Smallest Neural Network to Learn the Ising Criticality

Dongkyu Kim and Dong-Hee Kim*
Department of Physics and Photon Science, School of Physics and Chemistry, Gwangju Institute of Science and Technology, Gwangju 61005, Korea

Learning with an artificial neural network encodes the system behavior in a feed-forward function with a number of parameters optimized by data-driven training. An open question is whether one can minimize the network complexity without loss of performance to reveal how and why it works. Here we investigate the learning of the phase transition in the Ising model and find that having two hidden neurons can be enough for an accurate prediction of critical temperature. We show that the networks learn the scaling dimension of the order parameter while being trained as a phase classifier, explaining why the trained network works universally for different lattices of the same criticality.

Machine learning [1, 2] builds a data-driven predictive model of responding a given query, which drastically differs from the conventional approach based on a characterization of the system in question. Performing as a classifier, an artificial neural network can suggest a proper label for an unacquainted input without doing explicit analysis, which is done by training a large set of the network parameters to adapt themselves to already labeled data. In spite of many empirical successes, one intrinsic issue is that the network often works like a "black box" since it is generally difficult to see inside how it reaches at a particular output. Such lack of transparency is due to the high complexity coming out of the interplay between many network parameters. The more complex structure may help increasing flexibility in learning but at the same time makes it harder to understand how it extracts a desired feature from the data. In this paper, we present the opposite extreme of a minimally simple neural network to explain the observed accuracy and universality in its learning of the phase transition in the Ising model.

The ideas of machine learning have been actively applied to problems in classical and quantum physics. For instance, efficient Monte Carlo simulation methods were proposed by integrating machine learning into wavefunction representations [3–10] and cluster updates [11–16]. On the other hand, phase transitions have been extensively examined in various schemes of the supervised [17–31] and unsupervised [32–40] learning to classify phases, capture topological features, and locate transition points. In particular, the seminar work by Carrasquilla and Melko [17] demonstrated that the neural network trained to classify the phases of the Ising model in the square lattices can actually predict the critical temperature even when applied to the triangular lattices. Remarkably, the network outputs for different system sizes seemingly fell on the same curve in the finite-size-scaling tests.

We explain these behaviors by solving an analytically tractable model of the neural network that we devise to capture a typical structure emerging in the training of large-scale networks. While the number of hidden neurons is reduced to two in our network, the accuracy of locating the critical temperature is comparable to the previous result with 100 neurons [17]. It turns out that the information explicitly encoded in the network is the scaling dimension of the order parameter, indicating the interoperability within the class of the same criticality.

Let us first show the structure that we observe in the network trained in the square lattices (see Fig. 1). We consider a typical fully-connected feedforward network with a single hidden layer of 50 neurons between input and output where the sigmoid function normalizes the activation signals. The network is trained by assigning zero (one) to the desired output for the disordered (ordered)

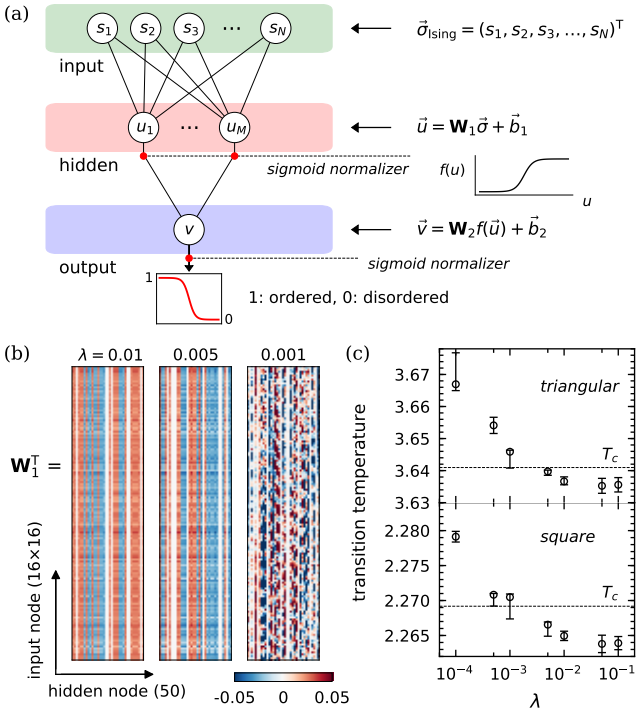

FIG. 1. Neural network as a phase classifier for the Ising model. (a) The schematic diagram of the signal processing. (b) The examples of the weight matrix $\mathbf{W}_1$ at different values of the regularization strength λ. (c) The transition temperature as a function of λ predicted by the 50-unit networks which are trained in the square lattices and then applied to the square and triangular lattices for interoperability tests.

phase based on the labeled dataset of 1800 spin configurations per temperature which are sampled from the Monte Carlo simulations [3] with the Ising Hamiltonian $\mathcal{H} = -\sum_{\langle i,j\rangle} s_i s_j$, where $s_i \in \{1, -1\}$ is the spin state at site i, and $\langle i, j\rangle$ runs over the nearest neighbors. The training dataset includes 229 temperatures distributed with the step size 0.01 in $(0.5T_c, 1.5T_c)$ around the exact critical temperature $T_c = 2/\ln(1+\sqrt{2})$. We use TensorFlow [2] to minimize the cross entropy with the L_2 regularization to avoid overfitting [43].

