# Accelerated Monte Carlo simulations with restricted Boltzmann machines

Li Huang[1] and Lei Wang[2,*]

[1]*Science and Technology on Surface Physics and Chemistry Laboratory, P.O. Box 9-35, Jiangyou 621908, China*

[2]*Beijing National Lab for Condensed Matter Physics and Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*

(Received 15 October 2016; published 4 January 2017)

Despite their exceptional flexibility and popularity, Monte Carlo methods often suffer from slow mixing times for challenging statistical physics problems. We present a general strategy to overcome this difficulty by adopting ideas and techniques from the machine learning community. We fit the unnormalized probability of the physical model to a feed-forward neural network and reinterpret the architecture as a restricted Boltzmann machine. Then, exploiting its feature detection ability, we utilize the restricted Boltzmann machine to propose efficient Monte Carlo updates to speed up the simulation of the original physical system. We implement these ideas for the Falicov-Kimball model and demonstrate an improved acceptance ratio and autocorrelation time near the phase transition point.

DOI: 10.1103/PhysRevB.95.035105

The Monte Carlo method is one of the most flexible and powerful methods for studying many-body systems [1,2]. Its application ranges from the physical sciences [3] including condensed matter physics [4,5], nuclear matter [6], and particle physics [7], all the way to the biological and social sciences [8–10]. Monte Carlo methods randomly sample configurations and obtain the answer as a statistical average. However, because the configuration spaces are exceptionally large for many-body systems, it is typically impossible to perform direct sampling, therefore one resorts to the Markov chain random walk approach to explore the configuration space. In this case, one only needs to know the relative ratio between the probabilities of two configurations.

Designing efficient strategies to explore the configuration space efficiently is at the heart of Markov chain Monte Carlo algorithms. This is, however, a challenging endeavor. Not even mentioning the fundamentally difficult case of glassy energy landscapes, naive Monte Carlo samplings are usually painfully slow close to the phase transitions. These drawbacks motivated noteworthy algorithmic developments in the past decades [11–16]. In essence, those algorithms exploit various physical aspects of the problem for efficient Monte Carlo updates. It is, however, difficult to devise a general strategy to guide optimal Monte Carlo algorithm design.

We address these difficulties in a general setting with insights from machine learning. Recently, there has been a rising interest in applying machine learning approaches to many-body physics problems. This includes classifying phases of matter [17–21], using the neural networks as variational wave functions [22,23], fitting the density functionals [24–26], and solving inverse problems in quantum many-body physics [27,28].

In this paper, we propose a general way to accelerate Monte Carlo simulations of statistical physics problems. We present two algorithmic innovations: a simple *supervised learning* approach to train the restricted Boltzmann machine (RBM) [29,30] as a proxy of the physical distributions, and an efficient Monte Carlo sampling strategy that exploits the latent structure of the RBM. The RBM is a building block for deep learning and plays an important role in its the recent renaissance [31]. The significance of using the RBM in Monte Carlo simulations is that it automatically identifies relevant features (such as correlations and collective modes) in the physics model and proposes updates with correspondingly high acceptance rates and low autocorrelations. This approach makes better use of the sampled Monte Carlo data because, in addition to estimating the physical observables, the RBM builds an adaptive model for the physical probability distribution and guides better explorations.

We illustrate these general ideas using the Falicov-Kimball model [32] as an example. The model describes mobile fermions and localized fermions interacting with on-site interactions. The Hamiltonian reads

$$\hat{H}_{\mathrm{FK}} = \sum_{i,j} \hat{c}_i^{\dagger} \mathcal{K}_{ij} \hat{c}_j + U \sum_{i=1}^{N} \left(\hat{n}_i - \frac{1}{2}\right)\left(x_i - \frac{1}{2}\right), \qquad (1)$$

where $x_i \in \{0,1\}$ is a classical binary variable representing the occupation number of the localized fermion at site $i$. $\hat{c}_i$ is the fermion annihilation operator and $\hat{n}_i \equiv \hat{c}_i^{\dagger}\hat{c}_i$ is the occupation number operator of the mobile fermion. $\mathcal{K}$ is the kinetic energy matrix of the mobile fermions. In the following, we consider the model on a periodic square lattice with $N$ sites. Thus, $\mathcal{K}_{ij} = -t$ for nearest neighbors and is zero otherwise. The $-1/2$ offsets in Eq. (1) ensure that both the mobile and localized fermions are half filled on average. Previous studies show that at $U/t = 4$ and temperature $T/t \approx 0.15$ the system undergoes a phase transition to the checkerboard density-wave (CDW) state [33–35].

