Critical temperature of the classical *XY* model via autoencoder latent space sampling

Brandon Willnecker and Mervlyn Moodley

School of Chemistry and Physics, University of KwaZulu-Natal, Westville Campus, Private Bag X54001, Durban, 4000, South Africa

(Received 19 June 2024; accepted 7 January 2025; published 21 January 2025)

The classical XY model has been consistently studied since it was introduced more than six decades ago. Of particular interest has been the two-dimensional spin model's exhibition of the Berezinskii-Kosterlitz-Thouless (BKT) transition. This topological phenomenon describes the transition from bound vortex-antivortex pairs at low temperatures to unpaired or isolated vortices and antivortices above some critical temperature. In this work we propose a machine learning based method to determine the emergence of this phase transition. Generating unique states can be difficult due to the $U(1)$ symmetry present. We introduce an auxiliary field (analogous to a vortex density field) corresponding to a given state in order to eliminate the unwanted symmetry. An autoencoder was used to map these auxiliary fields into a lower-dimensional latent space. Samples were taken from this latent space to determine the thermal average of the vortex density, which was then used to determine the critical temperature of the phase transition.

DOI: 10.1103/PhysRevE.111.015305

I. INTRODUCTION

Besides its application to a plethora of fields in the physical sciences [1], machine leaning based techniques have had a profound influence on the study of condensed matter systems [2]. Of importance in these systems is the characterization of phases of matter. Recently, Carrasquilla and Melko [3] used a simple supervised learning approach to identify phase transitions, and almost simultaneously, Wang [4] proposed unsupervised learning techniques for discovering phase transitions in many-body systems. In the latter's work, the order parameter and structure factor was used as indicators of phase transitions. Since then, there have appeared numerous papers on using machine learning to identify and classify phase transitions, including topological phase transitions [5–10] which proves to be more difficult since these are defined in terms of nonlocal properties.

There are several existing machine learning methods for studying the Berezinskii-Kosterlitz-Thouless (BKT) transition in the XY model. Zhang *et al.* [10] used supervised machine learning to determine the phase boundary. A fully connected neural network was trained on Markov chain Monte Carlo (MCMC) samples generated at temperatures before, near, and after an estimated critical temperature T_c. Once trained, this model was able to identify the transition temperature based on the switching in the models predictions. Ng and Yang [11] also used autoencoders in their study of the classical XY model as we did in this paper. However, they used the mean-square-error loss function as a measure of the disorder in the given system. Phase transition points (including first-order, second-order, and topological ones) could be detected by the peaks in the standard deviation of the loss function. Shiina *et al.* [12] adopted a similar technique to Ref. [10] but instead of using the spin configurations, they utilized long-range correlations, $g_i(r) = s_i s_{i+r}$, as the inputs to a fully connected neural network. Again, the switching in the predicted output was used to determine the transition temperature. Miyajima and Mochizuki [13] proposed two machine learning methods for the detection of phase transitions in Heisenberg, Ising, and XY-like models. They first used a supervised learning technique similar to Ref. [10] whereby inputs are labeled according to their phase. Once the neural network is trained, it can be sampled near the T_c point. This point can be determined once the neural network's output changes (i.e., a phase transition has been detected). The second method is a temperature prediction neural network. The input is a spin configuration $\vec{s} = (s_1, ..., s_{L\times L})$ for an $L \times L$ lattice and the output is a 200-dimensional dimensional vector, $\vec{o} = (o_1, ..., o_{200})$, where each entry o_n is a probability that the spin configuration is at temperature $T_n = n\Delta T = n0.01J$, where J is the coupling strength of the spin-spin interaction. The phase changes are detected by studying the distinct patterns in heat maps of the weights of the neural networks for $T < T_c$ and $T > T_c$. The change in the pattern indicates a change in the phase.

Instead of training a neural network to predict the phase of a given state, we propose a method to more efficiently sample the space of states. We train an autoencoder to map the large space of states to a lower-dimensional latent space. This latent space may be much smaller but each element still contains the required information to reconstruct the original state. We can then sample from this space in constant time to calculate certain quantities that show a phase transition has occurred.