Two features are notable from the link weights between the input and hidden layers as exemplified in Fig. 1(b). First, a hidden neuron tends to receive a signal accumulated with almost constant weights of either sign, suggesting that the input $\{s_i\}$ is reduced into its sum $\propto \pm\sum_i s_i$. This is consistent with the activation patterns of the hidden neurons observed previously [17, 30] and the concept of the toy model [17]. Second, there are neurons found effectively unlinked with vanishing weights, implying that the size of the hidden layer can be even smaller.

These features are robust at the regularization strength $\lambda > 0.001$ for all system sizes examined. The structure partly survives at $\lambda = 0.001$, while it fades away as λ gets weaker [43]. We find that the prediction of the transition temperature is consistent at $\lambda > 0.001$ for the networks that are trained in the square lattices and examined also in the triangular lattices (see Fig. 1(c)). In contrast, as λ gets weaker, the accuracy becomes inconsistent in this interoperability test between the different lattices.

Inspired from these observations, we propose a minimal network model by having only two neurons in the hidden layer. The one is linked from the input with a positive constant weight, reading $y \propto \sum_i s_i$, while the other is associated with the opposite sign, reading $-y$. We need a pair of them to learn the $\mathbb{Z}(2)$ symmetry of the Ising model, which is in contrast to the previous toy model [17]. Thus, we write the weight matrix $\mathbf{W}_1$ for the links and the bias vector $\vec{b}_1$ for the hidden neurons as

$$\mathbf{W}_1 = \frac{1}{N}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \end{pmatrix}, \quad \vec{b}_1 = -\mu\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad (1)$$

where N is the number of lattice sites, and μ is a bias parameter to be determined by training. The outgoing signals from the hidden layer are normalized through the sigmoid function $f(x) = 1/[1+\exp(-x)]$.

We treat the two signals delivered from the hidden to output layer on an equal footing by setting the second weight matrix as $\mathbf{W}_2 = 4\Lambda(1, 1)$ with the bias $b_2 = -2\Lambda$ on the output neuron. Being activated also with the sigmoid function, the final output is then written as $q_k = [1+\tanh(\Lambda z_k)]/2$ for an input $\{s_i^{(k)}\}$ where $z_k = 2[f(y_k-\mu)+f(-y_k-\mu)]-1$ for $y_k \equiv \frac{1}{N}\sum_i s_i^{(k)}$. The pseudo-transition temperature T^* can be typically given by $\langle q\rangle_{T^*} = 1/2$ where $\langle\cdot\rangle_T$ denotes an average over the dataset at temperature T. The training of our two-unit

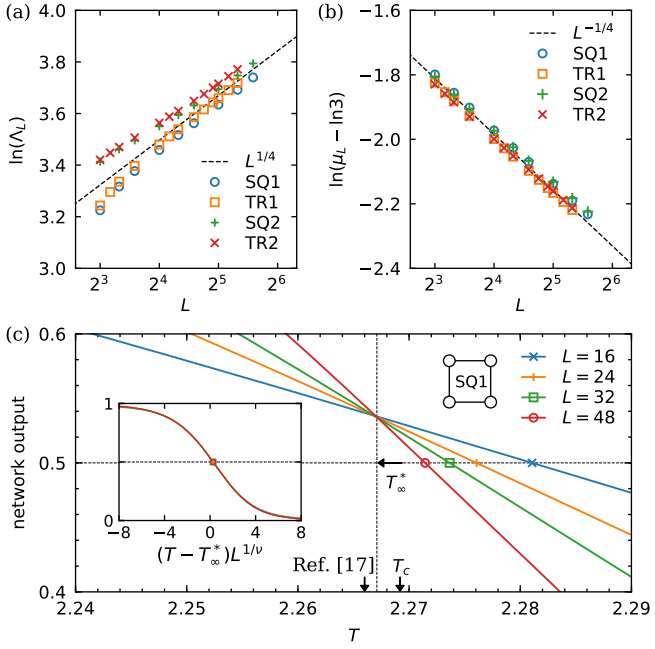

FIG. 2. Learning with the two-unit neural network. The system-size dependence of the network parameters (a) Λ_L and (b) μ_L trained in the square (SQ1,SQ2) and triangular (TR1,TR2) lattices for $T/T_c \in [0.5, 1.5]$ (SQ1,TR1) and $[0, 2]$ (SQ2,TR2). (c) The comparison of the transition point predicted by SQ1, the estimate with the 100-unit network [17], and the exact T_c. The inset shows the scaling collapse of the network outputs with the exponent $\nu = 0.94$.

network is done by minimizing the cross entropy,

$$\mathcal{L} = -\int_{T_l}^{T_u} dT \int_0^1 dy\, \rho_T(y)\,[p_1 \ln q + p_0 \ln(1-q)], \qquad (2)$$

where the reference classifier is given by the Heaviside step function $\Theta(x)$ as $p_0 \equiv \Theta(T - T_c)$ and $p_1 = 1 - p_0$, the network output is denoted by $q \equiv q[z(y, \mu), \Lambda]$, and $\rho_T(y)$ is the density of training data giving y at T.

