Group-equivariant autoencoder for identifying spontaneously broken symmetries

Devanshu Agrawal,[1] Adrian Del Maestro,[2,3,4] Steven Johnston,[2,4] and James Ostrowski[1]

[1]*Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, Tennessee 37996, USA*
[2]*Department of Physics and Astronomy, University of Tennessee, Knoxville, Tennessee 37996, USA*
[3]*Min H. Kao Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, Tennessee 37996, USA*
[4]*Institute for Advanced Materials and Manufacturing, University of Tennessee, Knoxville, Tennessee 37996, USA*

(Received 12 February 2022; revised 31 January 2023; accepted 4 April 2023; published 2 May 2023)

We introduce the group-equivariant autoencoder (GE autoencoder), a deep neural network (DNN) method that locates phase boundaries by determining which symmetries of the Hamiltonian have spontaneously broken at each temperature. We use group theory to deduce which symmetries of the system remain intact in all phases, and then use this information to constrain the parameters of the GE autoencoder such that the encoder learns an order parameter invariant to these "never-broken" symmetries. This procedure produces a dramatic reduction in the number of free parameters such that the GE-autoencoder size is independent of the system size. We include symmetry regularization terms in the loss function of the GE autoencoder so that the learned order parameter is also equivariant to the remaining symmetries of the system. By examining the group representation by which the learned order parameter transforms, we are then able to extract information about the associated spontaneous symmetry breaking. We test the GE autoencoder on the 2D classical ferromagnetic and antiferromagnetic Ising models, finding that the GE autoencoder (1) accurately determines which symmetries have spontaneously broken at each temperature; (2) estimates the critical temperature in the thermodynamic limit with greater accuracy, robustness, and time efficiency than a symmetry-agnostic baseline autoencoder; and (3) detects the presence of an external symmetry-breaking magnetic field with greater sensitivity than the baseline method. Finally, we describe various key implementation details, including a quadratic-programming-based method for extracting the critical temperature estimate from trained autoencoders and calculations of the DNN initialization and learning rate settings required for fair model comparisons.

DOI: 10.1103/PhysRevE.107.054104

I. INTRODUCTION

Mapping a material's phase diagram is an important endeavor in condensed-matter physics and materials science [1,2]. This is a crucial step towards practical applications as such diagrams can act as a road map to manipulating a material's functionality. From a theoretical perspective, predicting the locations of phase boundaries for a given material can provide important insights into the microscopic physics that govern its behavior and can provide crucial validation of proposed low-energy models.

In some cases (e.g., structural transitions), phase transitions are governed by high-energy processes and can be reasonably predicted with efficient computational methods [3] like density functional theory [4–8] or molecular dynamics simulations [9–11]. However, there are also numerous examples of phase transitions between novel states of matter that are completely governed by low-energy properties, which can be much harder to predict. Notable examples include the Mott and other metal-to-insulator transitions [12], unconventional superconductivity [13–15], and quantum magnetism and spin liquid behavior [16,17]. For example, predicting the low-energy properties of strongly correlated systems has proven to be extremely challenging, even with the advent of state-of-the-art computational algorithms and the widespread availability of high-performance computing. A case in point is the single-band Hubbard model, whose doping-temperature phase diagram is rich with many competing and intertwined orders [15,18] and may or may not contain a superconducting ground state [19–21].

In the Landau paradigm, a phase transition is characterized by an order parameter—a measurable quantity encoding some macroscopic property of the system that undergoes a discontinuous change at a critical point (e.g., a critical temperature, critical pressure, etc.). The change in the order parameter is tied to an associated symmetry breaking, where the order parameter is zero in the high-symmetry disordered phase and nonzero in the low-symmetry ordered phase. Identifying when a phase transition occurs thus requires knowledge of an appropriate order parameter or the corresponding symmetry.

Physical intuition or experimental input can provide insight towards the identification of the correct order parameter and its relevant symmetry. However, there are well-known examples of order parameters that are nonlocal or exist in a more abstract space. Notable examples include the Haldane transition in spin-1 antiferromagnetic chains [22,23], the breaking of gauge symmetry across the superconducting transition [24], or the emergence of topological order in the quantum Hall states [25]. In cases like these, there is no general method for identifying order parameters and their associated symmetries. There are also materials where cross-over behavior is observed that may or may not be associated with a true phase transition. Perhaps the most famous example of this is the pseudogap "phase" of the high-T_c cuprates [26]. Thus, it would be very advantageous to have a general method of identifying an order parameter, detecting sudden changes in its value, and determining the corresponding broken symmetry across the transition. Our goal here is to introduce such a method.

In recent years, techniques from machine learning—in particular, deep neural networks (DNNs) [27]—have been used to successfully identify phase transitions in both classical and quantum many-body lattice systems in a purely data-driven manner [28–35]. DNNs are complex parametric models consisting of an alternating composition of linear and nonlinear transformations; such models now constitute the state of the art for a variety of problems in domains such as computer vision and natural language understanding [36–38]. To date, most applications using DNNs to detect phase transitions have focused on Monte Carlo (MC) simulations of lattice models, which is natural given that large volumes of training and validation data can be easily generated [35].

More recently, methods from unsupervised learning have been applied to the problem of identifying phase transitions [39–43]. Unsupervised learning is the paradigm used to find structure in unlabeled data, such as its intrinsic dimensionality. Perhaps the best-known methods for dimensionality reduction are principal components analysis (PCA) and the autoencoder, where the latter is a DNN with an encoder-decoder architecture that may be thought of as a nonlinear generalization of PCA [44,45].

The main contribution of this paper is a DNN method for identifying phase transitions, which we call the group-equivariant autoencoder (GE autoencoder). In contrast to previous methods cited above, the GE autoencoder is specifically designed to identify which symmetries of a given system are broken at each point in a region of its phase diagram; the identification of the corresponding phase transition is thus a corollary. In this way, the GE autoencoder not only locates phase transitions but also gives insight into its mechanism via the associated spontaneous symmetry breaking (SSB). Our method assumes only that (1) we have knowledge of the symmetry group G of the system Hamiltonian and that (2) we have selected a latent dimensionality for the GE autoencoder; the key steps of the GE-autoencoder method are then the following:

(1) Use group theory to deduce the subgroup G_{NB} of "never-broken symmetries," i.e., the symmetries in G that remain intact in all phases of the system.

(2) Constrain the GE autoencoder such that it learns a G_{NB}-invariant order parameter.

(3) Train the GE autoencoder using "symmetry regularization" such that it learns a G-equivariant order parameter.

During training, the GE autoencoder learns the representation of G by which the order parameter transforms, and from this we can extract information about the associated SSB.

The advantage of the GE autoencoder over previous symmetry-agnostic DNN methods is threefold. First, the GE autoencoder exploits knowledge about the symmetries of the system that would otherwise be wasted. The point of using ML for identifying phase transitions is that it does not require us to have knowledge of the relevant order parameter, but this does not mean we should forget the knowledge we may have, such as symmetries of the high-energy microscopic Hamiltonian. Second, due to the never-broken symmetries constraining the GE autoencoder as well as training with symmetry regularization, we expect the GE autoencoder to locate phase transitions with greater accuracy, efficiency, and robustness than symmetry-agnostic methods. Third and finally, as already mentioned, the GE autoencoder not only identifies phase transitions but also provides information about the associated SSB, thereby elucidating its mechanism. Having access to details on broken symmetries provides information on how to couple to the order parameter via a conjugate field, a requirement for probing associated phase transitions in the laboratory.