The paper is organized as follows. In Sec. II we provide the details for how the autoencoder works and how the latent space samples are taken. In Sec. III we revise the classical XY lattice model and explain why it is advantageous to study the continuous analog, $\theta(x, y)$, with local $U(1)$ symmetry removed. This is done by introducing an auxiliary field $A(x, y)$ that is derived from the continuous $\theta(x, y)$ field. This field is analogous to the average energy of a spin and its neighbors. In Sec. IV we use the concept of vortex density to determine the temperature T_c after which the vortices in the XY model become unbounded. The conclusion follows in Sec. V.

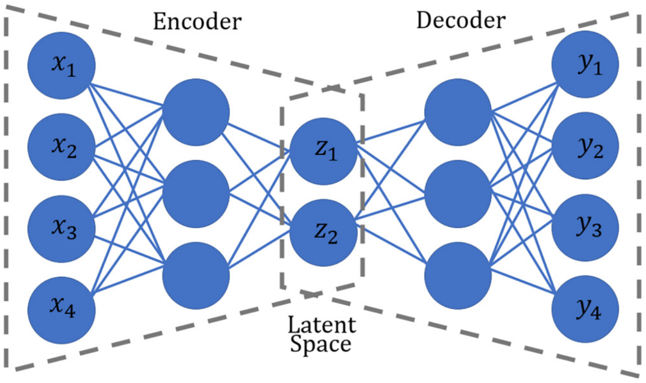

FIG. 1. The input vector (x_1, x_2, x_3, x_4) is compressed into a lower-dimensional latent space representation vector (z_1, z_2) by the encoder. The decoder then attempts to reconstruct the original vector, i.e., the vector norm $|x - y|$ is minimal for all input vectors x.

II. AUTOENCODERS

An autoencoder is a type of neural network that is used in supervised learning to provide efficient codes (compressed representation) to unlabeled input data [14,15]. An autoencoder is made of two sections called the encoder and decoder with a "bottle neck" in between, as can be seen in Fig. 1 below.

This can be described mathematically by the function

$$f : X \to Z, \tag{1}$$

which embeds the vectors from X into a lower-dimensional space Z, and the function

$$g : Z \to X, \tag{2}$$

which reverses the action of f. The aim is for the autoencoder to "learn" these functions such that

$$\forall x \in X, \ \ (g \circ f)(x) = x \tag{3}$$

The neural network is trained through back propagation in order to minimize a loss function. A typical loss function for an autoencoder is given by

$$L = \sum_{i=1}^{n} \frac{1}{2}(y_i - x_i)^2, \tag{4}$$

where x_i are the inputs and y_i are the reconstructed output [16,17]. This loss function ensures that the neural network correctly reconstructs the inputs by minimizing the difference between them. The narrowing of the network to a bottle neck is essential for the neural network to learn the required compression. The set of vectors produced from this compression Z is called the latent space. This latent space is a lower-dimensional representation of the input space.

III. THE MODEL

The classical XY model is a lattice model in which each site is occupied by a two-dimensional (2D) unit vector $\vec{s} = (\cos\theta, \sin\theta)$. The configuration, $S = \{\vec{s}_i\}$, is an assignment of each $\vec{s}_i$, or equivalently, an angle $\theta \in [-\pi, \pi]$ to each lattice site. The total energy of the configuration is given by the Hamiltonian

$$H = -\sum_{i \neq j} J_{ij}\vec{s}_i \cdot \vec{s}_j - \sum_i \vec{h}_i \cdot \vec{s}_i, \tag{5}$$

where J_{ij} is the strength of interaction between the ith and jth site, and $\vec{h}_i$ is a site-dependent external field. For our purposes, we will use a simplified version of the Hamiltonian, however it should be noted that the general case can be handled in a similar way. We will make three simplifying assumptions. Firstly, the strength of interaction will be taken as site independent. Secondly, we will not include an external field, and lastly, we will only consider nearest-neighbor interactions. Eqation (5) will therefore read