Tracing out the mobile fermions, the occupation number of the localized fermions $\mathbf{x} \in \{0,1\}^N$ follows the probability distribution $p_{\mathrm{FK}}(\mathbf{x}) = e^{-F_{\mathrm{FK}}(\mathbf{x})}/Z_{\mathrm{FK}}$, where $Z_{\mathrm{FK}}$ is a normalization factor. The negative "free energy" reads (omitting an unimportant constant $\beta U N/4$)

$$-F_{\mathrm{FK}}(\mathbf{x}) = \frac{\beta U}{2} \sum_{i=1}^{N} x_i + \ln \det(1 + e^{-\beta \mathcal{H}}), \qquad (2)$$

where $\beta = 1/T$ is the inverse temperature and $\mathcal{H}_{ij} = \mathcal{K}_{ij} + \delta_{ij} U (x_i - 1/2)$. One can diagonalize $\mathcal{H}$ to obtain its

*wanglei@iphy.ac.cn

 

eigenvalues $\varepsilon_i$ and compute $\sum_{i=1}^{N} \ln(1+e^{-\beta\varepsilon_i})$ for the second term of Eq. (2). The case of classical fields coupled to quadratic fermions represents a broad class of physics problems, including the double-exchange model [36], the mean-field model for phase fluctuated superconductors [37,38], and the Kitaev model after a transformation [39]. Moreover, if one allows imaginary-time dependence in Eq. (2), it covers an even broader class of condensed matter physics problems such as the Hubbard models [40] and spin-fermion models [41].

To compute the physical properties of the Falicov-Kimball model (1), one can perform the Monte Carlo sampling of the classical variables $\mathbf{x}$. To this end, one designs an ergodic strategy to update the variables $\mathbf{x}$ and decide whether to accept or reject each move. It is sufficient for the Markov chain to converge to the true distribution if the updates satisfy the detailed balance condition [42,43],

$$\frac{T(\mathbf{x}\to\mathbf{x}')}{T(\mathbf{x}'\to\mathbf{x})}\frac{A(\mathbf{x}\to\mathbf{x}')}{A(\mathbf{x}'\to\mathbf{x})}=\frac{p_{\mathrm{FK}}(\mathbf{x}')}{p_{\mathrm{FK}}(\mathbf{x})}, \tag{3}$$

where $T(\mathbf{x}\to\mathbf{x}')$ is the proposal probability of an update and $A(\mathbf{x}\to\mathbf{x}')$ is the acceptance probability of the update. A naive approach would randomly change the classical variables and recompute Eq. (2). To keep the acceptance rate high, one typically applies local updates such as randomly selecting a site $i$ and trying to flip the bit $x_i \to 1-x_i$. In this case the ratio $\frac{T(\mathbf{x}\to\mathbf{x}')}{T(\mathbf{x}'\to\mathbf{x})}=1$, and the acceptance ratio only depends on the free-energy difference of the physical model Eq. (2). However, this naive approach not only has an unfavorable $O(N^4)$ scaling with the system size but also has long autocorrelation times. Several improved update schemes [44–47] have been developed by exploiting the specific features of the Falicov-Kimball model (2). We next present a general approach to propose efficient Monte Carlo updates $T(\mathbf{x}\to\mathbf{x}')$ by training and simulating an RBM.

The RBM is a classical statistical mechanics system defined by the following energy function,

$$E(\mathbf{x},\mathbf{h})=-\sum_{i=1}^{N}a_i x_i-\sum_{j=1}^{M}b_j h_j-\sum_{i=1}^{N}\sum_{j=1}^{M}x_i W_{ij} h_j, \tag{4}$$

where $\mathbf{x}\in\{0,1\}^N$ and $\mathbf{h}\in\{0,1\}^M$ are binary numbers referred to as visible and hidden variables. Besides the local biases $a_i$ and $b_j$, the visible and hidden units are coupled by the weights $W_{ij}$. Crucially, there is no coupling within the visible or hidden units themselves [see Fig. 1(a)]. The joint probability distribution of the visible and hidden variables follows the Boltzmann distribution $p(\mathbf{x},\mathbf{h})=e^{-E(\mathbf{x},\mathbf{h})}/Z$, where the partition function $Z$ is a normalization factor.

Given a sufficiently large number of hidden units, one can tune the parameters of an RBM to let the marginal distribution $p(\mathbf{x})=\sum_{\mathbf{h}}p(\mathbf{x},\mathbf{h})$ approximate any discrete distribution [48,49]. This training task is similar to solving the "inverse Ising problem" in statistical physics [50]. The machine learning community has developed practical approaches to train the RBM by minimizing the negative log likehood $-\ln p(\mathbf{x})$ estimated on the samples drawn from the target distribution [51–53]. References [54,55] use this approach to train the RBM for the thermal distribution of the classical Ising model.

