Neural Network Renormalization Group

Shuo-Hui Li[1,2] and Lei Wang[1,3,*]
[1]Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Songshan Lake Materials Laboratory, Dongguan, Guangdong 523808, China

(Received 14 March 2018; revised manuscript received 22 October 2018; published 26 December 2018)

We present a variational renormalization group (RG) approach based on a reversible generative model with hierarchical architecture. The model performs hierarchical change-of-variables transformations from the physical space to a latent space with reduced mutual information. Conversely, the neural network directly maps independent Gaussian noises to physical configurations following the inverse RG flow. The model has an exact and tractable likelihood, which allows unbiased training and direct access to the renormalized energy function of the latent variables. To train the model, we employ probability density distillation for the bare energy function of the physical problem, in which the training loss provides a variational upper bound of the physical free energy. We demonstrate practical usage of the approach by identifying mutually independent collective variables of the Ising model and performing accelerated hybrid Monte Carlo sampling in the latent space. Lastly, we comment on the connection of the present approach to the wavelet formulation of RG and the modern pursuit of information preserving RG.

DOI: 10.1103/PhysRevLett.121.260601

The renormalization group (RG) is one of the central schemes in theoretical physics, whose impacts span from high-energy [1] to condensed matter physics [2,3]. In essence, RG keeps the relevant information while reducing the dimensionality of statistical data. Besides its conceptual importance, practical RG calculations have played important roles in solving challenging problems in statistical and quantum physics [4,5]. A notable recent development is to perform RG calculations using tensor network machinery [6–18].

The relevance of RG goes beyond physics. For example, in deep learning applications, the inference process in image recognition resembles the RG flow from microscopic pixels to categorical labels. Indeed, a successfully trained neural network extracts a hierarchy of increasingly higher-level concepts in its deeper layers [19]. In light of such intriguing similarities, Refs. [20–23] drew connections between deep learning and the RG, Ref. [24] proposed an RG scheme based on mutual information maximization, Ref. [25] employed deep learning to study holography duality, and Ref. [26] examined the adversarial examples from a RG perspective. Since the discussions are not totally uncontroversial [21,23,24,27,28], it remains highly desirable to establish a more concrete, rigorous, and constructive connection between RG and deep learning. Such a connection will not only bring powerful deep learning techniques into solving complex physics problems but also benefit theoretical understanding of deep learning from a physics perspective.

In this Letter, we present a neural network based variational RG approach (NeuralRG) for statistical physics problems. In this scheme, the RG flow arises from iterative probability transformation in a neural network. Integrating the latest advances in deep learning including normalizing flows [29–37], probability density distillation [38], and tensor network architectures, in particular, the multiscale entanglement renormalization ansatz (MERA) [6], the proposed NeuralRG approach has a number of interesting theoretical properties (variational, exact, and tractable likelihood, principled structure design via information theory) and high computational efficiency. The NeuralRG approach is closer in spirit to the original proposal based on Bayesian net [20] than more recent discussions on Boltzmann machines [21,23] and principal component analysis [22].

Figure 1(a) shows the proposed architecture. Each building block is a diffeomorphism, i.e., a bijective and differentiable function parametrized by a neural network, denoted by a bijector [39,40]. Figure 1(b) illustrates one realization of the bijector using real-valued nonvolume preserving flows (Real NVP) [32,41], which is one of the reversible generative models known as the normalizing flows [29–37].

The network relates the physical variables $\boldsymbol{x}$ and the latent variables $\boldsymbol{z}$ via an invertible transformation $\boldsymbol{x} = g(\boldsymbol{z})$. Their probability densities are also related [50]

$$\ln q(\boldsymbol{x}) = \ln p(\boldsymbol{z}) - \ln \left| \det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}\right)\right|, \tag{1}$$

where $q(\boldsymbol{x})$ is the normalized probability density of the physical variables. And $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{1})$ is the prior probability density of the latent variables chosen to be a normal distribution. The second term of Eq. (1) is the

log-Jacobian determinant. Since the log probability can be interpreted as a negative energy function, Eq. (1) shows that the renormalization of the effective coupling is provided by the log-Jacobian at each transformation step.