Treating this learning problem analytically, we find that an interesting system-size dependence is encoded in the network parameters Λ and μ. For simplicity, we approximate $\rho_T(y) \propto \delta(y - m)$ with the order parameter $m \equiv \langle|y|\rangle_T$ by ignoring the fluctuations in the input dataset of y. The minimization of $\mathcal{L}(\Lambda, \mu)$ then leads to the following coupled equations,

$$\int_{T_l}^{T_u} dT\,[2p_{1/2}z_m - z_m \tanh(\Lambda z_m)] = 0, \qquad (3)$$

$$\int_{T_l}^{T_u} dT \frac{\partial}{\partial\mu}\left[2p_{1/2}z_m - \frac{1}{\Lambda}\ln\cosh(\Lambda z_m)\right] = 0, \qquad (4)$$

where $z_m \equiv z(m, \mu)$, and $p_{1/2} = 1/2 - p_0$. The precise bounds of (T_l, T_u) are irrelevant if it is wide enough because the integrands vanish for a large $|\Lambda z_m|$. Thus, the criterion $|z_m| \lesssim 1/\Lambda$ allows us to define the effective bounds as $T^* \pm \delta T$ centered at the pseudo-transition

temperature T^* where $z_m(T^*) = 0$. Below we show that the effective range of a significant T is comparable to the critical window that scales as $L^{-1/\nu}$ with the length scale L of the system, which becomes essential to understand the behavior of the trained parameters of Λ_L and μ_L.

In the area of a small z_m around T^*, we may express z_m for m as $z_m \simeq \frac{3}{8} m_*(m - m_*)$ where $m_* \equiv m(T^*)$. In the transition area, we can also replace m by its finite-size-scaling ansatz $m_L = L^{-\Delta_\sigma} \tilde{m}[(T - T_c)L^{1/\nu}]$ with the scaling dimension $\Delta_\sigma \equiv \beta/\nu$, where $\tilde{m}(x)$ is a scale-invariant function. Then, by expanding Eq. (3) for z_m, we can simply write down the leading order behavior of Λ_L as

$$\Lambda_L \sim L^{2\Delta_\sigma} \cdot \frac{\int_{x_-}^{x_+} p_{1/2} \tilde{m}_* [\tilde{m}(x) - \tilde{m}_*] dx}{\int_{x_-}^{x_+} \tilde{m}_*^2 [\tilde{m}(x) - \tilde{m}_*]^2 dx}, \qquad (5)$$

where $x_\pm = (T_L^* - T_c \pm \delta T_L) L^{1/\nu}$. This reduces to $\Lambda_L \sim L^{2\Delta_\sigma}$ when $x_\pm \sim \mathcal{O}(1)$, which is indeed confirmed in Eq. (4). Through the similar procedures, one can also write down the scaling solution of Eq. (4) as

$$T_L^* - T_c \sim -L^{-1/\nu} \tilde{m}_* \int_{x_-}^{x_+} \tilde{m}(x) dx + 2\tilde{m}_*^2 \delta T_L, \qquad (6)$$

where Λ_L is replaced by $L^{2\Delta_\sigma}$. This holds when $T_L^* - T_c \sim L^{-1/\nu}$ and $\delta T_L \sim L^{-1/\nu}$, or equivalently $x_\pm \sim \mathcal{O}(1)$. For μ_L, the equation $z[L^{-\Delta_\sigma} \tilde{m}(T_L^*), \mu_L] = 0$ leads to the asymptotic behavior of $\mu_L - \ln 3 \sim L^{-2\Delta_\sigma}$.

We numerically verify the behavior of $\Lambda_L \sim L^{2\Delta_\sigma}$ and $\mu_L - \ln 3 \sim L^{-2\Delta_\sigma}$ by performing the learning based on the Monte-Carlo datasets. Specifically, we construct the input data distribution $\rho_T(y)$ by employing the two-parameter Wang-Landau sampling method for energy and magnetization [4, 43–45]. This allows us to compute Eq. (2) directly with the predetermined $\rho_T(y)$, which makes the minimization numerically straightforward. Figure 2 presents Λ_L and μ_L obtained in two dimensions (2D) for the different choices of the underlying geometry and temperature range for the learning. In all cases, the trained parameters become increasingly parallel to the lines of $\Lambda_L \sim L^{1/4}$ and $\mu_L - \ln 3 \sim L^{-1/4}$ for the exact exponent of $\Delta_\sigma = 1/8$ as L increases.