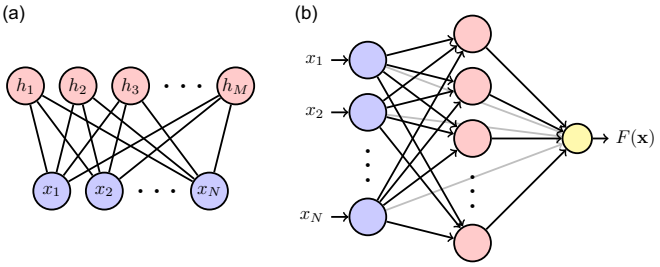


FIG. 1. (a) The restricted Boltzmann machine is an energy-based model for binary stochastic visible and hidden variables. Their probability distribution follows the Boltzmann distribution with the energy function in Eq. (4). (b) Viewing the RBM as a feed-forward neural network which maps the visible variables to the free energy Eq. (5). The gray arrows represent the first term of Eq. (5). The red circles are hidden neurons with the softplus activation function, corresponding to the second term of Eq. (5). Fitting $F(\mathbf{x})$ to the log probability of the physical models [e.g., Eq. (2)] determines the weights and biases of the RBM.

However, training the RBM can be even simpler for the statistical physics problems. Two observations are crucial here. First, in addition to the data set sampled from the target distribution, we do have access to the *unnormalized* probability (i.e., log probability up to a constant) of the statistical physical problems, e.g., Eq. (2). On the other hand, we can write the marginal distribution of the RBM as $p(\mathbf{x})=e^{-F(\mathbf{x})}/Z$, where the free energy of the visible variables reads

$$-F(\mathbf{x})=\sum_{i=1}^{N}a_i x_i+\sum_{j=1}^{M}\ln(1+e^{b_j+\sum_{i=1}^{N}x_i W_{ij}}). \tag{5}$$

Viewing the RBM as a functional mapping from $\mathbf{x}$ to $F(\mathbf{x})$ in Eq. (5) [56], it is tempting to employ a *supervised learning* approach to train its parameters by fitting the free energy Eq. (5) to the one of the physical model Eq. (2). Here, the second observation is important: Since only the relative probability ratio matters in the Markov chain Monte Carlo sampling, one thus only cares about $F(\mathbf{x})$ up to an additive constant. Thus, the intractable partition functions of the RBM and the physical model do not appear in the fitting. The overall constant offset of Eqs. (2) and (5) can be chosen at our convenience in the supervised learning.

We set up a feed-forward neural net for Eq. (5) shown in Fig. 1(b). Its trainable biases and connection weights correspond to the parameters $a_i$, $b_j$, and $W_{ij}$ of the RBM. The red hidden neurons activate via the softplus function $f(z)=\ln(1+e^z)$. The yellow output neuron sums up the outputs of the hidden neurons and the results coming directly from the input neurons for the final result. Physically, the mobile fermions induce effective interactions between the localized fermions. By training an RBM, we represent the fermionic environment by the classical binary fields represented by the hidden neurons. This supervised training approach is significantly simpler and more efficient compared to the conventional unsupervised learning approach [51–53,55].

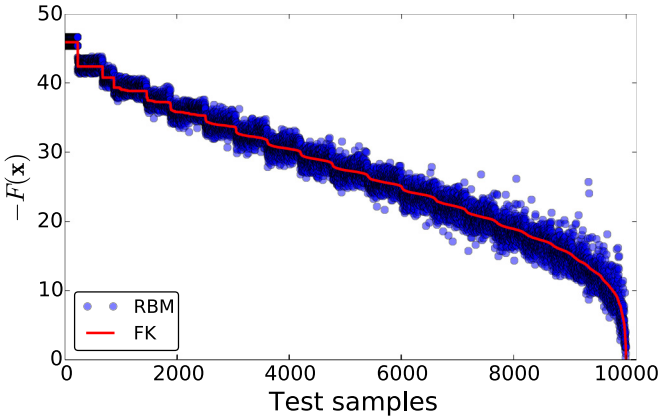


FIG. 2. Fitting of the log probability of the RBM, Eq. (5), to the one of the Falicov-Kamball model, Eq. (2), on a $N = 8^2$ square lattice. The parameters are $U/t = 4$, $T/t = 0.15$, and we use $M = 100$ hidden variables. We offset the data so that the minimum value is at zero.