Since diffeomorphisms form a group, an arbitrary composition of the building blocks is still a bijector. This motivates the modular design shown in Fig. 1(a). The layers alternate between disentangler blocks and decimator blocks. The disentangler blocks in light gray reduce correlation between the inputs and pass on less correlated outputs to the next layer. While the decimator blocks in dark gray pass only a subset of its outputs to the next layer and treat the remaining ones as irrelevant latent variables indicated by the crosses. The RG flow corresponds to the inference of the latent variables given the physical variables, $\boldsymbol{z} = g^{-1}(\boldsymbol{x})$. The kept degrees of freedom emerge as renormalized collective variables at coarser scales during the inference. In the reversed direction, the latent variables are injected into the neural network at different depths. And they affect the physical variables at different length scales.

The proposed NeuralRG architecture shown in Fig. 1(a) is largely inspired by the MERA structure [6]. In particular, stacking bijectors to form a reversible transformation is analogous to the quantum circuit interpretation of MERA. The difference is that the neural network transforms probability densities instead of quantum states. Compared to the tensor networks, the neural network has the flexibility that the blocks can be arbitrarily large and long-range connected. Moreover, arbitrary complex NeuralRG architecture constructed in a modular fashion can be trained efficiently using differentiable programming frameworks [51,52]. In practice, one can let the bijectors in the same layer share weights due to the translational invariances of the physical problem [53].

Compared to ordinary neural networks used in deep learning, the architecture in Fig. 1(a) has stronger physical and information theoretical motivations. To see this, we consider a simpler reference structure shown in Fig. 2(a) where one uses disentangler blocks at each layer. The resulting structure resembles a time-evolving block decimation network [54]. Since each disentangler block connects only a few neighboring variables, the causal light cone of the physical variables at the bottom can only reach a region of latent variables proportional to the depth of the network. Therefore, the correlation length of the physical variables is limited by the depth of the disentangler layers. The structure of Fig. 2(a) is sufficient for physical problems with finite correlation length, i.e., away from the criticality.

On the other hand, a network formed only by the decimators is similar to the tree tensor network [55]. For example, the mutual information (MI) between the variables at each decimation step shown in Fig. 2(b) follows

$$I(A:B) = I(z_1 \cup a : b \cup z_4) = I(a:b). \tag{2}$$

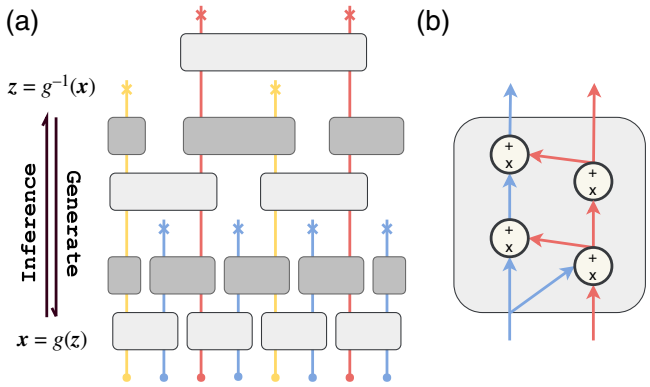

FIG. 1. (a) The NeuralRG network is formed by stacking bijector networks into a hierarchical structure. The solid dots at the bottom are the physical variables $\boldsymbol{x}$ and the crosses are the latent variables $\boldsymbol{z}$. Each block is a bijector. The light gray and the dark gray blocks are the disentanglers and the decimators, respectively. The RG flows from bottom to top, which corresponds to the inference of the latent variables conditioned on the physical variables. Conversely, one can directly generate physical configurations by sampling the latent variables according to the prior distribution and passing them downwards through the network. (b) The internal structure of the bijector block consists of normalizing flows [32].

