_{L=8}$ training set. Once we increase the number of iterations to 10,000, we get two possible ordered figures. We denote them case (A) and case (B) respectively as shown in Fig. 3.

Case (A) is characterized by $\sum_{ij} F_{ij} > 0$, and we can observe two qualitatively different regions in the heat map of the weight W, black colored region ($0.48 \lesssim \beta$) and gray colored region ($\beta \lesssim 0.48$). The boundary is close to the critical temperature $\beta_c \sim 0.44$. Case (B) is characterized by $\sum_{ij} F_{ij} < 0$, and values in the heat map for W are in gray colored region and almost homogeneous. We will discuss later the reason why only case (A) displays phase transition.

We now turn to the multi-filter case with $N = 100$, $N_f = 3$, $C = 5$, and $L = 8$. The results for all heat maps after 10,000 iterations are shown in Fig. 4. The stripe structure in the heat map of W corresponds to its values connecting to the convoluted and flatten nodes via five filters, (A), (A), (B), (B), (B) respectively. Empirically speaking, the number of filters should be large to detect the phase transition because the probability for appearance of (A) increases with increased statistics.

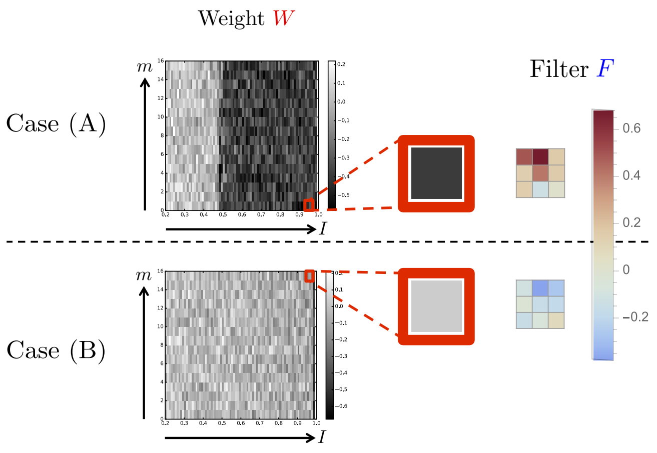

Fig. 3. (Color online) Heat maps of W_I^m and F_{ij} for the CNN with one filter. In case (A), there always exist two distinct regions (black and gray). In case (B), there is no such clear decomposition.

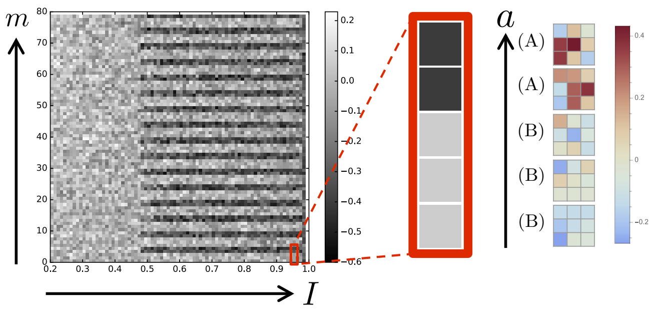

Fig. 4. (Color online) Heat maps of W_I^m and F_{ij}^a with five filters.

From the experiments, we know that our model (1) seems to discover the phase transition in the given training data for the Ising model. In order to verify this statement, we would like to extract the critical temperature from our CNN after the training. As a trial, we fix the parameters of the CNN as follows: the number of filters, channel and stride are $N_f = 3$, $C = 5$, and $s = L/4$ respectively. The number of classifications is taken as $N = 100$ as well as we did in previous section.

First, we plot heat maps for the weight matrix W in CNN trained by $\mathcal{T}_{L=8}$, $\mathcal{T}_{L=16}$, $\mathcal{T}_{L=32}$. For every lattice size, we observe a domain-like structure with a boundary around $\beta \sim 0.44$. However, the heat map does not give the location of the boundary quantitatively. We propose an order parameter based on the weight matrix W_I^m:

$$W_{\text{sum}}(\beta) = \sum_m W_I^m \text{cl}_N^I(\beta), \tag{13}$$

and estimate the critical temperature. The result is shown in Fig. 5. To quantify the boundary in the heat map for W, we define critical temperature extracted by CNN β_{CNN} by fitting $W_{\text{sum}}(\beta)$ with the following function: $\tilde{W}_{\text{sum}}(\beta) = a \tanh[c(\beta - \beta_{\text{CNN}})] - b$, where a, b, c, and β_{CNN} are fitting variables and β_{CNN} indicates the location of the jump. This function is motivated by the magnetization of the Ising model using the mean field approximation. Table I shows the fit results both for CNN and FC. Our results show that β_{CNN} matches the critical temperature to 2–8% accuracy. Compared to it, β_{FC} shows less accuracy.