Since we are introducing the GE autoencoder, we focus on the details of the methodology in this paper and test it as a proof of principle on the 2D ferromagnetic and antiferromagnetic Ising models. Moreover, we discuss numerous implementation details throughout the paper that were essential for obtaining conclusive results. The paper is organized as follows: In Sec. II we review as background the mathematical notion of SSB, the Ising model, and autoencoders. In Sec. III we describe the GE-autoencoder method in detail, focusing on the case of a 1D order observable for clarity; we also describe the experimental setup, including calculations of the DNN initialization and learning rate settings required for fair experimental comparisons. In Sec. IV we present our experimental results; we find that the GE autoencoder accurately identifies which symmetries are broken at each temperature and estimates the critical temperature with greater accuracy, time efficiency, and robustness than a baseline autoencoder. We concurrently give additional details of the data analysis, including a method for extracting stable critical temperature estimates from statistics of trained autoencoder models. In Sec. V we extend the GE autoencoder to support arbitrary finite symmetry groups and vector-valued order observables, paving the way for future applications. Finally, in Sec. VI we conclude the paper with a discussion of its key findings, implications, and directions for future work [46].

II. BACKGROUND

A. Spontaneous symmetry breaking

In this section we review the concept of spontaneous symmetry breaking (SSB) from a mathematical perspective, which will help us formulate the method. This discussion is based on the one given in Ref. [47].

Consider a classical many-body system on a lattice whose size is parameterized by L (e.g., for a hypercubic lattice, L is the size of one dimension). Let $\mathbb{X}$ denote the space of all lattice configurations, and suppose the system Hamiltonian is invariant under the action of a group G on $\mathbb{X}$. An *equilibrium state* is then a distribution of lattice configurations that maximizes the entropy subject to a fixed expected internal energy. In other words, the equilibrium state solves a constrained convex optimization problem. For finite systems, the equilibrium state is uniquely the well-known Boltzmann distribution over lattice configurations. In the thermodynamic limit ($L \to \infty$), on the other hand, uniqueness is no longer necessary, and in general we have a polyhedral solution set $\mathcal{S}$ of equilibrium states. An abrupt change in the structure (e.g., dimensionality) of this set $\mathcal{S}$ with respect to temperature or any other tuning parameter in the Hamiltonian is called a *phase transition*. Here we restrict ourselves to disorder-order phase transitions, where the equilibrium state changes from being unique (disordered) to not unique (ordered).

When viewed in this framework, every symmetry (element) in G sends an equilibrium state to an equilibrium state. Thus, in the disordered phase, the unique equilibrium state is itself G-invariant. However, in the ordered phase, the equilibrium states are no longer necessarily G-invariant, as they may permute under the action of an element of G. This phenomenon is often called *spontaneous symmetry breaking* (SSB).

The abstract polyhedral set $\mathcal{S}$ of equilibrium states is made concrete by way of an *order parameter*—a linear embedding of $\mathcal{S}$ into a Euclidean space of dimension $d = \dim(\mathcal{S})$, with the centroid of the polyhedron mapped to the origin. By the Riesz-Markov-Kakutani Representation Theorem, there exists an *order observable* $\mathcal{O} : \mathbb{X} \mapsto \mathbb{R}^d$ such that the order parameter sends each equilibrium state ν to the expectation

$$\langle \mathcal{O} \rangle_\nu = \int_{\mathbb{X}} \mathcal{O}(x)\, d\nu(x).$$

Without loss of generality, $\mathcal{O}$ can be chosen such that (1) $\langle \mathcal{O} \rangle_\nu = 0$ in the disordered phase and (2) it is G-equivariant; by G-equivariant, we mean $\mathcal{O}(gx) = \psi_g \mathcal{O}(x)$ for all $g \in G, x \in \mathbb{X}$, where ψ is a nontrivial real-orthogonal representation of G. Every symmetry $g \in G$ for which $\psi_g = 1$ is then said to be *never broken* and is otherwise *broken* in the ordered phase.

Markov chain Monte Carlo (MCMC) simulations of lattice systems are designed to converge to the average equilibrium state $\overline{\nu}$ (the centroid of the set $\mathcal{S}$), which is G-invariant and satisfies $\langle \mathcal{O} \rangle_{\overline{\nu}} = 0$ in both the disordered and ordered phases. Hence, the order parameter as defined above cannot be used to distinguish the two phases. It is possible, however, to define a general expression in terms of the observable $\mathcal{O}$ that can. For simplicity, for most of this work we focus our exposition on scalar order parameters ($d = 1$); for the case of higher-dimensional order parameters ($d \geqslant 2$), see Sec. V. In the scalar case, the expected absolute value $\langle |\mathcal{O}| \rangle_{\overline{\nu}}$ is sufficient to distinguish the two phases, taking a value of zero in the disordered phase (in the thermodynamic limit) and a nonzero value in the ordered phase. Although a misnomer, we will refer to $\langle |\mathcal{O}| \rangle_{\overline{\nu}}$ throughout this paper as the order parameter and will drop the subscript $\overline{\nu}$. In the limited case of $d = 1$, the representation ψ takes values in $\{-1, 1\}$ with $\psi_g = -1$ for at least one $g \in G$ since the representation must be nontrivial.

B. The Ising model

One of the simplest and best-studied lattice systems is the classical Ising model in two dimensions; it is both rich enough to exhibit a second-order phase transition while also admitting an exact solution [48]. We consider the Ising model on a square $L \times L$ lattice with L even and periodic boundary conditions. A lattice configuration is obtained by assigning to each lattice site a classical spin $x_\mathbf{i} = \pm 1$, where $\mathbf{i} = (i_x, i_y)$ are the spatial indices of the site. The space of all lattice configurations is $\mathbb{X} = \{-1, 1\}^{L \times L}$. The Ising Hamiltonian is

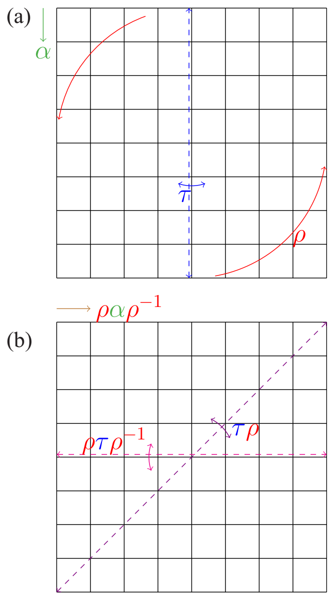

FIG. 1. (a) Generators of the spatial Ising symmetries acting on a square lattice and (b) some example compound spatial symmetries that can be formed from the generators.

$$\mathcal{H}(\mathbf{x}) = -J \sum_{\langle \mathbf{i}, \mathbf{j} \rangle} x_\mathbf{i} x_\mathbf{j}, \tag{1}$$

where $\mathbf{x}$ is an $L \times L$ matrix with entries $x_\mathbf{i} \in \{-1, 1\}$, J is the coupling constant, and the sum is taken over all pairs of neighboring lattice sites. We set $J = \pm 1$, where $J = 1$ (resp. $J = -1$) corresponds to a ferromagnetic (resp. antiferromagnetic) magnetic interaction.