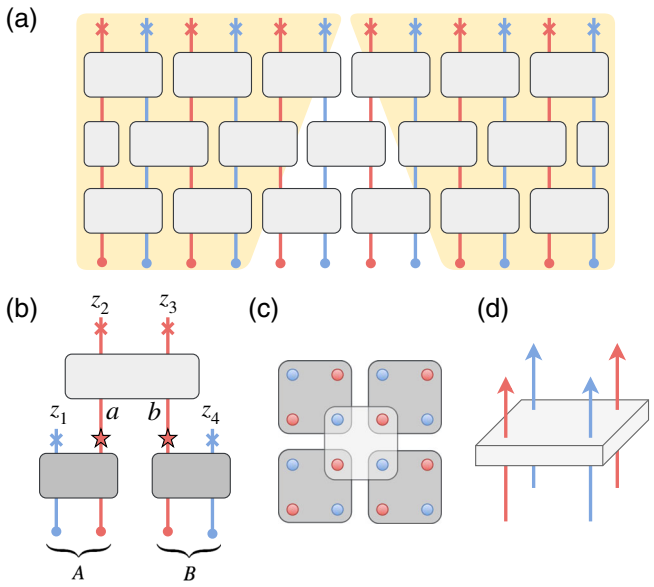

FIG. 2. (a) A reference neural network architecture with only disentanglers. The physical variables in the two shaded regions are uncorrelated because their causal light cones do not overlap in the latent space. (b) Mutual information is conserved at the decimation step, see Eq. (2). (c) The arrangement of the bijectors in the two-dimensional space. (d) Each bijector acts on four variables. Disentanglers reduce mutual information between variables. While for decimators, only one of its outputs is passed on to the next layer and the others are treated as latent variables.

The first equality is due to the MI being invariant under invertible transformation of variables within each group. While the second equality is due to the random variables z_1 and z_4 being independent of all other variables. Applying Eq. (2) recursively at each decimation step, one concludes that the MI between two sets of physical variables is limited by the top layer in a bijective net of the tree structure. One thus needs to allocate sufficient resources in the bottleneck blocks to successfully capture the MI of the data.

It is straightforward to generalize the NeuralRG architecture in Fig. 1 to handle data in higher dimensional space. For example, one can stack layers of bijectors in the form of Fig. 2(c). These bijectors accept 2×2 inputs as shown in Fig. 2(d). For the decimator, only one out of four outputs is passed on to the next layer. In a network with only disentanglers, the depth should scale linearly with system size to capture diverging correlation length at criticality. While the required depth only scales logarithmically with system size if one employs the MERA-like structure. Note that different from the tensor network modeling of quantum states [56], the MERA-like architecture is sufficient to model classical systems with short-range interactions even at criticality since they exhibit the MI area law [57].

Building the neural network using normalizing flows provides a generative model with explicit and tractable likelihoods compared to previous studies [21,23,24,58–60]. This feature is valuable for studying physical problems because one can have unbiased and quantitative control of the training and evaluation of the model. Consider a standard setup in statistical physics, where one has accesses to the bare energy function, i.e., the *unnormalized* probability density $\pi(\boldsymbol{x})$ of a physical problem, direct sampling of the physical configurations is generally difficult due to the intractable partition function $Z = \int d\boldsymbol{x}\pi(\boldsymbol{x})$ [61]. The standard Markov chain Monte Carlo (MCMC) approach suffers from the slow mixing problem in many cases [62].

We train the NeuralRG network by minimizing the probability density distillation (PDD) loss

$$\mathcal{L} = \int d\boldsymbol{x} q(\boldsymbol{x})[\ln q(\boldsymbol{x}) - \ln \pi(\boldsymbol{x})], \tag{3}$$

which was recently employed by DeepMind to train the Parallel WaveNet [38]. The first term of the loss is the negative entropy of the model density $q(\boldsymbol{x})$, which favors diversity in its samples. While the second term corresponds to the expected energy since $-\ln \pi(\boldsymbol{x})$ is the energy function of the physical problem.

In fact, the loss function Eq. (3) has its origin in the variational approaches in statistical mechanics [61,63,64]. To see this, we write

$$\mathcal{L} + \ln Z = \mathbb{KL}\left(q(\boldsymbol{x}) \| \frac{\pi(\boldsymbol{x})}{Z}\right) \geq 0, \tag{4}$$

where the Kullback-Leibler (KL) divergence measures the proximity between the model and the target probability densities [50,64]. Equation (4) reaches zero only when the two distributions are identical. One thus concludes that the loss Eq. (3) provides a variational upper bound of the physical free energy of the system, $-\ln Z$.

For the actual optimization of the loss function, we randomly draw a batch of latent variables according to the prior probability $p(\boldsymbol{z})$ and pass them through the generator network $\boldsymbol{x} = g(\boldsymbol{z})$, an unbiased estimator of the loss Eq. (3) is

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\ln p(\boldsymbol{z}) - \ln\left|\det\left(\frac{\partial g(\boldsymbol{z})}{\partial \boldsymbol{z}}\right)\right| - \ln \pi(g(\boldsymbol{z}))\right], \tag{5}$$

where the log-Jacobian determinant can be efficiently computed by summing the contributions of each bijector. Notice that in Eq. (5) all the network parameters are inside the expectation, which amounts to the reparametrization trick [50]. We perform stochastic optimization of Eq. (5) [65], in which the gradients with respect to the model parameters are computed efficiently using backpropagation. The gradient of Eq. (5) is the same as the one of the KL divergence Eq. (4) since the intractable partition function Z is independent of the model parameter.