It turns out that although we have only two neurons in the hidden layer, the transition point located in our two-unit network is as accurate as the previous estimate with 100 hidden neurons [17]. In Fig. 2(c) showing the outputs of the network SQ1 trained and examined in the square lattices, the extrapolation from T_L^* finds $T_\infty^* = 2.267(1)$ with the exponent $\nu = 0.94(2)$ which agrees well with the previous estimate of $T_c = 2.266(2)$ with $\nu = 1.0(2)$ [17]. Also, the location of T_∞^* is at the crossings between the curves of different L's, leading to the scale invariance in the network outputs at the transition temperature.

The deviation from the exact T_c is possibly due to the finite-size effects of the systems accessible in the learning which are apparent in Λ_L and μ_L at small L's. Since

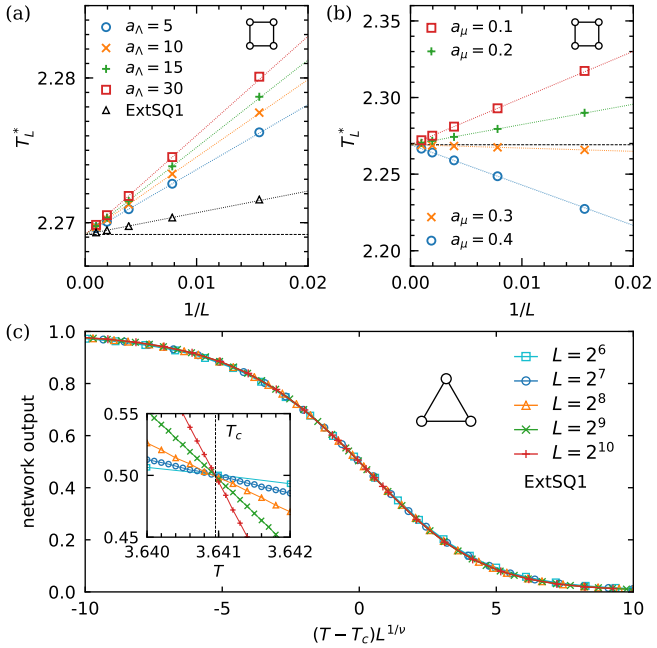

FIG. 3. Interoperability of the two-unit neural network. At the fixed scaling of $\Lambda_L = a_\Lambda L^{1/4}$ and $\mu_L = \ln 3 + a_\mu L^{-1/4}$, the consistency of finding T_c (dashed line) is examined by varying (a) a_Λ at $a_\mu = 0.25$ and (b) a_μ at $a_\Lambda = 15$ for the inputs prepared in the square lattices. The network ExtSQ1 is associated with (a_Λ, a_μ) extrapolated from SQ1 trained in the square lattices. (c) The finite-size-scaling test of the outputs of ExtSQ1 for the inputs from the triangular lattices. The exact values of $T_c = 4/\ln 3$ and $\nu = 1$ are used. The error bars (not shown) are much smaller than the symbol size.

we now know from the analytic results that Λ_L and μ_L should scale asymptotically with the exponent $2\Delta_\sigma$, we may try to remove the finite-size effects by extrapolating the network parameters as $\Lambda_L = a_\Lambda L^{1/4}$ and $\mu_L = \ln 3 + a_\mu L^{-1/4}$ with the expected exponent $\Delta_\sigma = 1/8$. We have observed that this parametrization provides $T_\infty^* \approx 2.269$ which is very close to the exact value of T_c.

An important implication of our analytic results is that the essential information encoded by the learning is only the exponent Δ_σ of the critical behavior. Thus, after the training is done, one may not be able to distinguish the networks by the system-specific properties of the training datasets, such as an underlying lattice geometry and a location of T_c, as long as they are in the same universality class. This implies that one can actually use the network trained in the square lattices for the prediction with the data in the triangular lattices, explaining the previous observation with the 100-unit network in Ref. [17].

Figure 3 shows that locating the precise T_c in the thermodynamic limit is not affected by the training-specific values of (a_Λ, a_μ) when Δ_σ is fixed. For these tests, the input datasets are prepared for the systems with the sizes up to $L = 1024$ in the Monte Carlo simulations [3, 43]. The interoperability between the square and triangular

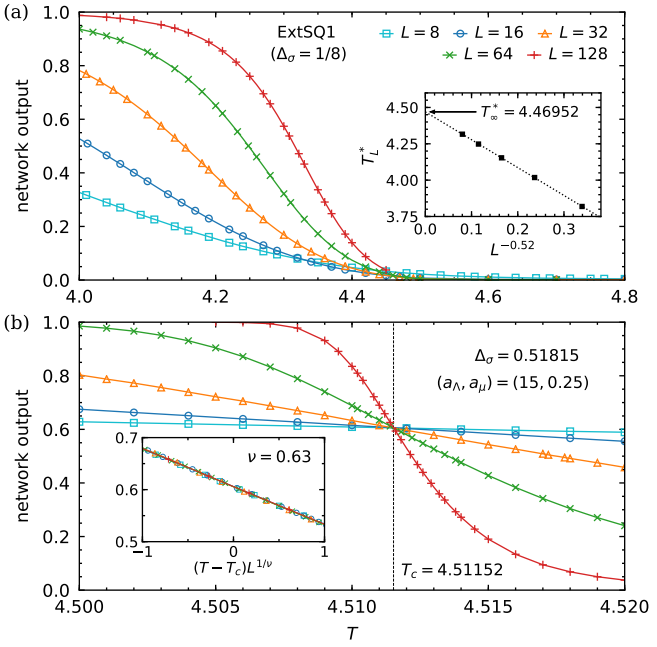

FIG. 4. Test of the 3D Ising model with the extrapolated two-unit networks. ExtSQ1 is used in (a), examining the behavior of the pseudo-transition points T_L^* and the crossing point (not existing) between the output curves. (b) The finite-size-scaling test of the outputs of the network made by using the previous 3D Ising estimate of $\Delta_\sigma = 0.51815$ [47].