Equation (1) has both spatial (translations, reflections, and orthogonal rotations) and spin-flip ($\mathbf{x} \to -\mathbf{x}$) internal symmetries. The Ising symmetry group G admits a presentation with independent generators α, ρ, τ, and σ (see Fig. 1). Picturing the Ising lattice as a matrix, α can be interpreted as a downward (cyclic) translation; ρ is a 90° counterclockwise rotation about the origin; τ is a reflection about the vertical line of symmetry; and σ is the spin-flip internal symmetry (not shown in Fig. 1). Every symmetry operation of the Ising model can be expressed in terms of the four generators of G. For example, a rightward translation can be expressed as $\rho\alpha\rho^{-1}$, while a reflection about the diagonal can be expressed as $\tau\rho$, as shown in Fig. 1. Algebraically, this presentation of G is defined by the following relations:

$$\begin{aligned}
\alpha^L &= \rho^4 = \tau^2 = \sigma^2 = 1,\\
\rho\tau &= \tau\rho^3,\\
\alpha\rho^2 &= \rho^2\alpha^{-1},\\
\alpha\tau &= \tau\alpha,\\
g\sigma &= \sigma g \,\forall g \in G.
\end{aligned}$$

In both the ferromagnetic and antiferromagnetic cases, the 2D Ising model exhibits a second-order phase transition at the

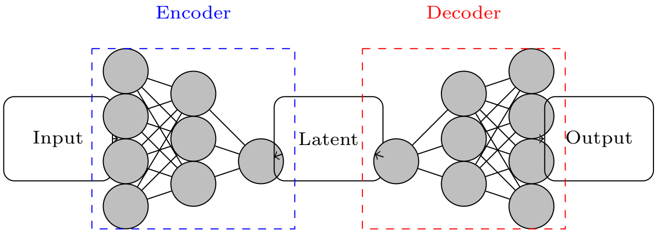

FIG. 2. Schematic illustration of the autoencoder architecture.

critical temperature $T_c/J = \frac{\sqrt{2}}{\log(1+\sqrt{2})} = 2.269\ldots$ [48]. The phase transition is associated with the spontaneous breaking of the $\mathbf{x} \mapsto -\mathbf{x}$ internal symmetry. The associated order parameter is the expected absolute value $\langle |M| \rangle$ (resp. $\langle |M_{\text{stag}}| \rangle$) of the magnetization M (resp. staggered magnetization M_{stag}) in the ferromagnetic (resp. antiferromagnetic) case, where

$$M(\mathbf{x}) = \frac{1}{L^2} \sum_{\mathbf{i}} x_{\mathbf{i}} \quad \text{and} \tag{2}$$

$$M_{\text{stag}}(\mathbf{x}) = \frac{1}{L^2} \left(\sum_{i_x+i_y \text{ even}} x_{\mathbf{i}} - \sum_{i_x+i_y \text{ odd}} x_{\mathbf{i}} \right). \tag{3}$$

Note that both M and M_{stag} are equivariant functions with respect to the spatial and internal Ising symmetries.

C. Autoencoders

The core of our method is the autoencoder, a DNN architecture used for various unsupervised learning tasks [44,45], which we use for dimensionality reduction or "compression." Given a data set $\{\mathbf{x}_n \in \mathbb{R}^m\}_{n=1}^N$, it is a common assumption in the traditional domains of computer vision and natural language understanding that the data points lie on a low-dimensional manifold embedded in $\mathbb{R}^m$. The autoencoder is a means to discovering this intrinsic manifold structure. An autoencoder consists of a pair of DNNs—an encoder $\mathcal{O} : \mathbb{R}^m \mapsto \mathbb{R}^d$ (which will ultimately represent an observable in our application) and a decoder $\mathcal{D} : \mathbb{R}^d \mapsto \mathbb{R}^m$, where $d < m$ is the assumed dimensionality of the intrinsic data manifold (Fig. 2). The encoder thus maps its input to a low-dimensional "latent" or "compressed" representation in terms of intrinsic coordinates on the manifold, and the decoder attempts to reconstruct the original input given the latent representation by learning the embedding of the manifold into $\mathbb{R}^m$. The autoencoder is trained by minimizing the reconstruction loss

$$\mathcal{L}(\mathcal{O}, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} L_{\text{metric}}(\mathcal{D}[\mathcal{O}(\mathbf{x}_n)], \mathbf{x}_n),$$

where $\mathcal{L}(\mathcal{O}, \mathcal{D})$ means that $\mathcal{L}$ is a function of the network parameters of $\mathcal{O}$ and $\mathcal{D}$, and L_{metric} is some metric (such as mean square error or binary cross-entropy) that measures the difference between the reconstructed and original inputs. Once trained, the encoder can be used to obtain low-dimensional "summaries" of the data. Below we describe how this aspect can be used for identifying phase transitions.

III. METHODS

A. Detecting phase transitions with autoencoders

In our autoencoder, the encoder is exactly the sought-after order observable $\mathcal{O} : \mathbb{X} \mapsto \mathbb{R}^d$ as introduced in Sec. II A, except that we do not require it to be G-equivariant at present. The decoder $\mathcal{D} : \mathbb{R}^d \times \mathbb{R} \mapsto \mathbb{X}$ then represents the conditional Boltzmann distribution of lattice configurations given a value of the observable $\mathcal{O}$ and a temperature T. Note that in contrast to traditional autoencoders, this decoder accepts a second argument—the temperature—as a direct input; we do this because we know the Boltzmann distribution of lattice configurations depends on temperature. This explicit temperature dependence is the first unique element of our method, as previous works on autoencoders for identifying phase transitions assumed a temperature-independent architecture. Once the autoencoder is trained, we may interpret any abrupt change in the distribution of the learned observable $\mathcal{O}$ with respect to temperature as indicative of a phase transition.

Even in the absence of any knowledge about the symmetries of the Hamiltonian, previous works found that autoencoders could identify phase transitions with some accuracy [42]. However, the reason for the autoencoder's efficacy remains unclear at this time. Rather than relying only on the empirical success of autoencoders, we motivate their use with the following intuition: Using general information theory, we can show that training an autoencoder is equivalent to maximizing the entropy of the observable $\mathcal{O}$ learned by the encoder, where we regard $\mathcal{O}$ as a function of the random lattice configuration across different temperatures. Thus, training the autoencoder moves the distribution of $\mathcal{O}$ closer to a uniform distribution with as large of a support as possible, and this in turn means that $\mathcal{O}$ learns to aggregate low-probability states together. This property seems to mimic the type of coarse-graining performed in Landau theory, where the competition between high-probability states and aggregations of low-probability states drives a phase transition.