Since the KL divergence is asymmetric, the PDD is different from the maximum likelihood estimation (MLE) which amounts to minimizing the empirical approximation of the KL divergence in an opposite direction $\mathbb{KL}(\pi(\boldsymbol{x})/Z \| q(\boldsymbol{x}))$ [50,64]. The most significant difference is that in PDD one does not rely on an additional way (such as efficient MCMC) to collect independent and identically distributed configurations of the physical problem for training. Moreover, optimizing the variational objectivity Eq. (5) can be more efficient than MLE because one directly makes use of the analytical functional form and gradient information of the target density $\pi(\boldsymbol{x})$. Finally, in the variational calculation, it is always better to achieve a lower value of the training loss Eq. (5) without the concern of overfitting [41].

The variational approach can also be integrated seamlessly with the MCMC sampling to produce unbiased physical results with enhanced efficiency. The partition function of the physical problem can be expressed in terms of the latent variables

$$Z = \int d\boldsymbol{z}\pi(g(\boldsymbol{z}))\left|\det\left(\frac{\partial g(\boldsymbol{z})}{\partial \boldsymbol{z}}\right)\right| = \int d\boldsymbol{z} p(\boldsymbol{z})\left[\frac{\pi(g(\boldsymbol{z}))}{q(g(\boldsymbol{z}))}\right], \tag{6}$$

where the first equality simply invokes *change of variables* from the physical space $\boldsymbol{x}$ to the latent space $\boldsymbol{z}$ using the learned normalizing flow, and the second equality rearranges terms using Eq. (1).

The integrand of Eq. (6) offers direct access to the renormalized energy function in the latent space induced by the flow $\boldsymbol{z} = g^{-1}(\boldsymbol{x})$. One sees that when the model density $q(\boldsymbol{x})$ perfectly matches the target density $\pi(\boldsymbol{x})/Z$, the energy function of the latent variables reduces to one

associated with the prior $p(z)$. The variational calculation Eq. (4) would then always push the latent distribution towards the independent Gaussian prior. Therefore, it would be advantageous to perform Metropolis [42] or hybrid Monte Carlo (HMC) sampling [43] in the latent space for better mixing. Given samples in the latent space, one can obtain the corresponding physical variable via $\boldsymbol{x} = g(\boldsymbol{z})$. This generalizes the Monte Carlo updates in the wavelet basis [66,67] to the case of adaptively latent space for a given physical problem.

As a demonstration, we apply NeuralRG to the two-dimensional Ising model, a prototypical model in statistical physics. To conform with the continuous requirement of the physical variables, we employ the continuous relaxations trick of Refs. [68,69]. We first decouple the Ising spins using a Gaussian integral, then sum over the Ising spins to obtain a target probability density

$$\pi(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\boldsymbol{x}^T(K+\alpha I)^{-1}\boldsymbol{x}\right)\prod_{i=1}^{N}\cosh(x_i), \quad (7)$$

where K is an $N \times N$ symmetric matrix, I is an identity matrix, and α is a constant offset such that $K + \alpha I$ is positive definite [70]. For each of the configurations, one can directly sample the discrete Ising variables $\boldsymbol{s} = \{\pm 1\}^{\otimes N}$ according to $\pi(\boldsymbol{s}|\boldsymbol{x}) = \prod_i (1 + e^{-2s_i x_i})^{-1}$. It is straightforward to verify that the marginal probability distribution $\int d\boldsymbol{x}\pi(\boldsymbol{s}|\boldsymbol{x})\pi(\boldsymbol{x}) \propto \exp\left(\frac{1}{2}\boldsymbol{s}^T K \boldsymbol{s}\right) \equiv \pi_{\text{Ising}}(\boldsymbol{s})$ restores the Boltzmann weight of the Ising model with the coupling matrix K. Therefore, Eq. (7) can be viewed as a dual version of the Ising model, in which the continuous variables $\boldsymbol{x}$ represent the field couple to the Ising spins. We choose K to describe the two-dimensional critical Ising model on a square lattice critical with periodic boundary condition.