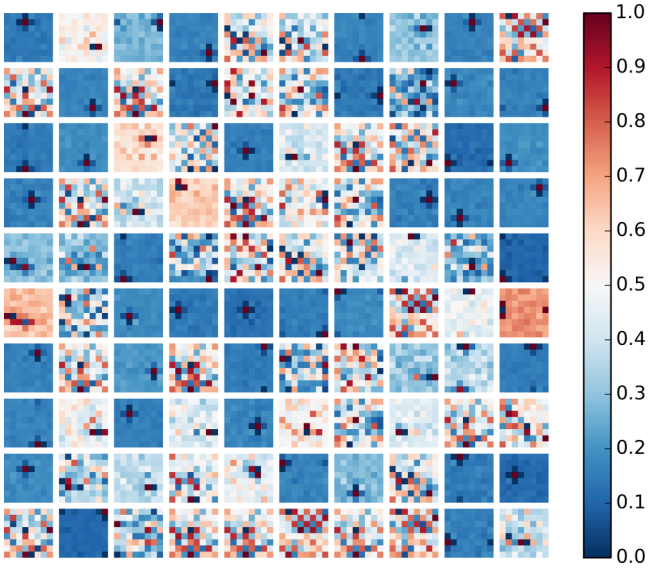


FIG. 3. Connection weights $W_{ij}$ of the restricted Boltzmann machine trained for the Falicov-Kimball model (1) on a square lattice with $N = 8^2$ sites at $U/t = 4$, $T/t = 0.15$. Each tile represents the connection weights of a hidden neuron. We scale the weights in the range of 0–1 for visibility.

To collect data and labels for the supervised learning task, we first run single bit-flip simulations of the model (1) to generate 50 000 independent configurations together with the corresponding negative log probabilities, Eq. (2). In light of the similarity between Eqs. (2) and (5), we set $a_i = \beta U/2$ and focus on the fitting of $b_j$ and $W_{ij}$. Since the sum of the softplus functions Eq. (5) is always positive, we subtract the minimum value in the collected labels to make them non-negative. This again uses the fact that the fitting is only up to an additive constant. We use 80% of the collected data for training and test on the remaining 20% data to check the generalizability of the fitted neural net. In addition, we apply $L2$ regulation to the connection weights. This not only prevents overfitting but also makes the sampling of the RBM easier [53].

Figure 2 shows the negative log probability of the test samples as a solid red line and the predictions of the neural net as blue dots. The fitting successfully captures the overall trend of the physical probability distribution. Moreover, the connection weights $W_{ij}$ shown in Fig. 3 also acquire appealing physical meaning by acting as feature detectors for the visible variables. One clearly sees many cross structures corresponding to the staggered density-wave pattern of the localized fermions. There are also several features extending throughout the lattice, indicating that the hidden unit is sensitive to the nonlocal features of the physical model. These learned weights change with the temperature [57], showing that the RBM can automatically pick up the characteristic features of the physics model. When simulating the RBM as a statistical mechanics system, an activated hidden neuron will stimulate the corresponding features in the visible layer. This is similar to the application of RBM to images of handwritten digits. There, the RBM can pick up features such as pen strokes and draw new images containing digits [51,52].

After training of the RBM, we use it to generate efficient Monte Carlo updates for the physical variables. To this end, we simulate the RBM using the Monte Carlo method and ensure the updates satisfy the detailed balance condition $\frac{T(\mathbf{x}\to\mathbf{x}')}{T(\mathbf{x}'\to\mathbf{x})} = \frac{p(\mathbf{x}')}{p(\mathbf{x})}$ for the visible variables [57]. Therefore, the Metropolis-Hastings [42,43] solution of the acceptance ratio in Eq. (3) reads

$$A(\mathbf{x}\to\mathbf{x}') = \min\left[1, \frac{p(\mathbf{x})}{p(\mathbf{x}')}\cdot\frac{p_{\mathrm{FK}}(\mathbf{x}')}{p_{\mathrm{FK}}(\mathbf{x})}\right]. \tag{6}$$

Ideally, the acceptance ratio is one if the RBM fits the Falicov-Kimball model perfectly. In this case, one accepts all the proposals from the RBM as was attempted in Ref. [55]. However, in practice, the fitting of the RBM is never perfect, given the limited number of hidden units. Equation (6) corrects this error by rejecting unlikely proposals and guaranteeing *exact* physical results even with an imperfectly trained RBM.