$$H = -J \sum_{<i,j>} \cos(\theta_i - \theta_j), \tag{6}$$

where summation is over nearest neighbors. The angles θ_i and θ_j are the angles of the vectors s_i and s_j, respectively. The Mermin-Wagner theorem [18] states that continuous symmetries cannot be spontaneously broken at finite temperature. The fact that this theorem does not apply to discrete symmetries was seen previously in the 2D Ising model. Since the XY model has a continuous symmetry ($\theta_i \to \theta_i + \delta\theta$), we do not expect a typical phase transition. Instead, we see a topological phase transition known as the Berezinskii-Kosterlitz-Thouless transition [19–22]. This transition can be studied by first taking the continuum limit of the lattice model. The continuum Hamiltonian is given by

$$H(\theta) = \int \frac{J}{2}(\nabla\theta)^2 dxdy, \tag{7}$$

where the field θ replaces the discrete angle assignments θ_i. The field configurations that give stationary H can be found using

$$\frac{\delta H}{\delta \theta} = 0 \Rightarrow \nabla^2\theta = 0, \tag{8}$$

which give two solutions. The first solution is the uninteresting ground state given by $\theta(x, y) =$ constant and the second, more interesting, solution involves the addition of vortices and antivortices which are topological defects in the θ field. These vortices are singular solutions to the equation

$$\nabla^2\theta = 0 \tag{9}$$

with

$$\oint_C \nabla\theta \cdot dl = 2\pi q, q \in \mathbb{Z}. \tag{10}$$

The integral is taken around a closed loop surrounding the singular point of the vortex. This integral gives an integer multiple of 2π because the net change in the spin vector must be some multiple of a full revolution. The integer q is the "charge" of the vortex or antivortex. Vortices have a charge $+1$ and antivortices have a charge of -1. Illustrations of these voticies and antivortices as they would appear on a lattice are shown in Fig. 2. To calculate the energy of a vortex, we first note that the angular symmetry of θ allows us to write $\theta = \theta(r)$. We can then use Eq. (10) to find $|\nabla\theta|$.

$$2\pi q = \oint_C \nabla\theta(r) \cdot dl = |\nabla\theta| 2\pi r \Rightarrow |\nabla\theta| = \frac{q}{r} \tag{11}$$

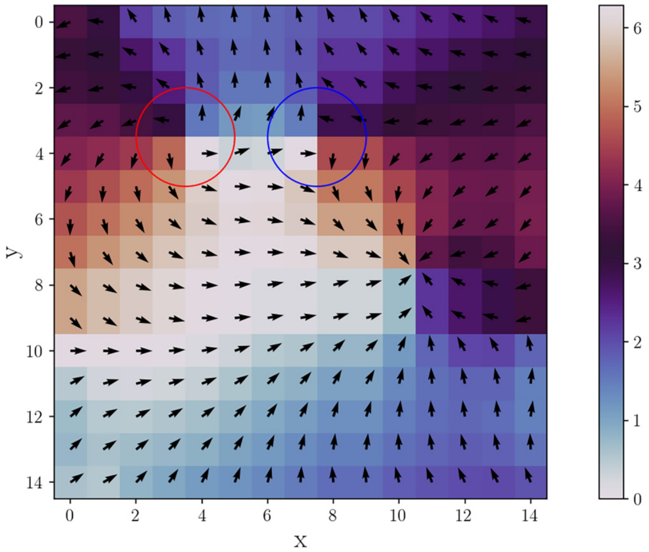

FIG. 2. Here we have examples of vortex (circled in red) and antivortex (circled in blue) configurations on the lattice. In the continuum limit, these become the singular solutions (topological defects) mentioned above. The background color shows the $\theta(x, y)$ field, i.e., the angle of the vector at the point (x, y).

Eqation (7) can be used to calculate the energy of a vortex or antivortex. This gives

$$E = \frac{J}{2}\int(\nabla\theta)^2 dxdy = Jq^2\pi\ \ln\left(\frac{L}{a}\right), \tag{12}$$

where L is length of the system and a is a lower cut-off value that can be taken as the lattice spacing from the original problem. This energy diverges in the thermodynamic limit so we do not have single-vortex or single-antivortex excitations. Instead, dipoles consisting of vortex and antivortex pairs can exist since they have finite energy. This is due to the fact that a closed loop surrounding the dipole contains no charge, $q_{\text{net}} = (+q) + (-q) = 0$.

