

Neural Network Renormalization Group

Shuo-Hui Li^{1,2} and Lei Wang^{1,*}

¹*Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*

²*University of Chinese Academy of Sciences, Beijing 100049, China*

We present a variational renormalization group (RG) approach using a deep generative model based on normalizing flows. The model performs hierarchical change-of-variables transformations from the physical space to a latent space with reduced mutual information. Conversely, the neural net 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.

Renormalization group (RG) is one of the central schemes in theoretical physics, whose broad 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 calculation using tensor network machinery [6–17]

The relevance of RG also 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 deep neural network extracts a hierarchy of increasingly higher-level concepts in its deeper layers [18]. In light of such intriguing similarities, Refs [19–22] drew connections between deep learning and RG, Ref. [23] proposed an RG scheme based on mutual information maximization, Ref. [24] employed deep learning to study holography duality, Ref. [25] examined the adversarial examples from an RG perspective. Since the discussions are not totally uncontroversial [20, 22, 23, 26, 27], it remains highly desirable to establish a more concrete, rigorous, and constructive connection between RG and deep learning. Such 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 paper, 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 deep neural network. Integrating latest advances in deep learning including *Normalizing Flows* [28–35] and *Probability Density Distillation* [36], and tensor network architectures, in particular, the multi-scale 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

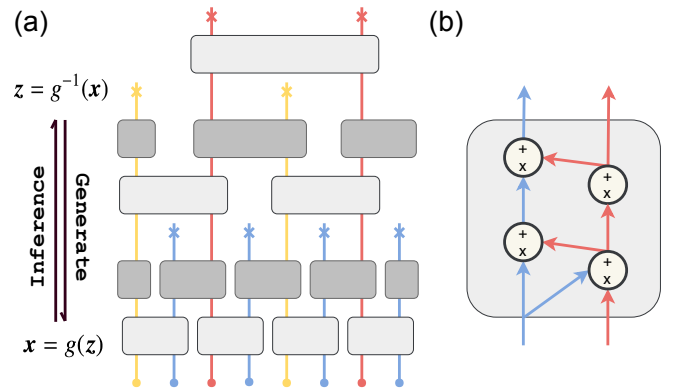


Figure 1. (a) The NeuralRG network is formed by stacking bijector networks into a hierarchical structure. The solid dots at the bottom are the physical variables x and the crosses are the latent variables z . Each block is a bijective and differentiable transformation parametrized by a bijector neural network. The light gray and the dark gray blocks are the disentangled and the decimators respectively. The RG flows from bottom to top, which corresponds to inference the latent variables conditioned on the physical variables. Conversely, by sampling the latent variables according to the prior distribution and passing them downwards one can generate the physical configuration directly. (b) The internal structure of the bijector block consists of a real-valued non-volume preserving flow [31].

original proposal based on Bayesian net [19] than more recent discussions on Boltzmann Machines [20, 22] and Principal Component Analysis [21].

Figure 1(a) shows the proposed neural net architecture. Each building block is a diffeomorphism, i.e., a bijective and differentiable function parametrized by a neural network, denoted by a bijector [37, 38]. Figure 1(b) illustrates a possible realization of the bijector using the real-valued non-volume preserving flow (Real NVP) [31] [39], which is one of the simplest invertible neural networks with efficiently computable Jacobian determinants known as Normalizing Flows [28–35].

The neural network relates the physical variables x and the latent variables z via a differentiable bijective map $x = g(z)$.

Their probability densities are also related [40]

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

where $q(\mathbf{x})$ is the normalized probability density of the physical variables. And $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{1})$ 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 of the bijective transformation. 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 the building blocks is still a bijector. This motivates the modular design of the network structure 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, $\mathbf{z} = g^{-1}(\mathbf{x})$. 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 bijective property is crucial for learning the RG flow in a controlled way. No matter how complex is the hierarchical transformations performed by the neural network, one can efficiently compute the normalized probability density $q(\mathbf{x})$ for either given or sampled physical configuration \mathbf{x} by keeping track of the Jacobian determinant of each bijector locally. One can let the bijectors in the same layer share weights due to the translational invariances of the physical problem [41].

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, thanks to the modularity, arbitrary complex NeuralRG architecture can be trained efficiently using standard differentiable programming tools offered in modern deep learning frameworks [42, 43].

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 [44]. Since each disentangler block connects only a few neighboring variables, the causal light cone of the physical variables