A standard way to simulate the RBM is the blocked Gibbs sampling approach. Because of the RBM's bipartite architecture, the conditional probability of the hidden variables factorizes $p(\mathbf{h}|\mathbf{x}) = p(\mathbf{x},\mathbf{h})/p(\mathbf{x}) = \prod_{j=1}^{M} p(h_j|\mathbf{x})$. Similarly, one has $p(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^{N} p(x_i|\mathbf{h})$, and

$$p(h_j = 1|\mathbf{x}) = \sigma\left(b_j + \sum_{i=1}^{N} x_i W_{ij}\right), \tag{7}$$

$$p(x_i = 1|\mathbf{h}) = \sigma\left(a_i + \sum_{j=1}^{M} W_{ij} h_j\right), \tag{8}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function. The blocked Gibbs sampler samples back and forth between the hidden and visible layers using Eqs. (7) and (8), shown in Fig. 4(a). When the simulation of the RBM is much cheaper than the original physical model, one can perform many of these blocked Gibbs sampling steps before evaluating Eq. (6). The RBM suggests nonlocal updates for the visible variables while still keeping the acceptance ratio high.

Moreover, we argue that already a single Gibbs sampling step can be beneficial for the simulation of the statistical

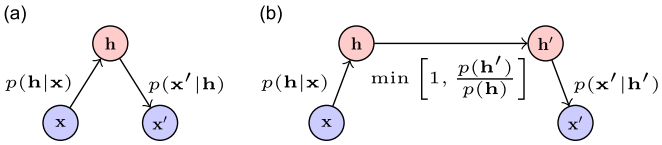


FIG. 4. Two strategies of proposing Monte Carlo updates using the RBM. (a) The blocked Gibbs sampling. Given the visible variables $\mathbf{x}$, we sample the hidden variable $\mathbf{h}$ under the conditional probability $p(\mathbf{h}|\mathbf{x})$, Eq. (7), then sample the visible variables under the conditional probability $p(\mathbf{x}'|\mathbf{h})$, Eq. (8). (b) The blocked Gibbs sampling with an additional Metropolis step in between. It updates the hidden variable $\mathbf{h} \to \mathbf{h}'$ according to the log probability of the hidden variables, Eq. (9).

physics model. Importantly, flipping a hidden variable with the local Gibbs sampling may have nonlocal effects on the physical variables. This is because the hidden neuron may control an extended region of visible variables, as shown in Fig. 3. To further encourage this effect, one can perform additional sampling of the hidden variables in between the Gibbs sampling steps, shown in Fig. 4(b). We suggest changing the hidden variables $\mathbf{h} \to \mathbf{h}'$ and accepting the update with probability $\min[1, \frac{p(\mathbf{h}')}{p(\mathbf{h})}]$, where $p(\mathbf{h}) = \sum_{\mathbf{x}} p(\mathbf{x},\mathbf{h}) = e^{-F(\mathbf{h})}/Z$ is the free energy of the hidden variables

$$-F(\mathbf{h}) = \sum_{j=1}^{M} b_j h_j + \sum_{i=1}^{N} \ln(1 + e^{a_i + \sum_{j=1}^{M} W_{ij} h_j}). \quad (9)$$

Notice that the visible variables have been traced out in Eq. (9), and sampling according to $p(\mathbf{h})$ captures the generic distribution of the hidden features and is unaffected by the current visible variables. This further improves the sampling of RBM by avoiding the visible and hidden variables to lock each others' feature.

We demonstrate the improvement of using RBM in the Monte Carlo simulation in Fig. 5. The acceptance ratio of the single bit-flip update simulation of the Falicov-Kimball model (1) decreases monotonously with decreasing temperature, shown in Fig. 5(a). This is because when the system enters into the CDW phase it is harder to add or remove fermions. In contrast, the acceptance ratio of RBM updates remains high in the whole temperature range across the phase transition. This is because the RBM correctly captures the distribution of the physical system. As a better measure of the improvement, Fig. 5(b) shows the autocorrelation time measured in units of Monte Carlo steps per lattice site [1]. The RBM updates reduce the autocorrelation time by at least a factor of 2. The scheme of Fig. 4(b) with four additional bit-flip attempts of the hidden units further reduces the autocorrelation time. The overhead of performing the sampling using the RBM is $O(MN)$, which is negligible compared to the cost of computing Eq. (2) via diagonalizing the fermionic Hamiltonian.

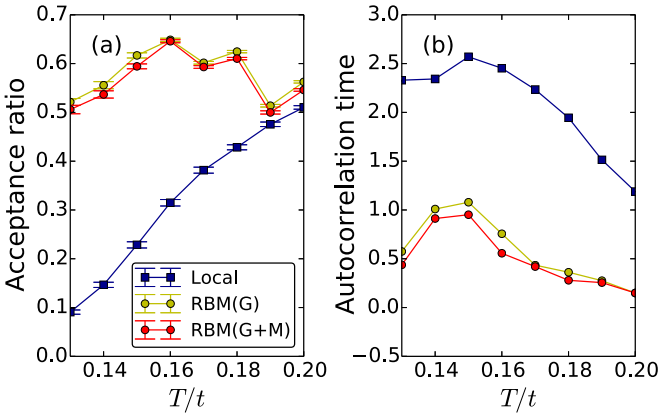


FIG. 5. (a) The acceptance ratio and (b) the total energy autocorrelation time of the Falicov-Kimball model on a $N = 8^2$ square lattice. Blue squares denote results of local bit-flip updates. The yellow and red dots are using the RBM update schemes of Figs. 4(a) and 4(b), respectively. The critical temperature is at $T/t \approx 0.15$ [33–35]. The estimated physical observables agree within error bars for all sampling approaches.