As already stated, there is no spontaneous symmetry breaking at finite temperature, however, there is a transition between long-range correlations at low temperature and short-range correlations at high temperature. Kosterlitz and Thouless [21] showed that at low temperatures the vortices occur in tightly bound pairs. As the temperature increases past a transition point, $k_B T_{KT}/J \approx 0.893$, the pairs undergo deconfinement, which results in a change in the order parameter from a power-law to exponential.

IV. GENERATING AND SAMPLING THE LATENT SPACE

We investigate the unbinding phenomena by studying the density of these vortices (number of vortices per unit area) as a function of temperature. We expect that the vortex density is almost zero for low temperatures and then increases after the transition temperature. In order to calculate the thermodynamic average of the vortex density, we can generate samples from the associated Boltzmann distribution [23]. This can be rather computationally expensive. We instead use an autoencoder, illustrated in Fig. 3, to generate a lower-dimensional latent space from the full configuration space. We can then easily sample points from this latent space, pass these points through the decoder, and thus generate as many field configurations as we need.

However, more work needs to be done if we simply use $\theta(x, y)$ field configurations. This is because these fields have internal $U(1)$ symmetry which needs to be taken into account. We can either design a neural network that would respect this symmetry or we could overspecify the training data in order for the neural network to learn the symmetry from examples [24,25]. Instead of doing either of these, we propose the introduction of an auxiliary field $A(x, y)$ that removes the symmetry from the $\theta(x, y)$ fields. We define the auxiliary field $A(x, y)$, given $\theta(x, y)$, by

$$A(x, y) = \frac{1}{\sigma\sqrt{8\pi}}\int_D N(u, v) \times \{1 - \cos[\theta(x, y) - \theta(u, v)]\}dudv, \tag{13}$$

where $D(x, y)$ is a disk of radius 2σ centered at (x, y) and σ is a length on the order of the size of a vortex or antivortex. We found that $\sigma = 0.5$, which implies a disk radius of $r = 1.0$, was a good estimate of the vortex or antivortex size. This field quantifies the average variation of $\theta(x, y)$ around a local neighborhood of radius σ. $N(u, v)$ is a Gaussian centered at (x, y). This term ensures that only field values close to (x, y) are considered in the averaging process. The cosine term is analogous to the term in Eq. (6) and is used to remove the unwanted symmetry. This term also ensures that the field is bounded, which will be important when implementing the autoencoder. An example of this $\theta(x, y)$ field and its corresponding $A(x, y)$ field are illustrated in Fig. 4. Vortices and antivortices will have large field variations in their neighborhood and so will result in a large field value. Regions with no vortices or antivortices will have little to no variation in the field. This will result in a very small field value. We can thus characterize the vortex or antivortex density by the magnitude of $A(x, y)$ across the extent of the field. The autoencoder is then trained using these fields that are derived from $\theta(x, y)$ fields generated using a standard MCMC method, which is explained in the Appendix. Other configuration generation methods such as those by Swendsen and Wolff [26,27] can be used to generate the training data. However, as long as a sufficient amount of training samples is generated, the precious method of generation is not important. Once trained, the latent space was analyzed by passing in $A(x, y)$ fields and generating histograms from the produced latent space values for each dimension of the latent space. These histograms were then used to determine the mean and standard deviation of

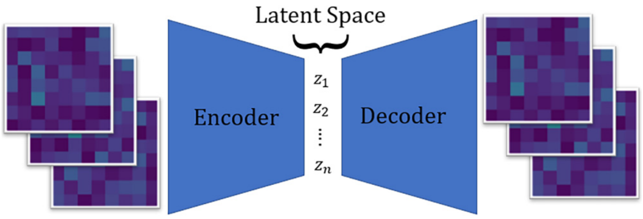

FIG. 3. The autoencoder was designed with a 60-dimensional latent space. The architecture and training details are elaborated on in the Appendix.