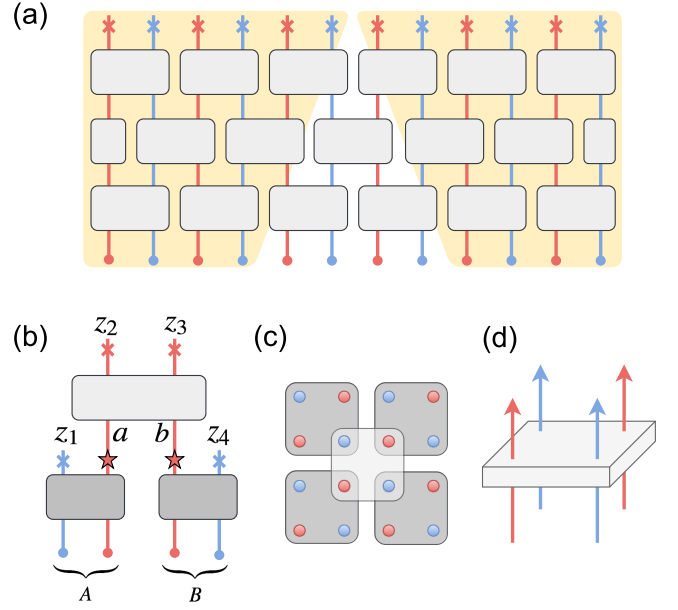


Figure 2. (a) A reference neural network architecture with only disentanglers. The physical variables in the two shaded regions are uncorrelated because their causal light cones do not overlap in the latent space. (b) Mutual information is conserved at the decimation step, see Eq. (2). (c) The arrangement of the bijectors in the two-dimensional space. (d) Each bijector acts on four variables. Disentanglers reduces mutual information between variables. While for decimators, only one of its outputs is passed on to the next layer and the others are treated directly as latent 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 [45]. For example, the mutual information (MI) between the variables at each decimation step shown in Fig. 2(b) follows

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

The first equality is due to that the MI is invariant under invertible transformation of variables within each group. While the second equality is due to the random variables z_1 and z_4 are 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 neural net of the tree structure. This does not necessarily impose a constraint on the expressibility of the network since the MI between two continuous variables can be arbitrarily large in principle. However, a neural network with decimators only could be limited in practice since it is rather unphysical to carry the MI between two extensive regions with only two variables [46].

It is straightforward to generalize the NeuralRG architec-

ture 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 of the outputs is passed on to the next layer. In a network with only disentangled, the depth should scale with the linear system size to capture diverging correlation length at criticality. While the required depth only scales logarithmically with the linear system size if one employs the MERA-like structure. Note that different from the tensor network modeling of quantum states [47], the MERA-like architecture is sufficient to model classical systems with short-range interactions even at criticality since they exhibit the MI area law [48].

Building the neural network using normalizing flows provides a generative model with explicit and tractable likelihoods Eq. (1) compared to previous studies [20, 22, 23, 49–51]. 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 access to the bare energy function, i.e. the *unnormalized* probability density $\pi(\mathbf{x})$ of a physical problem, direct sampling of the physical configurations is generally difficult due to the intractable partition function $Z = \int d\mathbf{x} \pi(\mathbf{x})$ [52]. The standard Markov chain Monte Carlo (MCMC) approach suffers from the slow mixing problem in many cases [53].

We train the NeuralRG network by minimizing the *Probability Density Distillation* (PDD) loss

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

which was recently employed by DeepMind to train the Parallel WaveNet [36]. The first term of the loss is the negative entropy of the model density $q(\mathbf{x})$, which favors diversity in its samples. While the second term corresponds to the expected energy since $-\ln \pi(\mathbf{x})$ is precisely the energy function of the target problem.

In fact, the loss function Eq. (3) has its origin in the variational approaches in statistical mechanics [52, 54, 55]. To see this, we write

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

where the Kullback-Leibler (KL) divergence measures the proximity between the model and the target probability densities [40, 55]. 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 latent variables according to the prior probability $p(\mathbf{z})$ and pass them through the generator network $\mathbf{x} = g(\mathbf{z})$, an unbiased estimator of the loss Eq. (3) is

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

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 but not in the sampling process, which amounts to the *reparametrization trick* [40]. We perform stochastic optimization of Eq. (5) [56], 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* $\mathbb{KL}\left(\frac{\pi(\mathbf{x})}{Z} \parallel q(\mathbf{x})\right)$ [40, 55]. 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 configuration 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 functional form and gradient information of the target density $\pi(\mathbf{x})$. 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 [39].

The variational approach can also be integrated seamlessly with 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

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

where the first equality states that the normalizing flow performs a learnable *change-of-variables* from the physical space \mathbf{x} to the latent space \mathbf{z} , and the second equality employs Eq. (1).

The integrand of Eq. (6) offers direct access to the renormalized energy function in the latent space. One sees that when the model density $q(\mathbf{x})$ perfectly matches the target density $\pi(\mathbf{x})/Z$, the energy function of the latent variables reduces to one of the prior distribution. The variational calculation would always push the distribution of the latent variables towards the independent Gaussian prior. Therefore, it would be advantageous to perform Metropolis [57] or hybrid Monte Carlo (HMC) sampling [58] in the latent space for better mixing. One can obtain the corresponding physical variable via $\mathbf{x} = g(\mathbf{z})$ for a given latent variable. This generalizes the Monte Carlo updates in the wavelet basis [59, 60] to the case of adaptively latent space for each 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. [63, 64]. We first decouple the Ising spins using a Gaussian integral, then sum over the Ising spins to obtain a target