For the example application of the Ising model, we define the encoder and decoder to have shallow neural network architectures, each with one hidden layer of nonlinear activation units:

$$\mathcal{O}(\mathbf{x}) = \mathbf{c} + \sum_{k=1}^{h} \mathbf{a}_k \phi(\langle \mathbf{w}_k, \mathbf{x} \rangle_F + b_k), \tag{4}$$

$$\mathcal{D}(\mathbf{z}, T) = \tanh\left[\mathbf{c}' + \sum_{k=1}^{h'} \mathbf{a}'_k \phi(\langle \mathbf{w}'_k, \mathbf{z} \rangle_F + b'_k + b'' T) \right], \tag{5}$$

where $b_k \in \mathbb{R}$, $\mathbf{a}_k, \mathbf{c} \in \mathbb{R}^d$, $\mathbf{w}_k \in \mathbb{R}^{L\times L}$; $b'_k, b'' \in \mathbb{R}$, $\mathbf{a}'_k, \mathbf{c}' \in \mathbb{R}^{L\times L}$, $\mathbf{w}'_k \in \mathbb{R}^d$; h and h' are the number of hidden neurons in the encoder and decoder, respectively; $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product (Hadamard product of matrices followed by a sum over all entries); tanh() is applied elementwise; and $\phi : \mathbb{R} \mapsto \mathbb{R}$ is the elementwise leaky rectified linear unit (ReLU) activation function defined as

$$\phi(y) = \begin{cases} 0.01y, & \text{if } y < 0 \\ y, & \text{otherwise.} \end{cases} \tag{6}$$

The tanh function is used in the decoder to guarantee each output component lies in the interval $(-1, 1)$. We set $h = 4$ and

$h' = 64$. We also set $d = 1$, as stated in Sec. II A. Since the magnetization and staggered magnetization are linear functions, a linear encoder and linear decoder would have been sufficient. However, for the purpose of demonstrating the efficacy of our method, we assume no knowledge of the system except a data set of MC-sampled lattice configurations over a range of temperatures and the group G of Ising model symmetries. We therefore consider an architecture deliberately more complex than a linear autoencoder, and one that would be a reasonable initial choice given no additional information about the system.

Now, given a data set $\{(\mathbf{x}_n, T_n) \in \mathbb{X} \times [0, \infty)\}_{n=1}^{N}$ of lattice configurations $\mathbf{x}_n$ at temperatures T_n, we train the autoencoder by minimizing the loss

$$\mathcal{L}(\mathcal{O}, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} L_{\mathrm{BCE}}(\mathcal{D}(\mathcal{O}(\mathbf{x}_n), T_n), \mathbf{x}_n), \tag{7}$$

where $L_{\mathrm{BCE}} : (-1, 1)^{L\times L} \times \{-1, 1\}^{L\times L} \mapsto (0, \infty)$ is the binary cross-entropy loss function defined as

$$L_{\mathrm{BCE}}(\hat{\mathbf{x}}, \mathbf{x}) = -\sum_{\mathbf{i}} \left[\left(\frac{1+x_{\mathbf{i}}}{2}\right) \log\left(\frac{1+\hat{x}_{\mathbf{i}}}{2}\right) + \left(\frac{1-x_{\mathbf{i}}}{2}\right) \log\left(\frac{1-\hat{x}_{\mathbf{i}}}{2}\right)\right], \tag{8}$$

where $\hat{\mathbf{x}}$ is the output of the autoencoder.

B. The group-equivariant autoencoder

We now extend the baseline autoencoder introduced in Sec. III A to a *group-equivariant autoencoder* (GE autoencoder) by incorporating our prior knowledge about the symmetries of the system's Hamiltonian into the network architecture. Once trained, we will then be able to interpret the GE autoencoder to infer which symmetries are spontaneously broken at any temperature.

1. The subgroup of never-broken symmetries

The first step is to see if we can identify a subgroup of "never-broken symmetries"—symmetries that do not spontaneously break at any temperature. Identifying these will reduce the number of symmetries that we will ultimately have to check. Our approach for this step is entirely group-theoretic.

Recall from Sec. II A that a symmetry $g \in G$ will remain unbroken as a function of model parameters and temperature if $\psi_g = 1$. In the absence of any knowledge about the true representation ψ associated with the order parameter, we can deduce a subgroup of never-broken symmetries by finding all symmetries $g \in G$ such that $\psi_g = 1$ for all representations $\psi : G \mapsto \{-1, 1\}$. We establish such a subgroup for the Ising symmetry group in Proposition 1 (see Appendix A 1); we denote the subgroup as $\mathrm{SCB}(L) = \langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle$ and refer to it as the *special checkerboard group*, as it represents the group of all proper (i.e., no reflections) symmetries of an $L \times L$ checkerboard that map black (resp. white) squares onto black (resp. white) squares.[1] Thus, all even-parity translational symmetries and the 180° rotational symmetry are never spontaneously broken in the Ising model.

Having established a subgroup of never-broken symmetries, it can be shown (see Appendix A 1) that the only symmetries we have to check for SSB are $\alpha^{m_1}\rho^{m_2}\tau^{m_3}\sigma^{m_4}$ for $m_i \in \{0, 1\}$. This represents a reduction from $16L^2$ to 16 symmetries to check, so that the complexity of detecting SSB is now independent of lattice size, an important advance of the proposed method.

2. Incorporating symmetries into the encoder

The next step of our method is to incorporate the deduced subgroup of never-broken symmetries into our autoencoder. Recall from Sec. II A that the observable $\mathcal{O} : \mathbb{X} \mapsto \mathbb{R}^d$, which is modeled by the encoder network of the autoencoder, must be G-equivariant; we start by first constraining the parameters of the encoder $\mathcal{O}$ [Eq. (4)] such that it is invariant to the subgroup of never-broken symmetries $\mathrm{SCB}(L)$. However, it turns out that there are many inequivalent ways to do this, and it is unclear which set of constraints is optimal. A complete classification of all ways this information can be incorporated, as well as the development of a metric by which to determine which way is best, is beyond the scope of this paper and is left for future work. Here we enforce invariance in a simple way and find that it yields good results. The general idea is illustrated in the top panel of Fig. 3.

We start with Eq. (4) for the observable encoder $\mathcal{O} : \mathbb{X} \mapsto \mathbb{R}$. We constrain the elements $w_{k,\mathbf{i}}$ of each matrix $\mathbf{w}_k$ to be

$$w_{k,\mathbf{i}} = \frac{2}{L^2} \begin{cases} u_k, & \text{if } i_x + i_y \text{ is even} \\ v_k, & \text{otherwise.} \end{cases} \tag{9}$$

Each $\mathbf{w}_k$ is thus constrained to have a "checkerboard" pattern and is invariant under the action of $\mathrm{SCB}(L)$. Since it can be shown that G acts orthogonally on all of $\mathbb{R}^{L\times L}$, then the invariance of $\mathcal{O}$ under $\mathrm{SCB}(L)$ immediately follows.

The constraints imposed on the $\mathbf{w}_k$ allow for a significant simplification of the expression for the encoder; this is illustrated in Fig. 3 (bottom). For an $L \times L$ lattice configuration $\mathbf{x}$, define

$$\check{x}_{\mathrm{B}} = \frac{2}{L^2} \sum_{i_x+i_y \text{ even}} x_{\mathbf{i}},$$

$$\check{x}_{\mathrm{W}} = \frac{2}{L^2} \sum_{i_x+i_y \text{ odd}} x_{\mathbf{i}}.$$

We refer to $(\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})$ as the "checkerboard average" of the lattice configuration $\mathbf{x}$, i.e., the average value over all "black squares" and the average value over all the "white squares." Then we have

$$\langle \mathbf{w}_k, \mathbf{x}\rangle_F = u_k \check{x}_{\mathrm{B}} + v_k \check{x}_{\mathrm{W}},$$

which allows for a more efficient implementation of the encoder. We define the reduced encoder $\check{\mathcal{O}} : [-1, 1]^2 \mapsto \mathbb{R}$

[1] The set of black squares (resp. white squares) is also referred to as sublattice A (resp. sublattice B) in the literature.