The proposed approach is general. Besides the Falikov-Kimball model and its relatives mentioned after Eq. (2), it is straightforward to use the RBM in Monte Carlo simulations with binary degree of freedoms, such as the Ising and $Z_2$ gauge fields models, variational [58] and determinantal [40] Monte Carlo simulations of the Hubbard models, and the Fermi bag approach of lattice field theories [59]. For models with continuous variables, one can use the RBM with Gaussian variables [53]. The RBM sampling approach can also be used in combination with the other efficient sampling approaches developed for statistical mechanics problems [11–16,44–47].

To make the presentation cleaner, we divided the computational tasks into three phases: collecting the training data, fitting the RBM, and the actual Monte Carlo simulations. In the future, one can use online learning to optimize the RBM progressively with newly collected configurations. After training the RBM in the equilibration phase of the Monte Carlo simulation, one can use it to generate new samples with improved efficiency. One also needs to check the scalability of the proposed approach for larger and more complicated physical systems.

Another future extension is to explore deep Boltzmann machines [60] and deep belief nets [51] for Monte Carlo simulations. Deeper hierarchical structures may allow even higher level abstractions of the physical degree of freedoms. There were observations indicating that a deep structure improves the mixing time of the Monte Carlo sampling [61]. To further exploit the translational symmetry of the physical problems, one may consider using a shift invariant RBM [22,62] or a convolutional RBM [63–65].

Last but not least, in this paper we view the RBM as a generative model and use it to accelerate the Monte Carlo simulation of physical systems. On the other hand, along the lines of Ref. [18], it is also interesting to view the RBM as an unsupervised feature detector and explore the patterns in the weights and the latent variables for the discovery of physical knowledge.

Recently we noted a related work [66].

L.W. is supported by the Ministry of Science and Technology of China under Grant No. 2016YFA0302400 and the startup grant of IOP-CAS. L.H. is supported by the Natural Science Foundation of China Grant No. 11504340. We acknowledge Jun-Wei Liu, Ye-Hua Liu, Zi-Yang Meng,

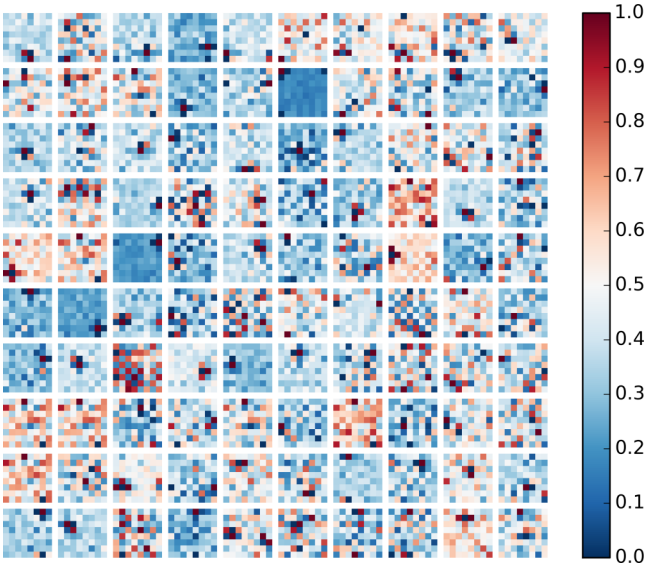


FIG. 6. Connection weights $W_{ij}$ of the RBM at $T/t = 0.13$. The other parameters are the same as the Fig. 3 of the main texts.

and Yang Qi for useful discussions. We use the Keras library [67] for training of the neural network and the ALPS library [68] for the Monte Carlo data analysis. Our implementation of the restricted Boltzmann machine is based on the scikit-learn library [69].

## APPENDIX A: LEARNED WEIGHTS AT $T/t = 0.13$

The learned weights change drastically near the critical temperature. Figure 6 shows the weights learned by the RBM at lower temperature. Compared to Fig. 3 at $T/t = 0.15$, there are more hidden neurons controlling extended regions of the visible variables, indicating enlarged correlation length at lower temperature. The checkerboard pattern of the low temperature phase is also more visible.