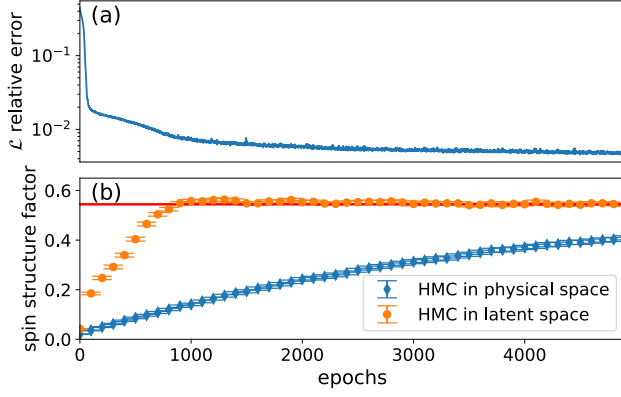


Figure 3. Physical results obtained for the continuous field theory of Eq. (7) equivalent to the Ising model on a $N = 16 \times 16$ lattice at critical coupling. (a) The relative error in the variational free energy Eq. (3) decreases with training epochs. The exact free energy is obtained from the analytical solution of the Ising model [61, 62]. (b) Uniform spin structure factor computed using hybrid Monte Carlo sampling in the latent and the physical spaces respectively. The errorbars are computed using independent batch of samples. The solid red line is the result of $\mathbb{E}_{s \sim \pi_{\text{Ising}}(s)} [\sum_{i,j} s_i s_j / N^2]$ computed directly for the Ising model.

probability density

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

where K is an $N \times N$ symmetric matrix, I is an identity matrix and α is constant offset such that $K + \alpha I$ is positive definite [65]. For each of the configuration, one can directly sample the discrete Ising variables $\mathbf{s} = \{\pm 1\}^{\otimes N}$ according to $\pi(\mathbf{s}|\mathbf{x}) = \prod_i (1 + e^{-2s_i x_i})^{-1}$. It is straightforward to verify that the marginal probability distribution $\int d\mathbf{x} \pi(\mathbf{s}|\mathbf{x}) \pi(\mathbf{x}) \propto \exp(\frac{1}{2}\mathbf{s}^T K \mathbf{s}) \equiv \pi_{\text{Ising}}(\mathbf{s})$ restores the Boltzmann weight of the Ising model with the coupling matrix K [66]. Therefore, Equation (7) can be viewed as a dual version of the Ising model, in which the continuous variables \mathbf{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 $-\ln Z = -\ln Z_{\text{Ising}} - \frac{1}{2} \ln \det(K + \alpha I) + \frac{N}{2} [\ln(2/\pi) - \alpha]$, where $Z_{\text{Ising}} = \sum_{\mathbf{s}} \pi_{\text{Ising}}(\mathbf{s})$ is known from the exact solution of the Ising model [61] on finite periodic lattice [62].

We perform the hybrid Monte Carlo (HMC) [39] 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

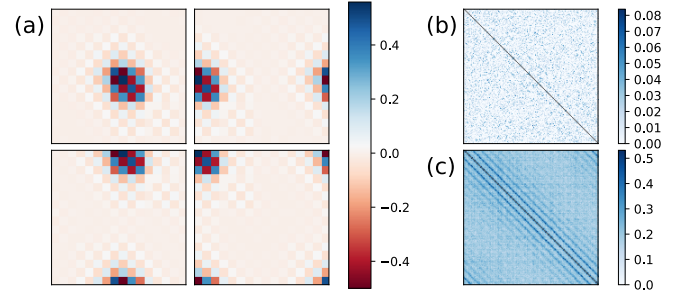


Figure 4. (a) The responses of the latent space collective variables with respect to the physical variables $\mathbb{E}_{\mathbf{x} \sim \pi(\mathbf{x})} [\partial z_i / \partial \mathbf{x}]$. (b) Mutual information between the latent variables and (c) the physical variables. Note different scales in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

same update steps. Even taking into account of the overhead of training and evaluating the neural network, sampling in the latent space is still significantly more efficient since one deals with nearly Gaussian distributed independent latent variables in the HMC sampling.

To reveal the physical meaning of the learned latent variables, we recall the wavelets interpretation of RG [67–69]. In our context, if each bijector performs the same linear transformation, the network precisely implements the discrete wavelet transformation [70]. 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 $\mathbf{z} = g^{-1}(\mathbf{x})$ using back-propagation through the network. Figure 4(a) visualizes the expected gradient $\mathbb{E}_{\mathbf{x} \sim \pi(\mathbf{x})} [\partial z_i / \partial \mathbf{x}]$ 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 responses similarly to a non-overlapping spatial region, which is indeed a reminiscence of the wavelets. On the other hand, the gradient $\partial z_i / \partial \mathbf{x}$ also exhibits variation for different physical variables. The variation is an indication of the *nonlinearity* of the learned transformation since otherwise the gradient is fixed in the ordinary linear wavelets transformation. Thus, the latent variables can be regarded as nonlinear and adaptive learned generalization of the wavelets representation. Employing more advanced feature visualization and interpretability tools in deep learning [71, 72] may help distill more useful information from the trained neural network.

Finally, to characterize the effective interactions in the latent space, we plot estimated MI [73] between the latent variables in Fig. 4(b). The network does not map the physical distribution into ideally factorized Gaussian prior in accordance to the gap in the variational 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 in

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. The linear transformations would not be able to remove dependence between the physical variables unless the physical problem is a free theory.

The minimalist implementation of NeuralRG [74] can be further improved in several aspects. First, one could even use a different prior density instead of the uninformative Gaussian distribution so that the RG does not always flow towards the infinite temperature fixed point. Second, the MERA inspired network structure can nevertheless be generalized by following considerations in tensor network architecture design [75, 76]. Lastly, one can improve each bijector using more expressive normalizing flows [29, 30, 32–35]. Since the size of each block is independent of the system size in NeuralRG, one may even employ more general bijectors [77] compared to the current choices in deep learning. Any of this improvement is likely to further to improve the variational upper bound of Fig. 3(a). Currently, the proposed NeuralRG framework is limited to problems with continuous variables since it relies on the probability transformation Eq. (1) and the differential learnability of the bijectors. Employing ideas from deep learning to generalize the approach to discrete variables is an interesting direction [78, 79].