lattices is more directly examined in Fig. 3(c) by using the network ExtSQ1 with (a_Λ, a_μ) being extrapolated from the SQ1 parameters trained in the square lattices. Testing with the data of the triangular lattices shows an excellent scaling collapse at the exact values of $T_c = 4/\ln 3$ and $\nu = 1$. The explicit use of SQ1 for small L's provides $T_c \approx 3.637$ [43] which is also comparable to the 100-unit network estimate of $T_c = 3.65(1)$ [17].

Finally, we discuss what happens in practice when the neural network operates on the system with an exponent mismatch. Figure 4(a) shows the case where the network ExtSQ1 with the 2D exponent is applied to the inputs given in 3D cubic lattices. Interestingly, the pseudo-transition temperatures T_L^* show a clean power-law convergence to reach at $T_\infty^* \approx 4.4695$. It is not a precise T_c and with a wrong exponent, but one might say that it is still not too far from the known T_c. However, it clearly loses a scale-invariant point of the output curves, and thus a finite-size scaling test is failed. On the other hand, a network parametrized with the known 3D exponent $\Delta_\sigma = 0.51815$ [47] provides $T_c = 4.51152(1)$ with $\nu = 0.63$ (see Fig. 4(b)) which is in excellent agreement with the previous Monte Carlo estimates [48].

In conclusion, we have shown that the minimal binary structure with two neurons in the hidden layer is essential in understanding the accuracy and interoperability of a neural network observed in the supervised learning of the phase transition in the Ising model. We have found that the scaling dimension of the order parameter is encoded into the system-size dependence of the network parameters in the learning process. This allows the conventional finite-size-scaling analysis with the network outputs to locate the critical temperature and, more importantly, explains why one neural network universally works for different lattices of the same Ising criticality.

Explainable machine learning aims to provide a transparent platform that allows an interpretable prediction which is crucial for the applications that require extreme reliability. In the learning of classifying the phases in the Ising model, we have attempted downsizing the neural network to reveal a traceable structure which turns out to be irreducibly simple and yet not to lose its performance. This suggests a necessity of further studies to explore interpretable building blocks of machine learning in a broader range of physical systems.

* dongheekim@gist.ac.kr

[1] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).
[2] Y. LeCun, Y. Bengio, G. Hinton, Nature **521**, 436 (2015).
[3] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[4] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada Phys. Rev. B **96**, 205152 (2017).
[5] X. Gao and L.-M. Duan, Nat. Commun. **8**, 662 (2017).
[6] D.-L. Deng, X. Li, and S. Das Sarma Phys. Rev. X **7**, 021021 (2017).
[7] Z. Cai and J. Liu. Phys. Rev. B **97**, 035116 (2018).
[8] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac Phys. Rev. X **8**, 011006 (2018).
[9] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang Phys. Rev. B **97**, 085104 (2018).
[10] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, arXiv:1703.05334.
[11] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[12] L. Wang, Phys. Rev. E **96**, 051301(R) (2017).
[13] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101(R) (2017).
[14] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 241104(R) (2017).
[15] X. Y. Xu, Y. Qi, J. Liu, L. Fu, and Z. Y. Meng, Phys. Rev. B **96**, 041119(R) (2017).
[16] Y. Nagai, H. Shen, Y. Qi, J. Liu, and L. Fu, Phys. Rev. B **96**, 161102(R) (2017).
[17] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[18] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).
[19] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Sci. Rep. **7**, 8823 (2017).
[20] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[21] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).
[22] A. Tanaka and A. Tomiya, J. Phys. Soc. Jpn. **86**, 063001 (2017).