## APPENDIX B: PROOF OF THE DETAILED BALANCE CONDITIONS

We prove the simulation of the RBM shown in Fig. 4 of the main texts satisfies the detailed balance condition.

For the case of blocked Gibbs sampling shown in Fig. 4(a)

$$\begin{aligned}\frac{T(\mathbf{x}\to\mathbf{x}')}{T(\mathbf{x}'\to\mathbf{x})} &= \frac{p(\mathbf{h}|\mathbf{x})p(\mathbf{x}'|\mathbf{h})}{p(\mathbf{h}|\mathbf{x}')p(\mathbf{x}|\mathbf{h})} \\ &= \frac{p(\mathbf{x},\mathbf{h})p(\mathbf{x}',\mathbf{h})}{p(\mathbf{x})p(\mathbf{h})}\cdot\frac{p(\mathbf{x}')p(\mathbf{h})}{p(\mathbf{x}',\mathbf{h})p(\mathbf{x},\mathbf{h})} \\ &= \frac{p(\mathbf{x}')}{p(\mathbf{x})} \end{aligned} \tag{B1}$$

For the case of Gibbs sampler with additional Metropolis steps for the hidden variables shown in Fig. 4(b)

$$\begin{aligned}\frac{T(\mathbf{x}\to\mathbf{x}')}{T(\mathbf{x}'\to\mathbf{x})} &= \frac{p(\mathbf{h}|\mathbf{x})T(\mathbf{h}\to\mathbf{h}')p(\mathbf{x}'|\mathbf{h}')}{p(\mathbf{h}'|\mathbf{x}')T(\mathbf{h}'\to\mathbf{h})p(\mathbf{x}|\mathbf{h})} \\ &= \frac{p(\mathbf{x},\mathbf{h})p(\mathbf{x}',\mathbf{h}')}{p(\mathbf{x})p(\mathbf{h}')}\cdot\frac{p(\mathbf{x}')p(\mathbf{h})}{p(\mathbf{x}',\mathbf{h}')p(\mathbf{x},\mathbf{h})}\cdot\frac{p(\mathbf{h}')}{p(\mathbf{h})} \\ &= \frac{p(\mathbf{x}')}{p(\mathbf{x})}. \end{aligned} \tag{B2}$$

For the second equality we use that the Metropolis update of the hidden variable satisfies $T(\mathbf{h}\to\mathbf{h}')/T(\mathbf{h}'\to\mathbf{h}) = p(\mathbf{h}')/p(\mathbf{h})$. This proof generalizes to compositions of several of such updates.

[1] M. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, Oxford, UK, 1999).
[2] W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford University Press, Oxford, UK, 2006).
[3] J. E. Gubernatis, *The Monte Carlo Method in the Physical Sciences*, AIP Conf. Proc. Vol. 690 (AIP, Melville, NY, 2003).
[4] W. M. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Mod. Phys. **73**, 33 (2001).
[5] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods* (Cambridge University Press, Cambridge, UK, 2016).
[6] J. Carlson, S. Gandolfi, F. Pederiva, S. C. Pieper, R. Schiavilla, K. E. Schmidt, and R. B. Wiringa, Rev. Mod. Phys. **87**, 1067 (2015).
[7] Z. Fodor and C. Hoelbling, Rev. Mod. Phys. **84**, 449 (2012).
[8] B. F. Manly, *Randomization, Bootstrap and Monte Carlo Methods in Biology*, Vol. 70 (CRC Press, Boca Raton, FL 2006).
[9] C. J. Mode, *Applications of Monte Carlo Methods in Biology, Medicine and Other Fields of Science* (InTech, Rijeka, Croatia, 2011).
[10] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Vol. 53 (Springer, Berlin, 2003).
[11] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[12] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987).
[13] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[14] N. V. Prokof'ev, Svistunov, BV, and I. S. Tupitsyn, J. Exp. Theor. Phys. **87**, 310 (1998).
[15] F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).
[16] H. G. Evertz, G. Lana, and M. Marcu, Phys. Rev. Lett. **70**, 875 (1993).
[17] J. Carrasquilla and R. G. Melko, arXiv:1605.01735.
[18] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[19] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, arXiv:1608.07848.
[20] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, arXiv:1609.02552.
[21] A. Tanaka and A. Tomiya, arXiv:1609.09087.
[22] G. Carleo and M. Troyer, arXiv:1606.02318.
[23] D.-L. Deng, X. Li, and S. Das Sarma, arXiv:1609.09060.
[24] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. **108**, 253002 (2012).
[25] L. Li, T. E. Baker, S. R. White, and K. Burke, arXiv:1609.03705.
[26] F. Brockherde, L. Li, K. Burke, and K.-R. Müller, arXiv:1609.02815.

[27] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).

[28] L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.