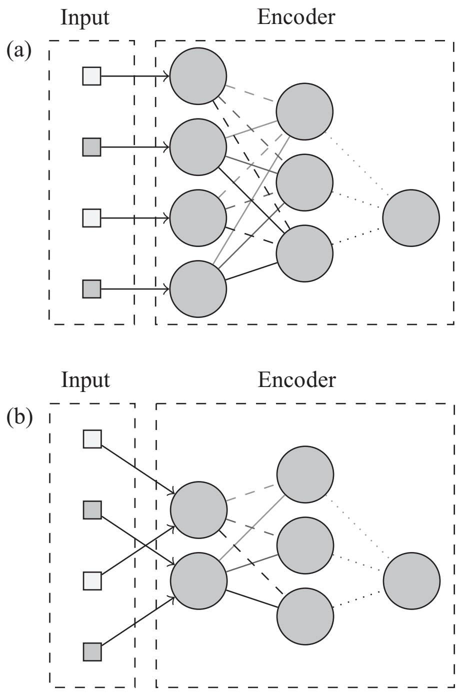

FIG. 3. (a) Schematic illustration of an encoder constrained to be invariant to a group of symmetries (even-unit cyclic translations in the illustration). Network edges of same line style (solid vs dashed vs dotted) and shade are constrained to have equal weight. (b) Reduced representation of the same encoder. Rather than replicating weights in the first layer, lattice sites that feed into the same input neuron are first averaged.

by

$$\check{\mathcal{O}}((\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})) = c + \sum_{k=1}^{h} a_k \phi(u_k \check{x}_{\mathrm{B}} + v_k \check{x}_{\mathrm{W}} + b_k). \tag{10}$$

This result allows us to evaluate the encoder in Eq. (4) in two separate steps: First, we compute the checkerboard average $(\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})$ of the input lattice configuration $\mathbf{x}$. This task is a one-time calculation and can be done across the entire available data set of lattice configurations as a preprocessing step. Second, we evaluate the reduced encoder [Eq. (10)] on these checkerboard averages. This task is now independent of the lattice size L. Note that the checkerboard average of $\mathbf{x}$ is manifestly invariant under the action of $\mathrm{SCB}(L)$, and thus so is the reduced encoder $\check{\mathcal{O}}$. Moreover, the form of the reduced encoder [Eq. (10)] places additional constraints on the spatial symmetries and implies $\psi_\alpha = \psi_\rho = \psi_\tau$ (see Proposition 2 in Appendix A 2 for details). The upshot is that we now need only estimate ψ_σ and one of ψ_α, ψ_ρ, and ψ_τ from the data; we choose ψ_τ without loss of generality. If $\psi_\tau = 1$, then all spatial symmetries in $\langle \alpha, \rho, \tau \rangle$ never break spontaneously. Estimating ψ_σ and ψ_τ is discussed in Sec. III B 4.

3. *Incorporating symmetries into the decoder*

As with the encoder, we now incorporate the subgroup of never-broken symmetries into the decoder of our autoencoder. The starting point is Proposition 3 (see Appendix A 3), which states that under suitable conditions, if an unsupervised model is fit to a data set containing symmetries, then the fit model will be invariant to those symmetries at least when restricted to the data set. Proposition 3 provides a strong motivation to assume that our autoencoder $f = \mathcal{D} \circ \mathcal{O}$ is G-invariant; for $g \in G$,

$$g\mathcal{D}(\mathcal{O}(\mathbf{x}), T) = \mathcal{D}(\mathcal{O}(g\mathbf{x}), T),$$
$$g\mathcal{D}(\mathcal{O}(\mathbf{x}), T) = \mathcal{D}(\psi_g \mathcal{O}(\mathbf{x}), T).$$

We demand this hold for all $\mathcal{O}(\mathbf{x})$ and make the stronger assumption

$$g\mathcal{D}(z, T) = \mathcal{D}(\psi_g z, T), \forall g \in G, z \in \mathbb{R}.$$

We have already deduced that $\psi_g = 1$ for all $g \in \mathrm{SCB}(L)$, giving us the constraint

$$g\mathcal{D} = \mathcal{D}, \forall g \in \mathrm{SCB}(L).$$

This constraint necessitates the output of the decoder to have a checkerboard pattern as in Eq. (9), and thus it suffices to have the decoder return only two values—one representing the value on the black squares and the other for the white squares. Therefore, we define the reduced decoder $\check{\mathcal{D}} : \mathbb{R} \times (0, \infty) \mapsto (-1, 1)^2$ as

$$\check{\mathcal{D}}(z, T) = \tanh\left[\mathbf{c}' + \sum_{k=1}^{h'} \mathbf{a}'_k \phi(w'_k z + b'_k + b'' T)\right], \tag{11}$$

where $\mathbf{a}'_k, \mathbf{c}' \in \mathbb{R}^2$ and $w'_k, b'_k, b'' \in \mathbb{R}$; h' is the number of hidden neurons (we set $h' = 64$); and the functions tanh and ϕ are applied elementwise. We interpret the output as the checkerboard average of the output of the unreduced decoder $\mathcal{D}$. Note that like the reduced encoder, the reduced decoder is now independent of the lattice size L. The reduced encoder [Eq. (10)] and reduced decoder [Eq. (11)] together comprise a reduced autoencoder $\check{\mathcal{D}} \circ \check{\mathcal{O}}$, which can now be trained directly on the preprocessed and reduced data set of checkerboard-averaged lattice configurations.

4. *Symmetry regularization*

The final step is to ensure that the (reduced) encoder not only is invariant to the subgroup $\mathrm{SCB}(L)$ of never-broken symmetries but also is in fact G-equivariant; we want $\check{\mathcal{O}}(g\check{x}) = \psi_g \check{\mathcal{O}}(\check{x})$ for all $g \in \{\alpha, \rho, \tau, \sigma\}$. By Proposition 2, it is sufficient to consider only τ and σ, and since $\psi_g = \pm 1$, then we want $\check{\mathcal{O}}(g\check{x}) = \pm \check{\mathcal{O}}(x)$ for all $g \in \{\tau, \sigma\}$. Ideally, we would implement this as a hard constraint to which the optimization of the reduced autoencoder loss function would be subject; however, because the representations of G are discrete objects, the feasible region defined corresponding to this hard constraint is likely disconnected, with inequivalent irreducible representations ψ corresponding to distinct connected components. To circumvent this issue, we relax the hard equivariant constraint to a soft constraint, implemented as regularization terms included in the reduced autoencoder loss function:

$$\begin{aligned}\check{\mathcal{L}}(\check{\mathcal{O}}, \check{\mathcal{D}}) = {} & \frac{1}{N}\sum_{n=1}^{N} L_{\mathrm{BCE}}(\check{\mathcal{D}}(\check{\mathcal{O}}(\check{x}_n), T_n), \check{x}_n) \\ & + \lambda \sum_{g\in\{\tau,\sigma\}} \left(1 - \frac{\|\check{\mathcal{O}} \circ g\|}{\|\check{\mathcal{O}}\|}\right)^2 \\ & + \lambda \sum_{g\in\{\tau,\sigma\}} [1 - L_{\cos}(\check{\mathcal{O}}, \check{\mathcal{O}} \circ g)^2] \\ & + \lambda \min_{g\in\{\tau,\sigma\}} [1 + L_{\cos}(\check{\mathcal{O}}, \check{\mathcal{O}} \circ g)], \end{aligned} \tag{12}$$

where $\lambda \geqslant 0$ is a regularization coefficient and $L_{\cos}$ is the cosine similarity between observables defined as

$$L_{\cos}(\mathcal{O}_1, \mathcal{O}_2) = \frac{\langle \mathcal{O}_1, \mathcal{O}_2 \rangle}{\|\mathcal{O}_1\| \|\mathcal{O}_2\|},$$

where

$$\langle \mathcal{O}_1, \mathcal{O}_2 \rangle = \frac{1}{N}\sum_{n=1}^{N} \mathcal{O}_1(\mathbf{x}_n)^\top \mathcal{O}_2(\mathbf{x}_n).$$