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 that the conventional approach focuses on identifying the fixed points under 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 which map complex physical probability densities to the predetermined prior distribution. Thus, its application is particularly relevant to off-lattice molecular simulations which involve a large number of continuous degrees of freedom which are often very difficult to simulate. We focused on physical systems with translational invariance in this paper, where it makes sense to use a predetermined homogenous architecture. For physical systems with disorders [82, 83] or realistic dataset in machine learning, it would be interesting to learn the network structure based on the mutual information pattern of the problem [84, 85].

Besides calling a revived attention to the probabilistic [86] and information theory [87] perspectives on the RG flow, the NeuralRG framework also possesses mathematical properties underline modern understanding of RG in terms of diffeomorphism [88, 89]. 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 [24, 68] in conjunction with holographic mapping.

Despite the similarity between the disentangled to the con-

volutional kernel and decimators to the pooling layers, the proposed NeuralRG architecture Fig. 1(a) is different from the convolutional neural network. A crucial difference is that each bijector performs *nonlinear bijective* transformation instead of linear transformations. We believe that it is the large-scale structural similarities between MERA and dilated convolutions [33, 34, 36] and factor out layers [31] used in modern deep generative models underline their successes in modeling quantum states and classical data. It is interesting to compare the performance of the NeuralRG architecture to conventional deep learning models in machine learning tasks [39] and study its general implications for neural network architecture design.

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,” *Physical Review* **95**, 1300–1312 (1954).
- [2] Kenneth G. Wilson, “Renormalization Group and Critical Phenomena. I. Renormalization Group and the Kadanoff Scaling Picture,” *Physical Review B* **4**, 3174 (1971).
- [3] Kenneth G. Wilson, “Renormalization Group and Critical Phenomena. II. Phase-Space Cell Analysis of Critical Behavior,” *Physical Review B* **4**, 3184 (1971).
- [4] Kenneth G. Wilson, “The renormalization group: Critical phenomena and the Kondo problem,” *Reviews of Modern Physics* **47**, 773–840 (1975).
- [5] Robert H. Swendsen, “Monte Carlo Renormalization Group,” *Physical Review Letters* **42**, 859 (1979).
- [6] G. Vidal, “A class of quantum many-body states that can be efficiently simulated,” *Physical Review Letters* **101**, 110501 (2008), [arXiv:0610099](https://arxiv.org/abs/0610099).
- [7] Michael Levin and Cody P. Nave, “Tensor renormalization group approach to two-dimensional classical lattice models,” *Physical Review Letters* **99**, 120601 (2007), [arXiv:0611687](https://arxiv.org/abs/0611687).
- [8] Zheng-Cheng Gu, Michael Levin, and Xiao-Gang Wen, “Tensor-entanglement renormalization group approach as a unified method for symmetry breaking and topological phase transitions,” *Physical Review B* **78**, 205116 (2008), [arXiv:0807.2010](https://arxiv.org/abs/0807.2010).
- [9] Zheng-Cheng Gu and Xiao-Gang Wen, “Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order,” *Physical Review B* **80**, 155131 (2009), [arXiv:0903.1069](https://arxiv.org/abs/0903.1069).
- [10] Z. Y. Xie, H.C. Jiang, Q.N. Chen, Z. Y. Weng, and T. Xiang, “Second Renormalization of Tensor-Network States,” *Physical Review Letters* **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,” *Physical*