We train the NeuralRG network of the structure shown schematically in Fig. 1(a) where the bijectors are of the size 2×2, as shown in Fig. 2(d). The results in Fig. 3(a) shows that the variational free-energy continuously decreases during the training. In this case, the exact lower bound reads $-\ln Z = -\ln Z_{\text{Ising}} - \frac{1}{2}\ln\det(K + \alpha I) + (N/2)[\ln(2/\pi) - \alpha]$, where $Z_{\text{Ising}} = \sum_{\boldsymbol{s}} \pi_{\text{Ising}}(\boldsymbol{s})$ is known from the exact solution of the Ising model [71] on the finite periodic lattice [72]. We show results obtained in a wider temperature range and generated samples in the Supplemental Material [41].

To make use of the learned normalizing flow, we perform the hybrid Monte Carlo (HMC) [41] sampling in the latent space in parallel to the training using the effective energy function Eq. (6). The physical results quickly converge to the correct value indicated by the solid red line. In comparison, the HMC simulation in the original physical space using Eq. (7) as the energy function fails to thermalize during the same HMC steps. Even taking into account the overhead of training and evaluating the neural network, sampling in the latent space is still significantly more efficient.

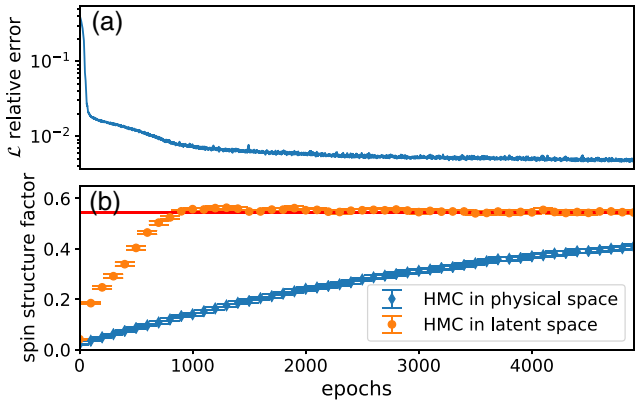

FIG. 3. Physical results obtained for the continuous field theory of Eq. (7) equivalent to the Ising model on a $N = 16 \times 16$ lattice at critical coupling. (a) The relative error in the variational free energy Eq. (3) decreases with training epochs. The exact free energy is obtained from the analytical solution of the Ising model [71,72]. (b) Uniform spin structure factor computed using hybrid Monte Carlo sampling in the latent and the physical spaces, respectively. The error bars are computed using independent batch of samples. The solid red line is the result of $\mathbb{E}_{\boldsymbol{s}\sim\pi_{\text{Ising}}(\boldsymbol{s})}[\sum_{i,j} s_i s_j / N^2]$ computed directly for the Ising model.

To reveal the physical meaning of the learned latent variables, we recall the wavelets interpretation of the RG [73–75]. In our context, if each bijector performs the same linear transformation, the network precisely implements the discrete wavelet transformation [76]. Using the wavelets language, the bijectors at each layer extract "smooth" and "detail" components of the input signal separately. And the bijectors in the next layer perform transformations only to these smooth components.

We probe the response of the latent variables by computing the gradient of the transformation $\boldsymbol{z} = g^{-1}(\boldsymbol{x})$ using backpropagation through the network. Figure 4(a) visualizes the expected gradient $\mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x})}[\partial z_i/\partial \boldsymbol{x}]$ averaged over a batch of physical samples, where z_i are the four top-level collective variables connecting to all of the physical variables. Each of them responds similarly to a nonoverlapping spatial region, which is indeed a reminiscence of the wavelets. On the other hand, the gradient $\partial z_i/\partial \boldsymbol{x}$ also exhibits variation for different physical variables. The variation is an indication of the *nonlinearity* of the learned transformation since, otherwise, the gradient is independent of data in the ordinary linear wavelet transformation. Thus, the latent variables can be regarded as a nonlinear and adaptive learned generalization of the wavelet representation. Employing more advanced feature visualization and interpretability tools in deep learning [77,78] may help distill more useful information from the trained neural network.

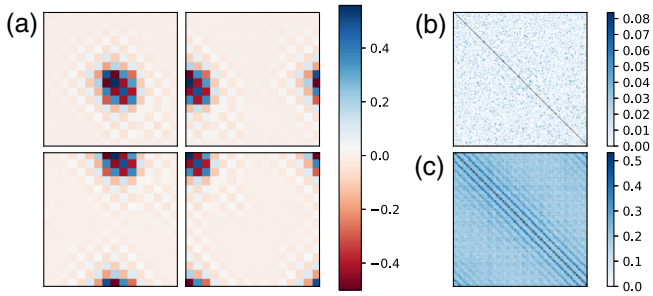

FIG. 4. (a) The responses of the latent space collective variables with respect to the physical variables $\mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x})}[\partial z_i/\partial \boldsymbol{x}]$. (b) Mutual information between the latent variables and (c) the physical variables. Note different scales in the color bars of (b) and (c).