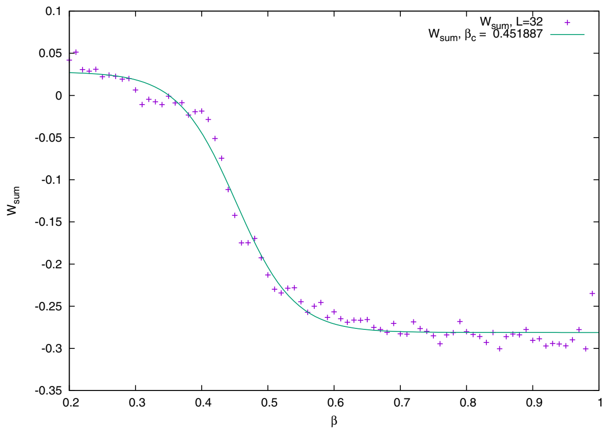

Fig. 5. (Color online) $W_{\rm sum}(\beta)$ for $L = 32$. The horizontal axis is the inverse temperature β which is translated using (7). We fit $W_{\rm sum}(\beta)$ data by the following smooth function: $a \tanh[c(\beta - \beta_{\rm CNN})] - b$ (Green curve), where $a, b, c, \beta_{\rm CNN}$ are fitting parameters, and regard the determined fitting parameter $\beta_{\rm CNN}$ as a critical temperature determined by CNN.

Let us conclude this letter. We have designed simple neural networks to study correlation between configuration of the 2D Ising model and inverse temperature, and we have trained them by SGD method implemented by TensorFlow. We have found that the weight W in neural networks captures a feature of phase transition of the 2D Ising model, and defined a new order parameter $W_{\rm sum}(\beta)$ in (13) via trained neural networks and have found that it can provide the value of critical inverse temperature.

Why are our neural networks able to find a feature of phase transition? There is an intuitive explanation thanks to CNN experiments. The filter with $N_f = 3$ in case (A) has a typical average around 0.1–0.2. This is close to the convolution with filter $F_{ij} = 1/N_f^2$ which is equivalent to a real space renormalization group transformation, and the filters reflect local magnetization which is related to a typical order parameter and it enables CNN to detect the phase transition. As an analog of this, FC NN might realize the real space renormalization group transformation in inside.

Our NN model has potential to investigate other statistical models. For example, it was reported that CNNs can distinguish phases of matters, topological phases in $\mathbb{Z}_2$ gauge theories,[19] phases in the Hubbard model[22] and Potts model.[35] It is interesting to apply our design of neural networks to these problems and see whether the NN can discover nontrivial phases automatically, as we did in this letter.

Acknowledgments We would like to thank to K. Doya, K. Hashimoto, T. Hatsuda, Y. Hidaka, M. Hongo, B. H. Kim, J. Miller, S. Nagataki, N. Ogawa, M. Taki, and Y. Yokokura for constructive comments and warm encouragement. We also thank to D. Zaslavsky for careful reading this manuscript. The work of Akinori Tanaka was supported in part by the RIKEN iTHES Project. The work of Akio Tomiya was supported in part by NSFC under Grant No. 11535012.

[*]akinori.tanaka@riken.jp
[†]akio.tomiya@mail.ccnu.edu.cn

1) K. G. Wilson and J. Kogut, Phys. Rep. **12**, 75 (1974).
2) J. Polchinski, arXiv:hep-th/9210046.
3) K. Intriligator and N. Seiberg, Nucl. Phys. B (Proc. Suppl.) **45**, 1 (1996).
4) R. Pasechnik and M. Šumbera, Universe **3**, 7 (2017).
5) P. Hohenberg and A. Krekhov, Phys. Rep. **572**, 1 (2015).
6) Q. Chen, J. Stajic, S. Tan, and K. Levin, Phys. Rep. **412**, 1 (2005).
7) B. Kramer, T. Ohtsuki, and S. Kettemann, Phys. Rep. **417**, 211 (2005).
8) X. G. Wen, Int. J. Mod. Phys. B **04**, 239 (1990).
9) A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).
10) M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).
11) J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
12) S. S. Schoenholz, E. D. Cubuk, D. M. Sussman, E. Kaxiras, and A. J. Liu, Nat. Phys. **12**, 469 (2016).
13) L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.
14) L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).
15) G. Carleo and M. Troyer, Science **355**, 602 (2017).
16) D.-L. Deng, X. Li, and S. D. Sarma, arXiv:1609.09060.
17) A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems*, ed. M. I. Jordan (MIT Press, Cambridge, MA, 2012) p. 1097.
18) O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, Int. J. Comput. Vis. **115**, 211 (2015).
19) J. Carrasquilla and R. G. Melko, Nat. Phys. (in press) [DOI: 10.1038/nphys4035].
20) P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, arXiv:1608.07848.
21) T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).
22) K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, arXiv:1609.02552.
23) T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **86**, 044708 (2017).
24) L. Wang, Phys. Rev. B **94**, 195105 (2016).
25) S. J. Wetzel, arXiv:1703.02435.
26) L. Onsager, Phys. Rev. **65**, 117 (1944).
27) Y. Nambu, *Broken Symmetry: Selected Papers of Y Nambu* (World Scientific, Singapore, 1995) World Scientific Series in 20th Century Physics, Vol. 13, p. 1.
28) P. Mehta and D. J. Schwab, arXiv:1410.3831.
29) K.-I. Aoki, T. Kobayashi, and H. Tomita, Int. J. Mod. Phys. B **23**, 3739 (2009).
30) M. Abadi et al., arXiv:1603.04467.
31) Q. V. Le, Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, 2013, p. 8595.
32) H. Kolanoski, *Application of Artificial Neural Networks in Particle Physics* (Springer, Heidelberg, 1996) International Conference on Artificial Neural Networks, p. 1.
33) Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
34) D. Kingma and J. Ba, arXiv:1412.6980.
35) C.-D. Li, D.-R. Tan, and F.-J. Jiang, arXiv:1703.02369.