The first regularization term [i.e., the second term in Eq. (12)] enforces the soft constraint $\|\check{\mathcal{O}} \circ g\| \approx \|\check{\mathcal{O}}\|$. The second regularization term drives the cosine similarity to one of its extreme values ± 1. Together, these two terms encode the constraint $\check{\mathcal{O}} \circ g \approx \pm 1 \check{\mathcal{O}}$ as desired. To explain the final regularization term, recall that we require $\psi_g = -1$ for some $g \in G$ to avoid a trivial representation. The last term drives at least one of ψ_τ and ψ_σ to -1 to satisfy this requirement. With this regularization approach, inequivalent irreducible representations ψ now correspond to distinct "valleys" in the loss landscape, with the regularization coefficient λ determining the "steepness" of these valleys.

Once trained, we estimate ψ_g for $g \in \{\tau, \sigma\}$ with the final cosine similarity:

$$\psi_g \approx L_{\cos}(\check{\mathcal{O}}, \check{\mathcal{O}} \circ g). \tag{13}$$

C. Experimental setup

1. Data sets

We generate data sets of lattice configurations by MC-sampling the 2D ferromagnetic and antiferromagnetic Ising models. We impose periodic boundary conditions on an $L \times L$ lattice and consider $L = 16$, 32, 64, and 128. For each lattice size, we consider 100 temperatures with 25 values in [1.04, 2] in increments of 0.04, 50 values in [2.01, 2.5] in increments of 0.01, and 25 values in [2.54, 3.5] in increments of 0.04. This distribution of temperatures is evenly distributed about the theoretical critical temperature $T_c = \frac{2}{\log(1+\sqrt{2})} = 2.269\ldots$ and denser near T_c. Although not uniform, the temperature samples are constant across all order observable models and thus do not effect the model comparison; we sample more temperatures near T_c only to ensure we achieve results sufficiently stable to draw meaningful conclusions. For each lattice size and temperature, we run the Wolff algorithm first for 10 000 iterations to allow for thermal equilibration and then for an additional 50 000 iterations during which we record the lattice configuration every 10 iterations. We thus obtain 5000 samples for each lattice size and temperature and for each of the ferromagnetic and antiferromagnetic cases (although in practice we only use 4096 samples). We also preprocess copies of these data sets by checkerboard averaging the lattices, which will be used to train the GE autoencoder. We evenly split each set of 4096 lattice configurations into a training-validation set and a test set, and we further partition the 2048 training-validation samples into eight "folds" each of size 256 that will be used to measure sampling variance.

2. Order observables

Given only the MCMC data sets and the Ising symmetry group G, our objective is to detect when a phase transition occurs by (1) identifying the associated spontaneous symmetry breaking and (2) estimating the temperature where it occurs (i.e., the critical temperature). Importantly, we assume no prior knowledge about the Ising model beyond the given data sets and the symmetry group.

We test three "order observables" from which we hope to derive order parameters:

(1) Magnetization [Eqs. (2) and (3)]: (In the antiferromagnetic case, "magnetization" will be understood to mean the staggered magnetization.) This observable is the standard order parameter used for the Ising model. Here we use it to provide a ground-truth estimate for the critical temperature in comparison to the exact value obtained from the Onsager solution.

(2) Baseline autoencoder [Eqs. (4)–(7)]: This autoencoder does not exploit the symmetry group G and is used as a machine learning baseline. We will refer to its encoder and decoder as baseline encoder and baseline decoder, respectively. Once trained, we interpret the output of the encoder as an order observable. Note that G-equivariance is not guaranteed *a priori*.

(3) GE autoencoder [Eqs. (10)–(12)]: This autoencoder takes advantage of the symmetry group G, and we thus expect it to be more accurate and more efficient than the baseline autoencoder. We will refer to its encoder and decoder as GE encoder and GE decoder, respectively. Once trained, we interpret the output of the encoder as a G-equivariant order observable. We also interpret its representation of G to identify which symmetries spontaneously break. Finally, as the GE autoencoder acts on checkerboard-averaged lattice configurations, the same network architecture can be applied to different sizes of lattices. Therefore, we will also consider the case of a "multiscale GE autoencoder," which is trained simultaneously on all four lattice sizes in our data set while using only one-quarter of the MCMC data for each lattice size.

We evaluate magnetization and the trained baseline encoder and GE encoder observables on all lattice configurations in our data sets to obtain measurement distributions and subsequently order parameters. Further details on using these order parameters to estimate the critical temperature are given in Sec. IV.

3. Training details

Independent of the lattice size L and training-validation fold j, we train and validate the baseline and GE autoencoders on a data set of $100N$ lattice configurations, where 50% of the data set is randomly selected for training and the remaining 50% is used for validation. This data set consists of the last N MC-sampled lattice configurations (out of the total 256 configurations in the jth fold) at each of the 100 temperatures considered. We test various values of N (ranging from 8 to 256 in powers of 2) to measure the data efficiency of the GE autoencoder vs baseline autoencoder, as well as the dependence of estimated T_c on the training-validation sample size. We train all autoencoders using the Adam optimizer with learning rate of 0.001 (or equivalent; see below) and minibatch size N for 64 epochs; this choice guarantees 50 iterations in each epoch and ensures that the scale of the noise generated from stochastic gradient descent is the same for all experiments [49]. We test three different seeds to randomly initialize the autoencoder network parameters. Together with eight training data folds, we thus have 24 trials of each autoencoder experiment.

While training the GE autoencoder, we include the symmetry regularization terms in the loss function [Eq. (12)] only for the second half of training epochs. Intuitively, we start with no (or weak) regularization so the randomly initialized GE autoencoder is free to escape the nearest valley in the loss landscape that could possibly correspond to an incorrect group representation ψ (i.e., the breaking of incorrect symmetries). Assuming there is enough information in the training data to guide the GE autoencoder into a neighborhood of the "correct" valley, we strengthen regularization after some time to drive the GE autoencoder to the minimum of the valley—thereby refining the learned representation ψ to a bona fide orthogonal group. We note that switching on regularization halfway through training is, in a sense, the simplest schedule from weak to strong regularization, and our choice of regularization schedule is further validated in the same way as all other optimizer hyperparameter settings, namely, we observed lower validation loss compared to constant or linearly increasing regularization strength.

Nontrivial parameter initialization and learning rate settings were needed to obtain reasonable comparisons (see Appendix B for details). We initialize the GE autoencoder as usual and set the learning rate to 0.001 based on validation learning curves. However, to ensure a fair comparison between the baseline and GE autoencoders, and to avoid artifactual and noisy results due to hand-tuned hyperparameter settings, we initialize the baseline autoencoder such that it is functionally equivalent to the initial GE autoencoder; i.e., the baseline autoencoder satisfies the same symmetry constraints as the GE autoencoder at initialization time. We then set a separate learning rate for each layer of the baseline autoencoder such that it would remain equivalent to the GE autoencoder throughout training if we maintained the symmetry constraints on the baseline autoencoder. This requires setting smaller learning rates for larger layers to prevent large sums of parameter updates flowing through the network. As a result of these settings, the baseline autoencoder and GE autoencoder are identical in terms of their initial values and their learning dynamics and differ only in the symmetry constraints and symmetry regularization imposed on the GE autoencoder.