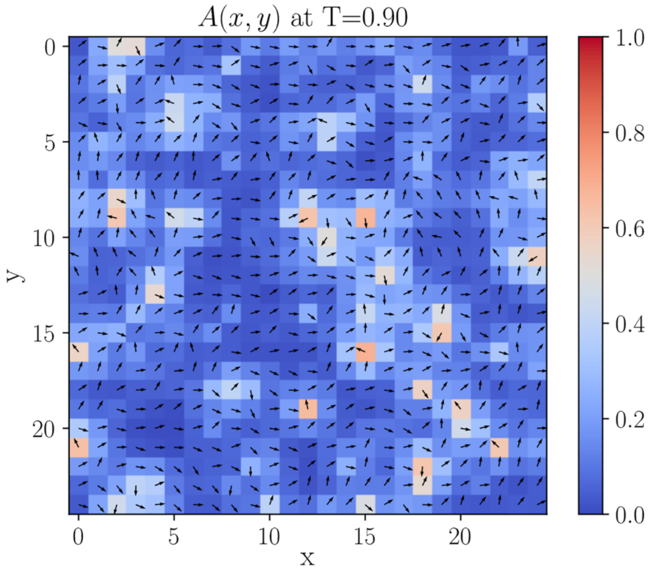

FIG. 4. This $\theta(x, y)$ field was generated using MCMC at $T = 0.9$. The background is the auxiliary $A(x, y)$ field calculated using Eq. (13).

the distribution for each latent space dimension. New fields can be generated by passing in a sampled latent space vectors, $z = (z_1, z_2, z_3, ..., z_{60})$, into the decoder portion of the autoencoder. Each z_i is sampled from the respective Gaussian with mean μ_i and standard deviation σ_i. The particular Gaussian distribution $N(\mu_1 = 0.14, \sigma_1 = 0.17)$ for the latent space dimension z_1 is shown in Fig. 5 below.

This method is similar to, but technically different from, a variational autoencoder [28]. Through the training process, an autoencoder learns to map a high-dimensional vector to a lower-dimensional latent space. The exact process is not known explicitly but is manifested in the learned parameters. On the other hand, a variational autoencoder first maps the input to a mean vector and a standard deviation vector of a predefined distribution. This distribution is then sampled to produce the latent space vector. The generation of these latent space distributions can be done using either method; however, it should be noted that you do not need to predefine a distribution over the latent space before training when using an autoencoder. The latent space sampling process is very computationally inexpensive compared to the standard MCMC algorithm. These sampled fields can then be used to calculate the thermodynamic average of the vortex density at a given temperature. We cannot directly count the number of vortices, so instead we use the average,

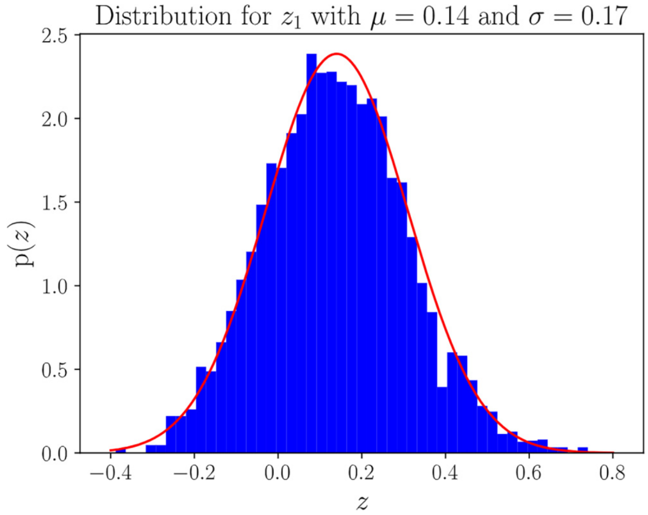

FIG. 5. Histogram of z_1 values sampled from the first latent space dimension. This was done at the temperature $T = 0.6$ for $A(x, y)$ fields of size $N = 50$. Similar distributions can be obtained for the other dimensions of the latent space as explained above for each temperature point in the range of interest.