[29] P. Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1 (MIT Press, Cambridge, MA, 1986), pp. 194–281.

[30] G. E. Hinton, Neural Comput. **14**, 1771 (2002).

[31] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).

[32] L. M. Falicov and J. C. Kimball, Phys. Rev. Lett. **22**, 997 (1969).

[33] M. M. Maśka and K. Czajka, Phys. Rev. B **74**, 035109 (2006).

[34] A. E. Antipov, E. Gull, and S. Kirchner, Phys. Rev. Lett. **112**, 226401 (2014).

[35] A. E. Antipov, Y. Javanmard, P. Ribeiro, and S. Kirchner, Phys. Rev. Lett. **117**, 146601 (2016).

[36] G. Alvarez and A. Feiguin, in *Nanoscale Phase Separation and Colossal Magnetoresistance* (Springer, Berlin, 2003), pp. 125–156.

[37] M. Mayr, G. Alvarez, C. Şen, and E. Dagotto, Phys. Rev. Lett. **94**, 217001 (2005).

[38] Y. Dubi, Y. Meir, and Y. Avishai, Nature (London) **449**, 876 (2007).

[39] J. Nasu, M. Udagawa, and Y. Motome, Phys. Rev. Lett. **113**, 197205 (2014).

[40] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, Phys. Rev. D **24**, 2278 (1981).

[41] E. Berg, M. A. Metlitski, and S. Sachdev, Science **338**, 1606 (2012).

[42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[43] W. K. Hastings, Biometrika **57**, 97 (1970).

[44] Y. Motome and N. Furukawa, J. Phys. Soc. Jpn. **68**, 3853 (1999).

[45] J. L. Alonso, L. A. Fernandez, F. Guinea, V. Laliena, and V. Martin-Mayor, Nucl. Phys. **596**, 587 (2001).

[46] G. Alvarez, C. Şen, N. Furukawa, Y. Motome, and E. Dagotto, Comput. Phys. Commun. **168**, 32 (2005).

[47] S. Kumar and P. Majumdar, Eur. Phys. J. B **50**, 571 (2006).

[48] Y. Freund and D. Haussler, Unsupervised learning of distributions of binary vectors using two layer networks, in *Proceeding NIPS'91 Proceedings of the 4th International Conference on Neural Information Processing Systems*, edited by J. E. Moody S. J. Hanson, and R. P. Lippmann (Morgan Kaufmann Publishers, San Francisco, CA, 1991), pp. 912–919 .

[49] N. Le Roux and Y. Bengio, Neural Comput. **20**, 1631 (2008).

[50] J. Albert and R. H. Swendsen, Phys. Procedia **57**, 99 (2014).

[51] G. E. Hinton, S. Osindero, and Y. W. Teh, Neural Comput. **18**, 1527 (2006).

[52] T. Tieleman, *Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient* (ACM, New York, 2008).

[53] G. E. Hinton, in *Neural Networks: Tricks of the Trade* (Springer, Berlin, 2012), pp. 599–619.

[54] P. Mehta and D. J. Schwab, arXiv:1410.3831.

[55] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).

[56] B. M. Marlin, K. Swersky, B. Chen, and N. De Freitas, JMLR Workshop Conf. Proc. **9**, 509 (2010).

[57] See appendix for the weights learned at $T/t = 0.13$ and proof of the detailed balance condition of the Monte Carlo updates using RBM.

[58] H. Yokoyama and H. Shiba, J. Phys. Soc. Jpn. **56**, 1490 (1987).

[59] S. Chandrasekharan, Phys. Rev. D **82**, 025007 (2010).

[60] R. Salakhutdinov and G. E. Hinton, JMLR Workshop Conf. Proc. **5**, 448 (2009).

[61] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, JMLR Workshop Conf. Proc. **28**, 552 (2013).

[62] K. Sohn and H. Lee, in *Proceedings of the 29th International Conference on Machine Learning*, edited by J. Langford and J. Pineau (Omnipress, Madison, WI, 2012), p. 1311.

[63] G. Desjardins and Y. Bengio, Empirical evaluation of convolutional RBMs for vision, DIRO Technical Report No. 1327, 2008 (unpublished).

[64] M. Norouzi, M. Ranjbar, and G. Mori, in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009* (IEEE, New York, 2009), pp. 2735–2742.

[65] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations* (ACM, New York, 2009).

[66] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101 (2017).

[67] Keras (http://keras.io) is a high level deep learning library based on Theano (http://deeplearning.net/software/theano) and TensorFlow (https://www.tensorflow.org).

[68] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawlowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwock, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel, J. Stat. Mech.: Theor. Exp. (2011) P05001.

[69] http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.BernoulliRBM.html##.