- Review B **81**, 174411 (2010), [arXiv:1002.1405](#).
- [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,” *Physical Review B* **86**, 045139 (2012), [arXiv:1201.1144](#).
 - [13] Efi Efrati, Zhe Wang, Amy Kolan, and Leo P. Kadanoff, “Real-space renormalization in statistical mechanics,” *Reviews of Modern Physics* **86**, 647–667 (2014), [arXiv:1301.6323v1](#).
 - [14] G. Evenbly and G. Vidal, “Tensor Network Renormalization,” *Physical Review Letters* **115**, 180405 (2015), [arXiv:1412.0732](#).
 - [15] G. Evenbly and G. Vidal, “Tensor Network Renormalization Yields the Multiscale Entanglement Renormalization Ansatz,” *Physical Review Letters* **115**, 200401 (2015), [arXiv:1502.05385](#).
 - [16] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, “Renormalization Group Flows of Hamiltonians Using Tensor Networks,” *Physical Review Letters* **118**, 250602 (2017), [arXiv:1703.00365](#).
 - [17] Markus Hauru, Clement Delcamp, and Sebastian Mizera, “Renormalization of tensor networks using graph-independent local truncations,” *Phys. Rev. B* **97**, 045111 (2018).
 - [18] Matthew D Zeiler and Rob Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision* (Springer, 2014) pp. 818–833.
 - [19] Cédric Bény, “Deep learning and the renormalization group,” [arXiv \(2013\)](#), [arXiv:1301.3124](#).
 - [20] Pankaj Mehta and David J. Schwab, “An exact mapping between the Variational Renormalization Group and Deep Learning,” [arXiv \(2014\)](#), [arXiv:1410.3831](#).
 - [21] Serena Bradde and William Bialek, “PCA Meets RG,” *Journal of Statistical Physics* **167**, 462–475 (2017), [arXiv:1610.09733](#).
 - [22] Satoshi Iso, Shotaro Shiba, and Sumito Yokoo, “Scale-invariant feature extraction of neural network and renormalization group flow,” *Phys. Rev. E* **97**, 053304 (2018).
 - [23] Maciej Koch-Janusz and Zohar Ringel, “Mutual Information, Neural Networks and the Renormalization Group,” *Nature Physics* **14**, 578 (2018), [1704.06279](#).
 - [24] Yi-Zhuang You, Zhao Yang, and Xiao-Liang Qi, “Machine learning spatial geometry from entanglement features,” *Phys. Rev. B* **97**, 045153 (2018).
 - [25] R. Kenway, “Vulnerability of Deep Learning,” *ArXiv e-prints* (2018), [arXiv:1803.06111 \[stat.ML\]](#).
 - [26] Henry W. Lin, Max Tegmark, and David Rolnick, “Why Does Deep and Cheap Learning Work So Well?” *Journal of Statistical Physics* **168**, 1223–1247 (2017), [arXiv:1608.08225](#).
 - [27] David J. Schwab and Pankaj Mehta, “Comment on “Why does deep and cheap learning work so well?” [arXiv:1608.08225],” [arXiv \(2016\)](#), [arXiv:1609.03541](#).
 - [28] Laurent Dinh, David Krueger, and Yoshua Bengio, “NICE: Non-linear Independent Components Estimation,” [arXiv \(2014\)](#), [1410.8516](#), [arXiv:1410.8516](#).
 - [29] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle, “MADE: Masked Autoencoder for Distribution Estimation,” [arXiv \(2015\)](#), [arXiv:1502.03509](#).
 - [30] Danilo Jimenez Rezende and Shakir Mohamed, “Variational Inference with Normalizing Flows,” [arXiv \(2015\)](#), [arXiv:1505.05770](#).
 - [31] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, “Density estimation using Real NVP,” [arXiv \(2016\)](#), [1605.08803](#), [arXiv:1605.08803](#).
 - [32] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling, “Improving Variational Inference with Inverse Autoregressive Flow,” [arXiv \(2016\)](#), [arXiv:1606.04934](#).
 - [33] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu, “Pixel Recurrent Neural Networks,” in *International Conference on Machine Learning I(CML)*, Vol. 48 (2016) pp. 1747–1756, [arXiv:1601.06759](#).
 - [34] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” [arXiv \(2016\)](#), [arXiv:1609.03499](#).
 - [35] George Papamakarios, Theo Pavlakou, and Iain Murray, “Masked Autoregressive Flow for Density Estimation,” [arXiv \(2017\)](#), [arXiv:1705.07057](#).
 - [36] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, “Parallel WaveNet: Fast High-Fidelity Speech Synthesis,” [arXiv \(2017\)](#), [arXiv:1711.10433](#).
 - [37] Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous, “TensorFlow Distributions,” [arXiv \(2017\)](#), [arXiv:1711.10604](#).
 - [38] Pyro Developers, “Pyro,” (2017).
 - [39] See the Supplementary Materials 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.
 - [40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning* (MIT Press, 2016).
 - [41] 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.
 - [42] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” [arXiv \(2016\)](#), [arXiv:1603.04467](#).
 - [43] Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito, “Automatic differentiation in PyTorch,” in *NIPS 2017 Workshop Autodiff* (2017).
 - [44] Guifre Vidal, “Efficient classical simulation of slightly entangled quantum computations,” *Physical Review Letters* **91**, 147902 (2003), [arXiv:0301063](#).
 - [45] Y. Y. Shi, L. M. Duan, and G. Vidal, “Classical simulation of quantum many-body systems with a tree tensor network,” *Physical Review A* **74**, 022320 (2006), [arXiv:0511070](#).
 - [46] The decimator only tree structure NeuralRG network is sufficient to model one dimensional physical systems with short-range interactions since the MI is constant [48].
 - [47] Thomas Barthel, Martin Kliesch, and Jens Eisert, “Real-space renormalization yields finite correlations,” *Physical Re-*