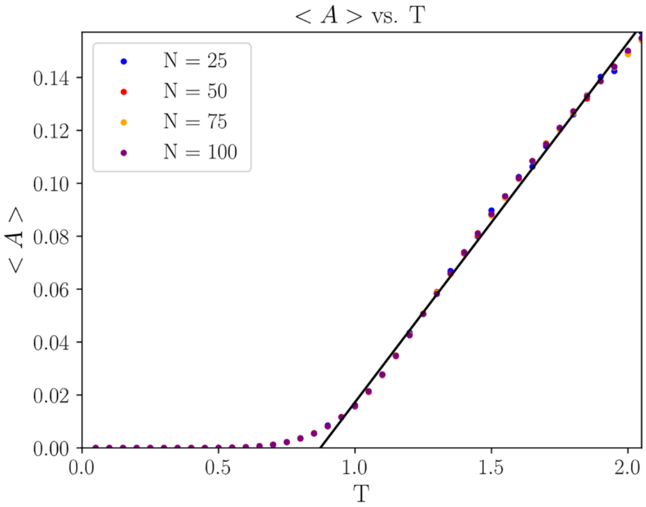

FIG. 6. We calculate $\langle A\rangle$ as a function of T over the temperature range $0 < T < 2$. This was done for field sizes $N = 25, 50, 75$, and 100. The straight line in black is a line of best fit.

$$\langle A\rangle = \frac{1}{L^2}\int_0^L\int_0^L A(x, y)dxdy, \tag{14}$$

as a proxy. $A(x, y)$ is analogous to the energy at and around the point (x, y). If each "vortex" has energy ϵ then $\frac{1}{\epsilon}\langle A\rangle = n$ gives the average number of unbound votices over the extent of the field. Ideally, we would expect the function $\langle A\rangle$ to be

$$\langle A\rangle = \begin{cases} 0 & \text{if } T < T_c \\ a(T - T_c) & \text{if } T \geqslant T_c, \end{cases} \tag{15}$$

where we have vortex unbinding above the critical temperature. The number of vortices, and hence vortex density, will then grow linearly with temperature. In practice, finite size effects and finite sampling effects will result in an approximate form of $\langle A\rangle$ as shown in Fig. 6. Note the approximate

TABLE I. Table of critical temperature estimates and associated errors for varying system sizes.

N	T_c Estimate	Error ΔT
25	0.872 833	0.020 067
50	0.872 213	0.020 687
75	0.871 779	0.021 121
100	0.872 756	0.02144

fitting to Eq. (15). We can use the intercept point of the line of best fit as an approximation for T_c. The results are summarized in Table 1.

V. CONCLUSION

It was shown that an autoencoder can be a useful tool in reducing a given configuration space to a lower-dimensional latent space that may be much easier to sample from. In this case, it was found that the latent space could be sampled using various Gaussian distributions. This latent space was sampled to calculate the thermal average of the vortex density, and from this one could determine the critical temperature (T_{KT}) at which this vortex density becomes nonzero. This method of latent space sampling is very general and so can be applied to many systems. The only requirement is that the system needs to contain a large enough amount of correlation between its constituents in order for the autoencoder to learn how to compress it with little loss. A large amount of correlation means that a large amount of redundant information can be removed during the compression stage of the autoencoder. With this in mind, one can extend this method to systems of large particles with solid-liquid phase transitions or systems with topological phase transitions [29] since one does not need any order parameters during the training process. Systems with very little to no correlations (like an ideal gas) cannot be mapped to a lower-dimensional space since the knowledge of the behavior of part of the system gives no information on the behavior of another part, i.e., the specification of the system cannot be reduced. Future work for this method includes extensions to other learnable features such as magnetization where a magnetization density vector field $M(x, y)$ is derived from the $\theta(x, y)$ field instead of the auxiliary $A(x, y)$ field, as done in this paper. Other work includes extensions to the generalized XY model, which includes fractional vorices. In that case, a new vortex counting method needs to be implemented.

ACKNOWLEDGMENT

We would like to acknowledge the insightful comments and suggestions made by the referees.

APPENDIX

The $\theta(x, y)$ and $A(x, y)$ fields were discretized into lattices of sizes N = 25, 50, 75, and 100. The size of the autoencoder input layer for each lattice is N^2. The size of each subsequent layer is $\frac{3}{4}$ of the previous layer in order to create the required bottleneck for the autoencoder. The size of the latent space was chosen to be 60. This choice was a good enough compromise between computational cost and reconstruction detail. Quadratic cost (MSE) was chosen as the cost function for its simplicity. The training was done for 5000 epochs with a batch size of 100 and a learning rate of 0.001. The sigmoid activation function was used for all layers. Standard MCMC methods were then used to generate the training data. For each temperature value, 10 000 samples were generated and used as training data. The full code (with comments) for this paper can be found at Ref. [30].