IV. RESULTS

A. Identifying phase transitions

Figure 4 is representative of the observed distributions of the magnetization, baseline-encoder output, and GE-encoder output over all lattice configurations in our data set at each temperature. For brevity, we present the distributions only for the largest lattice size $L = 128$ and the largest number of training-validation samples per temperature $N = 256$. In contrast to magnetization and the GE encoder, the distribution of the baseline encoder is not symmetric about zero in the antiferromagnetic case. This asymmetry is a consequence of a redundancy in the autoencoder network: The (baseline) encoder may be freely transformed by any invertible affine function since the first layer of the decoder can always undo it. The center of the baseline-encoder distribution is therefore arbitrary. Although previous works [42] have reported approximately symmetric encoder distributions for the Ising model, our results show that this is not guaranteed unless some form of explicit symmetry regularization is used, as in the GE encoder. Similarly, the scale of the encoder is arbitrary as well (even in the GE encoder), although the scale is not relevant for SSB. Nevertheless, all distributions exhibit an abrupt qualitative change near the theoretically known critical temperature $T_c = \frac{2}{\log(1+\sqrt{2})} = 2.269\ldots$, and hence all three observables are able to identify the phase transition in the Ising model to some degree.

We derive an order parameter from each of the three observables by calculating the mean absolute value under each distribution at each temperature (Fig. 5). Note that while this procedure is justified for magnetization and the GE encoder as these are G-equivariant observables, it is not justified *a priori* for the asymmetric baseline encoder observable in the antiferromagnetic case. Nevertheless, we do it anyways to provide a baseline case where symmetries were not taken into consideration. Due to symmetry constraints and regularization, the GE encoder learns a smoother order parameter that is almost identical to the magnetization order parameter up to a scale factor (this is made quantitative in Appendix C). We also compare each order parameter to Onsager's exact solution for spontaneous magnetization in the thermodynamic limit [48]

$$M_{\mathrm{ONS}}(T) = \begin{cases} \left[1 - \sinh^{-4}\left(\frac{2}{T}\right)\right]^{\frac{1}{8}}, & \text{if } T < T_c \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

which is plotted as the dashed blue line in Fig. 5. The smooth order parameter of the GE encoder provides a better approximation to Onsager's solution compared to the baseline encoder (Fig. 5). Moreover, the GE encoder in the thermodynamic limit converges to Onsager's solution with less error than the baseline encoder (see Appendix C for details). As an immediate consequence, the GE encoder is able to identify the phase transition as being second-order, which previous works [42] could not do.

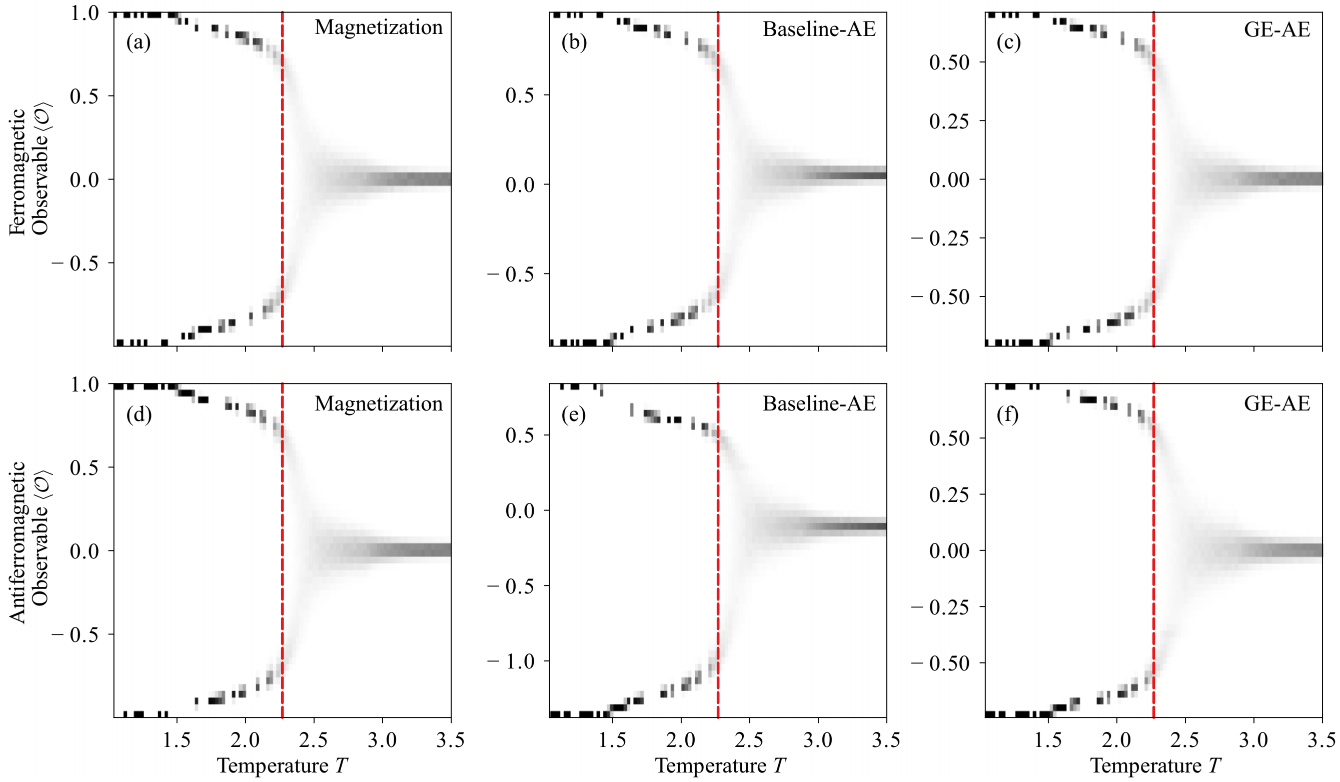

FIG. 4. Distributions of the order observable values at each temperature T on a lattice of size $L = 128$ for one example training data fold and initialization seed and using $N = 256$ training-validation samples per temperature. The top row shows results for the ferromagnetic model with the observable derived from (a) the magnetization, (b) the baseline AE, and (c) the GE AE. The bottom row (d)–(f) shows corresponding results obtained for the antiferromagnetic model. The observable scales (y axes) of the baseline encoder and GE encoder are arbitrary, as any nonzero rescaling of the encoder can be compensated by the inverse scaling in the decoder; thus, only the relative shapes of these distributions are meaningful. All three order observables suggest a phase transition in both the ferromagnetic and antiferromagnetic cases near the theoretical critical temperature (dashed red).