- view *Letters* **105**, 010502 (2010), [arXiv:1003.2319](#).
- [48] Michael M. Wolf, Frank Verstraete, Matthew B. Hastings, and J. Ignacio Cirac, “Area laws in quantum systems: Mutual information and correlations,” *Physical Review Letters* **100**, 070502 (2008), [arXiv:0704.3906](#).
- [49] Zhaocheng Liu, Sean P. Rodrigues, and Wenshan Cai, “Simulating the Ising Model with a Deep Convolutional Generative Adversarial Network,” *arXiv* (2017), [arXiv:1710.04987](#).
- [50] Li Huang and Lei Wang, “Accelerated Monte Carlo simulations with restricted Boltzmann machines,” *Physical Review B* **95**, 035105 (2017), [arXiv:1610.02746](#).
- [51] Junwei Liu, Yang Qi, Zi Yang Meng, and Liang Fu, “Self-learning Monte Carlo method,” *Physical Review B* **95**, 041101 (2017), [arXiv:1610.03137](#).
- [52] David J C MacKay, *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, 2005).
- [53] Jun S. Liu, *Monte Carlo Strategies in Scientific Computing* (Springer, 2001).
- [54] R. P. Feynman, *Statistical Mechanics: A Set of Lectures* (W. A. Benjamin, Inc., 1972).
- [55] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
- [56] Diederik P Kingma and Jimmy Lei Ba, “Adam: A Method for Stochastic Optimization,” *arXiv* (2014), [arXiv:1412.6980v9](#).
- [57] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller, “Equation of state calculations by fast computing machines,” *Journal Chemical Physics* **21**, 1087–1092 (1953), [arXiv:5744249209](#).
- [58] Simon Duane, A D Kennedy, Brian J Pendleton, and Duncan Roweth, “Hybrid Monte Carlo,” *Physics Letters B* **195**, 216–222 (1987).
- [59] Ahmed E Ismail, Gregory C Rutledge, and George Stephanopoulos, “Multiresolution analysis in statistical mechanics. i. using wavelets to calculate thermodynamic properties,” *The Journal of chemical physics* **118**, 4414–4423 (2003).
- [60] Ahmed E Ismail, George Stephanopoulos, and Gregory C Rutledge, “Multiresolution analysis in statistical mechanics. ii. the wavelet transform as a basis for monte carlo simulations on lattices,” *The Journal of chemical physics* **118**, 4424–4431 (2003).
- [61] Lars Onsager, “Crystal statistics. I. A two-dimensional model with an order-disorder transition,” *Physical Review* **65**, 117–149 (1944).
- [62] Bruria Kaufman, “Crystal Statistics. II. Partition Function Evaluated by Spinor Analysis,” *Phys. Rev.* **76**, 1232–1243 (1949).
- [63] M. E. Fisher, “Scaling, universality, and renormalization group theory,” in *Critical Phenomena*, edited by F. J. W. Hahne (Springer-Verlag, Berlin, 1983).
- [64] Yichuan Zhang, Charles Sutton, and Amos Storkey, “Continuous Relaxations for Discrete Hamiltonian Monte Carlo,” *Advances in Neural Information Processing Systems* **25**, 3194–3202 (2012).
- [65] We choose α such that the lowest eigenvalue of $K + \alpha I$ equals to 0.1.
- [66] Note that Eq. (7) is not a unique. By diagonalizing $(K + \alpha I)$ one obtains a model where each continuous variable couples to one Fourier component of the original Ising spins [63, 64]. Performing NeuralRG on such problem would correspond to the momentum space RG.
- [67] Battle Guy, *Wavelets and renormalization*, Vol. 10 (World Scientific, 1999).
- [68] Xiao-Liang Qi, “Exact holographic mapping and emergent space-time geometry,” *arXiv* (2013), [arXiv:1309.6282](#).
- [69] Glen Evenbly and Steven R. White, “Entanglement renormalization and wavelets,” *Phys. Rev. Lett.* **116**, 140403 (2016).
- [70] Glen Evenbly and Steven R. White, “Representation and design of wavelets using unitary circuits,” *Phys. Rev. A* **97**, 052314 (2018).
- [71] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert, “Feature visualization,” *Distill* (2017), 10.23915/distill.00007, <https://distill.pub/2017/feature-visualization>.
- [72] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev, “The building blocks of interpretability,” *Distill* (2018), 10.23915/distill.00010, <https://distill.pub/2018/building-blocks>.
- [73] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger, “Estimating mutual information,” *Phys. Rev. E* **69**, 066138 (2004).
- [74] See <https://github.com/li012589/NeuralRG> for a PyTorch implementation of the code.
- [75] L. Tagliacozzo, G. Evenbly, and G. Vidal, “Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law,” *Physical Review B* **80**, 235127 (2009), [arXiv:0903.5017](#).
- [76] G. Evenbly and G. Vidal, “Class of highly entangled many-body states that can be efficiently simulated,” *Physical Review Letters* **112**, 240502 (2014), [arXiv:1210.1895](#).
- [77] Scott Shaobing Chen and Ramesh A. Gopinath, “Gaussianization,” *Advances in Neural Information Processing Systems* **13**, 423–429 (2001).
- [78] Eric Jang, Shixiang Gu, and Ben Poole, “Categorical Reparameterization with Gumbel-Softmax,” *arXiv* (2016), [arXiv:1611.01144](#).
- [79] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh, “The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables,” *arXiv* (2016), [arXiv:1611.00712](#).
- [80] Alessandro Barducci, Massimiliano Bonomi, and Michele Parrinello, “Metadynamics,” *Wiley Interdisciplinary Reviews: Computational Molecular Science* **1**, 826–843 (2011).
- [81] Michele Invernizzi, Omar Valsson, and Michele Parrinello, “Coarse graining from variationally enhanced sampling applied to the Ginzburg-Landau model,” *Proceedings of the National Academy of Sciences* **114**, 3370–3374 (2017).
- [82] Shang Keng Ma, Chandan Dasgupta, and Chin Hun Hu, “Random antiferromagnetic chain,” *Physical Review Letters* **43**, 1434–1437 (1979).
- [83] Daniel S. Fisher, “Random transverse field Ising spin chains,” *Physical Review Letters* **69**, 534–537 (1992).
- [84] C. K. Chow and C. N. Liu, “Approximating Discrete Probability Distributions with Dependence Trees,” *IEEE Transactions on Information Theory* **14**, 462–467 (1968).
- [85] Katharine Hyatt, James R. Garrison, and Bela Bauer, “Extracting Entanglement Geometry from Quantum States,” *Physical Review Letters* **119**, 140502 (2017), [arXiv:1704.01974](#).
- [86] G. Jona-Lasinio, “The renormalization group: A probabilistic view,” *Il Nuovo Cimento B Series* **11**, 26, 99–119 (1975).
- [87] S. M. Apenko, “Information theory and renormalization group flows,” *Physica A: Statistical Mechanics and its Applications* **391**, 62–77 (2012), [arXiv:0910.2097](#).
- [88] Michael Lässig, “Geometry of the renormalization group with an application in two dimensions,” *Nuclear Physics B* **334**, 652–668 (1990).
- [89] Ariel Caticha, “Changes of variables and the renormalization group,” (2016), [arXiv:1605.06366](#).
- [90] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” *arXiv* (2015), [arXiv:1511.07289](#).