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[2] E. Bedolla, L. C. Padierna, and R. Castanneda-Priego, Machine learning for condensed matter physics, J. Phys.: Condens. Matter **33**, 053001 (2021).

[3] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[4] L. Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[5] Y. Zhang and E.-A. Kim, Quantum loop topography for machine learning, Phys. Rev. Lett. **118**, 216401 (2017).

[6] M. Richter-Laskowska, H. Khan, N. Trivedi, and M. M. Masa, A machine learning approach to the Berezinskii-Kosterlitz-Thouless transition in classical and quantum models, Condens. Matter Phys. **21**, 33602 (2018).

[7] J. F. Rodriguez-Nieva and M. S. Scheurer, Identifying topological order through unsupervised machine learning, Nat. Phys. **15**, 790 (2019).

[8] M. J. S. Beach, A. Golubeva, and R. G. Melko, Machine learning vortices at the Kosterlitz-Thouless transition, Phys. Rev. B **97**, 045207 (2018).

[9] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock and C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, Nat. Phys. **15**, 917 (2019).

[10] W. Zhang, J. Liu, and T. C. Wei, Machine learning of phase transitions in the percolation and *XY* models, Phys. Rev. E **99**, 032142 (2019).

[11] K. K. Ng and M. F. Yang, Unsupervised learning of phase transitions via modified anomaly detection with autoencoders, Phys. Rev. B **108**, 214428 (2023).

[12] K. Shiina, H. Mori, Y. Okabe and H. K. Lee, Machine-learning studies on spin models, Sci. Rep. **10**, 2177 (2020).

[13] Y. Miyajima and M. Mochizuki, Machine-learning detection of the Berezinskii-Kosterlitz-Thouless transition and the second-order phase transition in the XXZ models, Phys. Rev. B **107**, 134420 (2023).

[14] M. A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. **37**, 233 (1991).

[15] L. Theis, W. Shi, A. Cunningham, and F. Huszár, Lossy image compression with compressive autoencoders, arXiv:1703.00395.

[16] K. Janocha and W. M. Czarnecki, On loss functions for deep neural networks in classification, arXiv:1702.05659.

[17] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015).

[18] N. D. Mermin and H. Wagner, Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models, Phys. Rev. Lett. **17**, 1307 (1966).

[19] V. L. Berezinsky, Destruction of long range order in one-dimensional and two-dimensional systems having a continuous

symmetry group. I. Classical systems, Sov. Phys. JETP **32**, 493 (1971); [Zh. Eksp. Teor. Fiz. 59, 907 (1971)].

[20] V. L. Berezinskii, Destruction of long-range order in one-dimensional and two-dimensional systems possessing a continuous symmetry group. II. Quantum systems, Sov. Phys. JETP **34**, 610 (1972).

[21] J. M. Kosterlitz and D. J. Thouless, Ordering, metastability and phase transitions in two-dimensional systems, J. Phys. C **6**, 1181 (1973).

[22] J. M. Kosterlitz, The critical properties of the two-dimensional XY model, J. Phys. C **7**, 1046 (1974).

[23] X. Leoncini, A. D. Verga and S. Ruffo, Hamiltonian dynamics and the phase transition of the XY model, Phys. Rev. E **57**, 6377 (1998).

[24] B. Bloem-Reddy and Y. W. Teh, Probabilistic symmetries and invariant neural networks, J. Mach. Learn. Res. **21**, 1 (2020).

[25] J. Wood and J. Shawe-Taylor, Representation theory and invariant neural networks, Discrete Appl. Math. **69**, 33 (1996).

[26] R. H. Swendsen and J. S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett. **58**, 86 (1987).

[27] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett. **62**, 361 (1989).

[28] D. P. Kingma and M. Welling, *An Introduction to Variational Autoencoders* (Now Foundations and Trends, Boston, 2019).

[29] M. H. Zarei, Ising order parameter and topological phase transitions: Toric code in a uniform magnetic field, Phys. Rev. B **100**, 125159 (2019).

[30] github.com/BrandonWillnecker.