Finally, to characterize the effective interactions in the latent space, we plot estimated MI [79] between the latent variables in Fig. 4(b). The network does not map the physical distribution into ideally factorized Gaussian prior, in line with the gap to the exact free energy Fig. 3(a). However, the remaining MI between the latent variables is much smaller compared to the ones between the physical variables shown in Fig. 4(c). Obtaining a mutually independent representation of the original problem underlines the efficiency boost of the latent space HMC demonstrated in Fig. 3(b). Adaptive learning of a *nonlinear* transformation is a distinct feature of the present approach compared to *linear* independent component analysis and wavelet transformations. These linear transformation approaches would not be able to remove dependence between the physical variables unless the physical problem is a free theory.

The NeuralRG approach provides an automatic way to identify mutually independent collective variables [80,81]. Note that the identified collective variables do not need to be the same as the ones in the conventional RG. This significant difference is due to the conventional approach focusing on identifying the fixed points under the iterative application of the same predetermined transformation to the physical variables (e.g., block decimation or momentum shell integration). While the present approach aims at finding out a set of hierarchical transformations that map complex physical probability densities to the predetermined prior distribution. Thus, its application is particularly relevant to off-lattice molecular simulations that involve a large number of continuous degrees of freedom which are often very difficult to simulate.

Last, the conventional RG is a semigroup since the process is irreversible. However, the NeuralRG networks built on normalizing flows form a group, which can be useful for exploring the information preserving RG [25,74,82] in conjunction with holographic mapping.

The authors thank Yang Qi, Yi-Zhuang You, Pan Zhang, Jin-Guo Liu, Lei-Han Tang, Chao Tang, Lu Yu, Long Zhang, Guang-Ming Zhang, and Ye-Hua Liu for discussions and encouragement. We thank Wei Tang for providing the exact free energy value of the 2D Ising model on finite lattices. The work is supported by the Ministry of Science and Technology of China under the Grant No. 2016YFA0300603 and the National Natural Science Foundation of China under Grant No. 11774398.

*wanglei@iphy.ac.cn

[1] M. Gell-Mann and F. E. Low, Quantum electrodynamics at small distances, Phys. Rev. **95**, 1300 (1954).
[2] K. G. Wilson, Renormalization group and critical phenomena. I. Renormalization group and the Kadanoff scaling picture, Phys. Rev. B **4**, 3174 (1971).
[3] K. G. Wilson, Renormalization group and critical phenomena. II. Phase-space cell analysis of critical behavior, Phys. Rev. B **4**, 3184 (1971).
[4] K. G. Wilson, The renormalization group: Critical phenomena and the Kondo problem, Rev. Mod. Phys. **47**, 773 (1975).
[5] R. H. Swendsen, Monte Carlo Renormalization Group, Phys. Rev. Lett. **42**, 859 (1979).
[6] G. Vidal, A Class of Quantum Many-Body States that can be Efficiently Simulated, Phys. Rev. Lett. **101**, 110501 (2008).
[7] M. Levin and C. P. Nave, Tensor Renormalization Group Approach to Two-Dimensional Classical Lattice Models, Phys. Rev. Lett. **99**, 120601 (2007).
[8] Z.-C. Gu, M. Levin, and X.-G. Wen, Tensor-entanglement renormalization group approach as a unified method for symmetry breaking and topological phase transitions, Phys. Rev. B **78**, 205116 (2008).
[9] Z.-C. Gu and X.-G. Wen, Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order, Phys. Rev. B **80**, 155131 (2009).
[10] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Second Renormalization of Tensor-Network States, Phys. Rev. Lett. **103**, 160601 (2009).
[11] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang, Renormalization of tensor-network states, Phys. Rev. B **81**, 174411 (2010).
[12] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Coarse-graining renormalization by higher-order singular value decomposition, Phys. Rev. B **86**, 045139 (2012).
[13] E. Efrati, Z. Wang, A. Kolan, and L. P. Kadanoff, Real-space renormalization in statistical mechanics, Rev. Mod. Phys. **86**, 647 (2014).
[14] G. Evenbly and G. Vidal, Tensor Network Renormalization, Phys. Rev. Lett. **115**, 180405 (2015).
[15] G. Evenbly and G. Vidal, Tensor Network Renormalization Yields the Multiscale Entanglement Renormalization Ansatz, Phys. Rev. Lett. **115**, 200401 (2015).
[16] S. Yang, Z.-C. Gu, and X.-G. Wen, Loop Optimization for Tensor Network Renormalization, Phys. Rev. Lett. **118**, 110504 (2017).
[17] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, Renormalization Group Flows of Hamiltonians Using Tensor Networks, Phys. Rev. Lett. **118**, 250602 (2017).