- [91] Radford M Neal, “Mcmc using hamiltonian dynamics,” *Handbook of Markov Chain Monte Carlo* **2** (2011).
- [92] Jiaming Song, Shengjia Zhao, and Stefano Ermon, “A-NICE-MC: Adversarial Training for MCMC,” [arXiv \(2017\)](#), [arXiv:1706.07561](#).
- [93] Daniel Levy, Matthew D. Hoffman, and Jascha Sohl-Dickstein, “Generalizing Hamiltonian Monte Carlo with Neural Networks,” [arXiv \(2017\)](#), [arXiv:1711.09268](#).
- [94] David A Moore, “Symmetrized variational inference,” in *NIPS Workshop on Advances in Approximate Bayesian Inference* (2016).

Supplemental Materials: Neural Network Renormalization Group

Training Algorithm

Algorithm 1 shows the variational training procedure of the NeuralRG network.

Algorithm 1 Variational training algorithm of the NeuralRG network

Require: Normalized prior probability density $p(\mathbf{z})$, e.g. a Normal distribution

Require: Unnormalized target probability density $\pi(\mathbf{x})$

Ensure: A normalizing flow neural network $\mathbf{x}=\mathbf{g}(\mathbf{z})$ with normalized probability density $q(\mathbf{x})$

Initialize a normalizing flow \mathbf{g}

while Stop Criterion Not Met **do**

 Sample a batch of latent variables \mathbf{z} according to the prior $p(\mathbf{z})$

 Obtain physical variables $\mathbf{x}=\mathbf{g}(\mathbf{z})$ and compute their densities $q(\mathbf{x})$

▷ Eq. (1)

$\text{loss} = \text{mean}\{\ln[q(\mathbf{x})] - \ln[\pi(\mathbf{x})]\}$

▷ Eq. (5)

 Optimization step for the loss

end while

Details about the Real NVP Bijector

To implement the bijector we use the real-valued non-volume preserving (Real NVP) net [31], which belongs to a general class of bijective neural networks with tractable Jacobian determinant [28–35]. Real NVP is a generative model with explicit and tractable likelihood. One can efficiently evaluate the model probability density $q(\mathbf{x})$ for any sample, either given externally or generated by the network itself. This feature is important for integrating with an unbiased Metropolis sampler.

The Real NVP block divides the inputs into two groups $\mathbf{z} = \mathbf{z}_< \cup \mathbf{z}_>$, and updates only one of them with information of another group

$$\begin{cases} \mathbf{x}_< = \mathbf{z}_<, \\ \mathbf{x}_> = \mathbf{z}_> \odot e^{s(\mathbf{z}_<)} + t(\mathbf{z}_<), \end{cases} \quad (\text{S1})$$

where $\mathbf{x} = \mathbf{x}_< \cup \mathbf{x}_>$ is the output. $s(\cdot)$ and $t(\cdot)$ are two arbitrary functions parametrized by neural networks. In our implementation, we use multilayer perceptrons with 64 hidden neurons of exponential linear activation [90]. The output activation of the scaling s -function is a tanh with learnable scale. While the output of the translation t -function is a linear function. The \odot symbol denotes element-wise product. The transformation Eq. (S1) is easy to invert by reversing the basic arithmetical operations. Moreover, the transformation has a triangular Jacobian matrix, whose determinant can be computed efficiently by summing over each component of the outputs of the scaling function $\ln \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = \sum_i [s(\mathbf{z}_<)]_i$. The transformation Eq. (S1) can be iterated so that each group of variables is updated. In our implementation, we update each of the two groups four times in an alternating order. The log-Jacobian determinant of the bijector block is computed by summing up contributions of each layer. Within each layer, we use the same block with shared parameters. The log-Jacobian determinant is computed by summing up contributions of each block.

Hybrid Monte Carlo in the latent space

Hybrid Monte Carlo (HMC) [58] is a powerful sampling approach widely adopted in physics and machine learning [91]. HMC reduces the diffusive behavior of the traditional Metropolis updates [57] via exploiting the Hamiltonian dynamics of continuous variables. Further acceleration of the HMC using neural networks is an active research direction in deep learning [92, 93].