[23] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[24] Y. Zhang and E.-A. Kim, Phys. Rev. Lett. **118**, 216401 (2017).
[25] Y. Zhang, R. G. Melko, and E.-A. Kim, Phys. Rev. B **96**, 245119 (2017).
[26] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).
[27] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **86**, 044708 (2017).
[28] P. Zhang, H. Shen, and H. Zhai, Phys. Rev. Lett. **120**, 066401 (2018).
[29] M. J. S. Beach, A. Golubeva, and R, G. Melko, Phys. Rev. B **97**, 045207 (2018).
[30] P. Suchsland and S. Wessel, arXiv:1802.09876.
[31] M. Koch-Janusz and Z. Ringel, arXiv:1704.06279; Nat. Phys. (2018).
[32] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[33] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).
[34] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).
[35] N. C. Costa, W. Hu, Z. J. Bai, R. T. Scalettar, and R. R. P. Singh, Phys. Rev. B **96**, 195138 (2017).
[36] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[37] P. Ponte and R. G. Melko, Phys. Rev. B **96**, 205146 (2017).
[38] K. Ch'ng, N. Vazquez, and E. Khatami, Phys. Rev. E **97**, 013306 (2018).
[39] A. Morningstar and R. G. Melko, arXiv:1708.04622.
[40] S. Iso, S. Shiba, and S. Yokoo, arXiv:1801.07172.
[41] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[42] M. Abadi *et. al.*, *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems* (2015); software available from `https://tensorflow.org`.
[43] See Supplemental Material for the technical details of the simulations and further numerical results.
[44] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).
[45] F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).
[46] D. P. Landau, S.-H. Tsai, and M. Exler, Am. J. Phys. **72**, 1294 (2004).
[47] S. El-Showk, M. F. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin, and A. Vichi, J. Stat. Phys. **151**, 869 (2014).
[48] M. Hasenbusch, Phys. Rev. B **82**, 174433 (2010).

Supplemental Material

A. Training a 50-unit neural network with the L_2-regularization

For the training of the 50-unit neural network, we construct the loss function $\mathcal{L}$ by combining the cross entropy and regularization terms (for the overview, see Ref. [1]) as

$$\mathcal{L}(\mathbf{W}_1, \mathbf{W}_2, \vec{b}_1, \vec{b}_2; \lambda) = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} [p \ln q + (1-p)\ln(1-q)] + \frac{\lambda}{4} \sum_{l=1,2} \|\mathbf{W}_l\|_{\mathrm{F}}^2 ,$$

where the reference classifier p is set to be 1 if the temperature of the input $\vec{\sigma}_T$ is below T_c and 0 otherwise, the network output q is a function of the parameter set $(\mathbf{W}_1, \mathbf{W}_2, \vec{b}_1, \vec{b}_2)$ and the input $\vec{\sigma}_T = \{s_1, \ldots, s_N\}$, and the last term is the L_2-regularization with the strength λ which helps to avoid overfitting. The training dataset includes 1800 spin configurations per temperature sampled with the spin-up-down symmetry being imposed in the Monte Carlo sampling processes. The spin configurations are sampled for 229 temperatures regularly space with step size 0.01 in the range of $(0.5T_c, 1.5T_c)$, giving the total number of the training data $n_{data} = 412200$. The minimization is performed by using the Adam optimizer implemented in TensorFlow [2], and the learning proceeds with the entire training dataset during 30000 epochs at the learning rate 0.0005. The training is done in the $L \times L$ square lattices for $L = 16, 20, 24, 32, 40$.

Figure 5(a) visualizes the weight matrix $\mathbf{W}_1$ of the neural network trained at various values of λ ranging from 0.1 to 0.0001. In the typical validation test of the phase classification with the reserved test dataset of 200 configurations per temperature in the same range, we have very similar success percentages of about 95% for all λ's examined, while slightly higher percentages are found at $0.0005 \leq \lambda \leq 0.01$ (see Fig. 5(b)). However, this simple classification test does not fully validate the actual accuracy and performance of the network for our purpose of investigating its ability of predicting the transition temperature and interoperability with different underlying lattice geometries.

The direct tests of finding the transition temperature with the networks trained in the square lattices are performed with the datasets separately prepared in the triangular and square lattices. The performance shown in these tests seems to be closely related to the existence or non-existence of the plus-minus structure of the weight observed in the weight matrix $\mathbf{W}_1$ which turns out to undergo a crossover from a structured to unstructured one around $\lambda = 0.001$. This crossover does not change with the size of the system that we have examined. One way to notice the change of the visibility of the structure is to look at the weight sum of the incoming links of a hidden neuron as exemplified in Figs. 5(c-e). At $\lambda = 0.005$, the plus-minus structure is very clear since all contributing weights to a hidden neuron are of the same sign. While the weight sums are still well separated into plus, minus, and zeros, defects start to appear at $\lambda = 0.001$, which is shown by the finite length of the bar in Fig. 5(d) that indicates the contribution of the weights with the opposite sign to the total sum at a given neuron. At the weaker regularizations of $\lambda \leq 0.0005$, the length of the bar tends to get larger to be comparable to the magnitude of the weight sum. The behavior of the pseudo-transition temperatures differs as well at these λ's as shown in Fig. 5(f-h). The mark at $L = \infty$ is from the extrapolation with the last three points while the errorbars indicates the combined uncertainty of the three- and four-point fittings. Up to $\lambda = 0.005$ of having the clear structure in $\mathbf{W}_1$, the system-size extrapolation with $1/L$ is very consistent. At $\lambda = 0.001$, the finite-size behavior becomes severe, and below $\lambda = 0.0005$, the accuracy of predicting the transition temperature in the triangular lattices becomes poor as λ further decreases, implying that the overfitting to the reference of a step-function-like classification may have occurred during the training with the data in the square lattices at such small λ.