[18] M. Hauru, C. Delcamp, and S. Mizera, Renormalization of tensor networks using graph-independent local truncations, Phys. Rev. B **97**, 045111 (2018).

[19] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, in *Proceedings of the European Conference on Computer Vision* (Springer, New York, 2014), pp. 818–833.

[20] C. Bény, Deep learning and the renormalization group, arXiv:1301.3124.

[21] P. Mehta and D. J. Schwab, An exact mapping between the variational renormalization group and deep learning, arXiv:1410.3831.

[22] S. Bradde and W. Bialek, PCA meets RG, J. Stat. Phys. **167**, 462 (2017).

[23] S. Iso, S. Shiba, and S. Yokoo, Scale-invariant feature extraction of neural network and renormalization group flow, Phys. Rev. E **97**, 053304 (2018).

[24] M. Koch-Janusz and Z. Ringel, Mutual information, neural networks and the renormalization group, Nat. Phys. **14**, 578 (2018).

[25] Y.-Z. You, Z. Yang, and X.-L. Qi, Machine learning spatial geometry from entanglement features, Phys. Rev. B **97**, 045153 (2018).

[26] R. Kenway, Vulnerability of deep learning, arXiv:1803.06111.

[27] H. W. Lin, M. Tegmark, and D. Rolnick, Why does deep and cheap learning work so well?, J. Stat. Phys. **168**, 1223 (2017).

[28] D. J. Schwab and P. Mehta, Comment on "Why does deep and cheap learning work so well?" [arXiv:1608.08225]; arXiv:1609.03541.

[29] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, arXiv:1410.8516.

[30] M. Germain, K. Gregor, I. Murray, and H. Larochelle, MADE: Masked autoencoder for distribution estimation, arXiv:1502.03509.

[31] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, arXiv:1505.05770.

[32] L. Dinh, J. Sohl-Dickstein, and S. Bengio, Density estimation using real NVP, arXiv:1605.08803.

[33] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, Improving variational inference with inverse autoregressive flow, arXiv:1606.04934.

[34] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, Pixel recurrent neural networks, arXiv:1601.06759.

[35] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, WaveNet: A generative model for raw audio, arXiv:1609.03499.

[36] G. Papamakarios, T. Pavlakou, and I. Murray, Masked autoregressive flow for density estimation, arXiv:1705.07057.

[37] D. P. Kingma and P. Dhariwal, Glow: Generative flow with invertible 1×1 convolutions, arXiv:1807.03039.

[38] A. van den Oord *et al.*, Parallel WaveNet: Fast high-fidelity speech synthesis, arXiv:1711.10433.

[39] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, TensorFlow distributions, arXiv:1711.10604.

[40] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, Pyro: Deep universal probabilistic programming, arXiv:1810.09538.

[41] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.260601 for details of the training and sampling algorithms, the implementation of the real NVP bijectors, and ways to exploit the symmetry of the physical problem in the variational calculation, which includes Refs. [29–36,42–49].

[42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[43] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, Phys. Lett. B **195**, 216 (1987).

[44] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), arXiv:1511.07289.

[45] R. M. Neal, MCMC using Hamiltonian dynamics, *Handbook of Markov Chain Monte Carlo* (Chapman Hall/CRC, 2011), Vol. 2.

[46] J. Song, S. Zhao, and S. Ermon, A-NICE-MC: Adversarial training for MCMC, arXiv:1706.07561.

[47] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein, Generalizing Hamiltonian Monte Carlo with neural networks, arXiv:1711.09268.

[48] D. A. Moore, Symmetrized variational inference, in *NIPS Workshop on Advances in Approximate Bayesian Inference* (2016), http://approximateinference.org/accepted/Moore2016.pdf.

[49] Y. LeCun, C. Cortes, and C. J. C. Burges, MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[51] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, arXiv:1603.04467.

[52] A. Paszke, G. Chanan, Z. Lin, S. Gross, E. Yang, L. Antiga, and Z. Devito, Automatic differentiation in PyTorch, in *NIPS 2017 Workshop Autodiff* (2017).