For our application, we can either perform the HMC sampling of the physical variables given the normalized probability distribution or in the latent space. In the latter case, a key step of the HMC is the integration of the equation-of-motion according to effective energy function Eq. (6). Note that the auto-differentiation tool in deep learning package conveniently provides tools to compute the force, i.e., the gradient of the energy with respect to the variables. Algorithm 2 outlines the key steps of the HMC simulation.

Algorithm 2 Hybrid Monte Carlo simulation in the latent space

Require: Energy function of the latent variables $U(\mathbf{z}) = -\ln[p(\mathbf{z})] - \ln[\pi(g(\mathbf{z}))] + \ln[q(g(\mathbf{z}))]$ ▷ Eq. (6)
 Initial state of the latent variable \mathbf{z}
while Stop Criterion Not Met **do**
 Sample velocity \mathbf{v} from a Normal distribution
 Leapfrog integration using the energy function $U(\mathbf{z})$
 Metropolis acceptance according to the change of total energy $\mathbf{v}^T \mathbf{v} / 2 + U(\mathbf{z})$
end while
 Obtain the physical variable $\mathbf{x} = g(\mathbf{z})$ and estimate physical observables

Symmetrized variational calculation

One can further incorporate physical symmetries of the target problem into the variational scheme. As a concrete example, we discuss the implementation for the discrete inversion symmetry of the problem $\pi(\mathbf{x}) = \pi(-\mathbf{x})$. For general discussions, please refer to [94].

We introduce the symmetrized variational density

$$q_{\text{sym}}(\mathbf{x}) = \frac{1}{2} [q(\mathbf{x}) + q(-\mathbf{x})], \quad (\text{S2})$$

where $q(\mathbf{x})$ is given by the normalizing flow network. To evaluate $q(-\mathbf{x})$ we will first need to compute $\mathbf{z} = g^{-1}(-\mathbf{x})$, and then use Eq. (1) of the main text. The density q_{sym} defined in this way manifestly respect the inversion symmetry. The training loss of symmetrized model reads

$$\mathcal{L}_{\text{sym}} = \int d\mathbf{x} q_{\text{sym}}(\mathbf{x}) [\ln q_{\text{sym}}(\mathbf{x}) - \ln \pi(\mathbf{x})] \quad (\text{S3})$$

$$= \int d\mathbf{x} q(\mathbf{x}) [\ln q_{\text{sym}}(\mathbf{x}) - \ln \pi(\mathbf{x})], \quad (\text{S4})$$

where for the second equality we used the fact the expression in the square bracket is symmetric respect to inversion. Thus, the practical overhead of symmetrized calculation is merely evaluating $q_{\text{sym}}(\mathbf{x})$ instead of $q(\mathbf{x})$. Compared to Eq. (3) in the main text, the loss of the symmetrized model would always be lower since $\mathcal{L} - \mathcal{L}_{\text{sym}} = \mathbb{KL}(q(\mathbf{x}) \parallel q_{\text{sym}}(\mathbf{x})) \geq 0$. Therefore, one can achieve better variational free energy by exploiting the symmetry of the physical problem.

Writing the symmetrized density Eq. (S2) as a mixture model $q_{\text{sym}}(\mathbf{x}) = \frac{1}{2} \sum_{\eta=\pm} q(\eta\mathbf{x})$, we can treat the sign variable η on the equal footing with the latent variable \mathbf{z} . In this regards, the generation process is deterministic given both \mathbf{z} and η , i.e. $p(\mathbf{x}|\mathbf{z}, \eta) = \delta(\mathbf{x} - \eta g(\mathbf{z}))$. Further marginalizing over the random sign η , one obtains the conditional probability $p(\mathbf{x}|\mathbf{z}) = \frac{1}{2} \sum_{\eta} \delta(\mathbf{x} - \eta g(\mathbf{z}))$, which amounts to randomly flip the sign of the outcome of the normalizing flow network. Lastly, marginalizing over \mathbf{z} in the joint probability $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, one obtains the symmetric density Eq. (S2) as a consistency check.

For inferencing the latent variable given the physical variable, we compute the posterior using the Bayes' rule,

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_{\text{sym}}(\mathbf{x})} = \frac{1}{2q_{\text{sym}}(\mathbf{x})} \sum_{\eta=\pm} q(\eta\mathbf{x}) \delta(\mathbf{z} - g^{-1}(\eta\mathbf{x})). \quad (\text{S5})$$

Note that the two choices of the sign are weighted by $q(\eta\mathbf{x})$ in the posterior. In Fig. 4(a) the latent vector is inferred in this way, and the physical variables are flipped accordingly. Finally, the posterior also allows us to transform the physical probability density in the latent space

$$\int d\mathbf{x} p(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) = \frac{1}{2} \sum_{\eta=\pm} p(\mathbf{z}) \left[\frac{\pi(\eta g(\mathbf{z}))}{q_{\text{sym}}(\eta g(\mathbf{z}))} \right] \quad (\text{S6})$$

$$= p(\mathbf{z}) \left[\frac{\pi(g(\mathbf{z}))}{q_{\text{sym}}(g(\mathbf{z}))} \right]. \quad (\text{S7})$$

In the last equality, we used the symmetry condition of the target and the model densities. The resulting probability density is the same as Eq. (6) in the main text.