B. Preparing the input dataset of spin configurations and magnetizations

The Monte Carlo simulations with the Wolff cluster update algorithm [3] is used to produce the input dataset for training and testing. The spin configurations and magnetizations are sampled at every $N/\langle N_c \rangle$ cluster flips, where N and $\langle N_c \rangle$ are the number of the lattice sites and the average cluster size, respectively. In the measurement of the output of the extrapolated two-unit network, the first 10000 samples are thrown away during the thermalization, and then 30 bins of 100000 samples are used for the measurements. The error bars are estimated by using the jackknife method, but it turns out that they are much smaller than the symbol sizes in all plots and thus are not shown in the figures of the main paper.

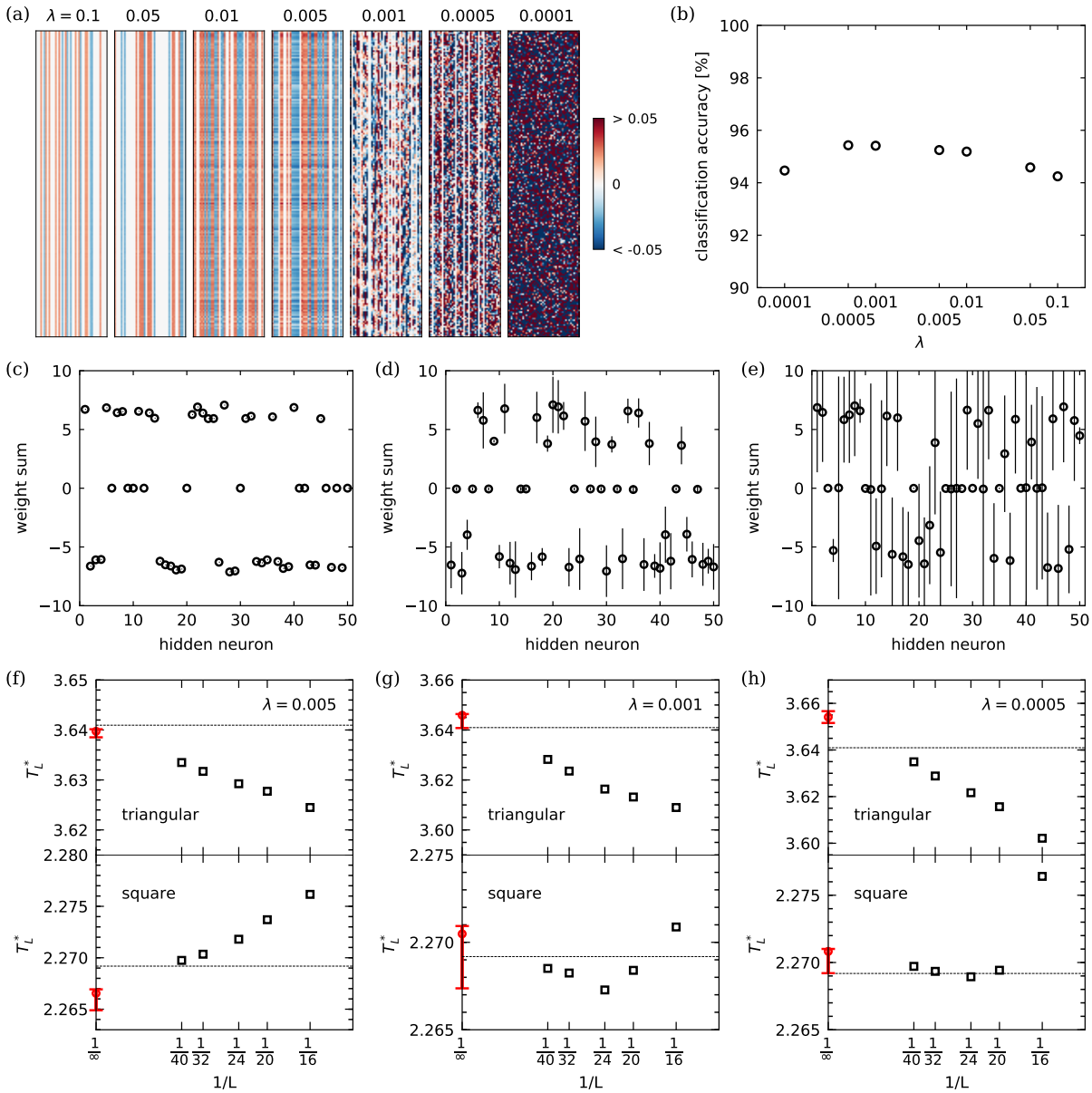

FIG. 5. Dependence of the training of a neural network on the L_2-regularization strength λ. (a) The visualization of the weight matrix $\mathbf{W}_1^{\mathrm{T}}$ of the 50-unit network trained in the 16×16 square lattices. (b) The success percentage in the phase classification test given as a function of λ. The sum of the weights of the incoming links to a hidden neuron is plotted for (c) $\lambda = 0.005$, (d) 0.001, and (e) 0.0005, where the length of the bar indicates the magnitude of the partial sum of the weights having the sign opposite to the total weight sum. The panels (f-h) present the predictions of the transition temperature in the square and triangular lattices at $\lambda = 0.005, 0.001, 0.0005$.