[53] If needed, one can even share weights in the depth direction due to scale invariance emerged at criticality. The scale invariant reduces the number of parameters to be independent of the system size. In this case, one can iterate the training process for increasingly larger system size and reuse the weights from the previous step as the initial value.

[54] G. Vidal, Efficient Classical Simulation of Slightly Entangled Quantum Computations, Phys. Rev. Lett. **91**, 147902 (2003).

[55] Y. Y. Shi, L. M. Duan, and G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, Phys. Rev. A **74**, 022320 (2006).

[56] T. Barthel, M. Kliesch, and J. Eisert, Real-Space Renormalization Yields Finite Correlations, Phys. Rev. Lett. **105**, 010502 (2010).

[57] M. M. Wolf, F. Verstraete, M. B. Hastings, and J. Ignacio Cirac, Area Laws in Quantum Systems: Mutual Information and Correlations, Phys. Rev. Lett. **100**, 070502 (2008).

[58] Z. Liu, S. P. Rodrigues, and W. Cai, Simulating the Ising model with a deep convolutional generative adversarial network, arXiv:1710.04987.

[59] L. Huang and L. Wang, Accelerated Monte Carlo simulations with restricted Boltzmann machines, Phys. Rev. B **95**, 035105 (2017).

[60] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning Monte Carlo method, Phys. Rev. B **95**, 041101 (2017).
[61] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, Cambridge, England, 2005).
[62] J. S. Liu, *Monte Carlo Strategies in Scientific Computing* (Springer, New York, 2001).
[63] R. P. Feynman, *Statistical Mechanics: A Set of Lectures* (W. A. Benjamin, Inc., New York, 1972).
[64] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
[65] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[66] A. E. Ismail, G. C. Rutledge, and G. Stephanopoulos, Multiresolution analysis in statistical mechanics. I. Using wavelets to calculate thermodynamic properties, J. Chem. Phys. **118**, 4414 (2003).
[67] A. E. Ismail, G. Stephanopoulos, and G. C. Rutledge, Multiresolution analysis in statistical mechanics. II. The wavelet transform as a basis for Monte Carlo simulations on lattices, J. Chem. Phys. **118**, 4424 (2003).
[68] M. E. Fisher, Scaling, universality, and renormalization group theory, in *Critical Phenomena*, edited by F. J. W. Hahne (Springer-Verlag, Berlin, 1983).
[69] Y. Zhang, C. Sutton, and A. Storkey, Continuous relaxations for discrete Hamiltonian Monte Carlo, Adv. Neural Inf. Process. Syst. **25**, 3194 (2012).
[70] We choose α such that the lowest eigenvalue of $K + \alpha I$ equals to 0.1.
[71] L. Onsager, Crystal statistics. I. A two-dimensional model with an order-disorder transition, Phys. Rev. **65**, 117 (1944).
[72] B. Kaufman, Crystal statistics. II. Partition function evaluated by spinor analysis, Phys. Rev. **76**, 1232 (1949).
[73] B. Guy, *Wavelets and Renormalization* (World Scientific, Singapore, 1999), Vol. 10.
[74] X.-L. Qi, Exact holographic mapping and emergent space-time geometry, arXiv:1309.6282.
[75] G. Evenbly and S. R. White, Entanglement Renormalization and Wavelets, Phys. Rev. Lett. **116**, 140403 (2016).
[76] G. Evenbly and S. R. White, Representation and design of wavelets using unitary circuits, Phys. Rev. A **97**, 052314 (2018).
[77] C. Olah, A. Mordvintsev, and L. Schubert, Feature visualization, Distill, 2017, https://distill.pub/2017/feature-visualization.
[78] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, The building blocks of interpretability, Distill, 2018, https://distill.pub/2018/building-blocks.
[79] A. Kraskov, H. Stögbauer, and P. Grassberger, Estimating mutual information, Phys. Rev. E **69**, 066138 (2004).
[80] A. Barducci, M. Bonomi, and M. Parrinello, Metadynamics, Comput. Mol. Sci. **1**, 826 (2011).
[81] M. Invernizzi, O. Valsson, and M. Parrinello, Coarse graining from variationally enhanced sampling applied to the Ginzburg-Landau model, Proc. Natl. Acad. Sci. U.S.A. **114**, 3370 (2017).
[82] B. Swingle, Entanglement renormalization and holography, Phys. Rev. D **86**, 065007 (2012).