C. The Wang-Landau preparation of $\rho_T(y)$ to train the two-unit network model

We employ the two-parameter Wang-Landau method [4] to generate the joint density of states $g(E, M)$ of the Ising model by following the standard procedures (for instance, see Ref. [5] and the references therein). The variables $E = \sum_{\langle i,j \rangle} s_i s_j$ and $M = \sum_i s_i$ cover all possible values of the energy and total magnetization. In all sizes of the system examined, the flatness criterion of the histogram is set to be larger or equal to 0.9, and the stopping criterion of the modification factor is given as $\ln f < 10^{-8}$. The two-parameter Wang-Landau calculations are known to consume a huge amount of computational time, but still we have obtained $g(E, M)$ up to $L = 48$ ($L = 40$) in the square (triangular) lattices, where the largest calculation took about four months on a single 3.4 GHz Xeon E3 processor. Note that we have obtained a single set of $g(E, M)$ for each system within our computational resources, and thus the

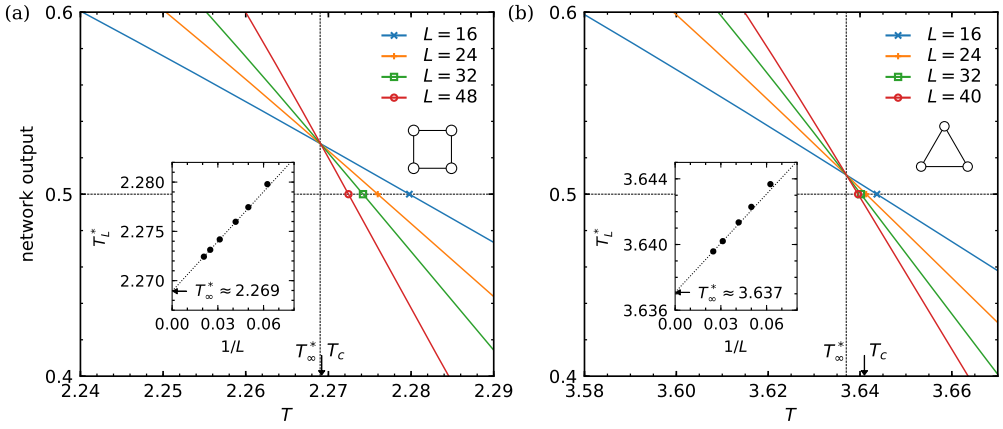

FIG. 6. Locating the transition temperature with the two-unit networks, ExtSQ1 (a) in the square lattices and SQ1 (b) in the triangular lattices. The mark T_∞^* indicates the pseudo-transition point extrapolated in the thermodynamic limit which is identified in the insets as $T_\infty^* \approx 2.269$ (a) in the square lattices and $T^* \approx 3.637$ (b) in the triangular lattices. The arrows with T_c indicate the location of the exact critical temperature.

curves in Fig. 2 of the main paper have given without errorbars. Once the joint density of states $g(E, M)$ is obtained, the distribution function $\rho_T(y)$ of the magnetization y to be used to evaluate the two-unit network can be computed at any temperature T as

$$\rho_T(y)|_{y=M/N} = \frac{\sum_E g(E,M)\exp[JE/k_{\rm B}T]}{\sum_{E,M} g(E,M)\exp[JE/k_{\rm B}T]},$$

where the ferromagnetic coupling J and the Boltzmann constant $k_{\rm B}$ are set to be unity.

D. Supplemental figures of locating the transition point

Figure 6 displays the supplemental figures of finding the transition temperatures with the two-unit neural networks in the square and triangular lattices. The extrapolated network ExtSQ1 is associated with the parameter set (a_Λ, a_μ) fitted to those of the network SQ1 that was explicitly trained in the square lattices. In the validation the network ExtSQ1 for the transition temperature with the data in the square lattices, the extrapolation of the pseudo-transition temperatures (T_L^* along the 0.5-line of the output) provides $T_\infty^* \approx 2.269$ which agrees very well with the exact value of the critical point $T_c = 2/\ln(1+\sqrt{2})$. On the other hand, in the additional interoperability test of SQ1 with the data in the square lattices, we obtain $T_\infty^* \approx 3.637$ which is also very comparable to the value of $T_c = 4/\ln 3$ of the exact solution in the triangular lattices.

* dongheekim@gist.ac.kr

[1] M. A. Nielson, *Neural Networks and Deep Learning* (Determination Press, 2015).
[2] M. Abadi *et. al.*, *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems* (2015); software available from `https://tensorflow.org`.
[3] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[4] D. P. Landau, S.-H. Tsai, and M. Exler, Am. J. Phys. **72**, 1294 (2004).
[5] W. Kwak, J. Jeong, J. Lee, and D.-H. Kim, Phys. Rev. E **92**, 022134 (2015).