B. Identifying spontaneously broken symmetries

Can we identify at each temperature which symmetries of the system have spontaneously broken? We have seen that the GE-encoder order parameter becomes nonzero below some critical temperature (Fig. 5) and thus breaks the $\mathbb{Z}_2 = \psi_G$ symmetry. It follows that every Ising symmetry $g \in G$ such that $\psi_g = -1$ breaks below this critical temperature, while $\psi_g = 1$ implies that g remains unbroken. Using Eq. (13) we estimate ψ_g for each generator of G (Table I). In the ferromagnetic case, we find that $\psi_g \approx -1$ only for $g = \sigma$, and hence only the internal spin-flip symmetry breaks. In the antiferromagnetic case, we obtain $\psi_g \approx -1$ for every generator $g \in \{\alpha, \rho, \tau, \sigma\}$. In other words, the internal spin-flip symmetry as well as every spatial symmetry not in the special checkerboard subgroup SCB(L) breaks. Our results are in agreement with the known SSB in the Ising model across the magnetic transition, and thus we conclude that our GE-autoencoder method can correctly and accurately detect SSB.

For contrast, we also measure the degree to which the baseline encoder is equivariant; we again use Eq. (13) but replace the GE encoder $\check{\mathcal{O}}$ with the baseline encoder $\mathcal{O}$. As with the GE encoder, we average estimates over all 24 trials (eight training data folds and three initialization seeds) as well as over all lattice sizes L and training-validation sample sizes N. In the ferromagnetic case, we find that $\psi_g \approx 0.9998(2)$ for spatial symmetry generators $g \in \{\alpha, \rho, \tau\}$ and $\psi_\sigma \approx -0.92(8)$ for the internal spin-flip symmetry generator σ. In the antiferromagnetic case, we obtain $\psi_g \approx -0.87(5)$ for both spatial and internal symmetry generators. The baseline encoder is thus approximately equivariant and transforms by approximately the correct group representation. However, the GE autoencoder learns the representation ψ with significantly greater accuracy and sometimes with orders of

TABLE I. Estimated latent representation ψ_g of the spatial symmetry generators ($g = \alpha, \rho, \tau$) and the internal symmetry generator ($g = \sigma$). Estimates were averaged over all 24 trials (eight training data folds and three initialization seeds) as well as over all lattice sizes L and training-validation sample sizes N as they showed little variation; the reported uncertainties are standard deviations. In the antiferromagnetic case, odd spatial symmetries and the internal symmetry spontaneously break at some temperature; in the ferromagnetic case, only the internal symmetry spontaneously breaks.

	Spatial	Internal
Ferromagnetic	0.99996(27)	−1.00001(29)
Antiferromagnetic	−0.99982(42)	−0.99993(40)

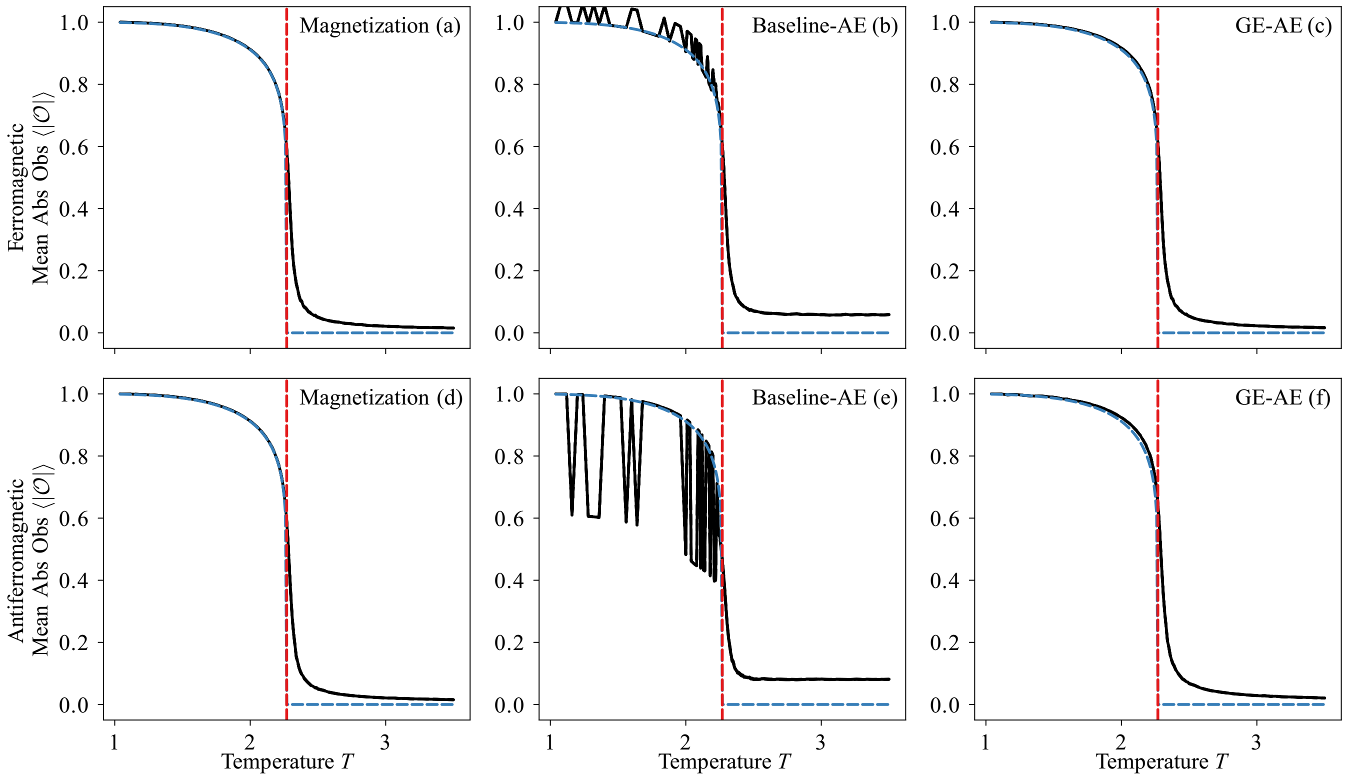

FIG. 5. Order parameters (mean absolute value of the distributions in Fig. 4) vs temperature T on a lattice of size $L = 128$ for one example training data fold and initialization seed and using $N = 256$ training-validation samples per temperature. The top row shows results for the ferromagnetic model with the observable derived from (a) the magnetization, (b) the baseline AE, and (c) the GE AE. The bottom row (d)–(f) shows corresponding results obtained for the antiferromagnetic model. The curves have been normalized to the same scale as Onsager's solution for comparison. The standard deviation curves (dashed) were obtained using the jackknife resampling method (these may be difficult to see as the standard deviations are small). All three order parameters suggest a phase transition in both the ferromagnetic and antiferromagnetic cases near the theoretical critical temperature (dashed red). However, the order parameters derived from magnetization and the GE encoder are smoother and give better approximations to Onsager's solution (dashed blue).

magnitude more robustness than the baseline, particularly in the antiferromagnetic case.

C. Estimating the critical temperature

We now turn to our second question: Is an SSB-based approach to identifying phase transitions from data more accurate than a purely data-driven approach? In particular, does the GE encoder give a more accurate estimate of T_c than the baseline encoder?

We begin by estimating T_c independently for each lattice size based on the fourth Binder cumulant [50]:

$$U_4 = 1 - \frac{\langle \mathcal{O}^4 \rangle}{3\langle \mathcal{O}^2 \rangle^2},$$

where $\mathcal{O}$ is the order observable. We obtain Binder cumulant vs temperature curves for each of the three order observables (Fig. 6). (Once again, for brevity, we present the curves only for the largest lattice size $L = 128$ and number of training-validation samples per temperature $N = 256$.) We emphasize that we are simulating a scenario in which we have access to only a data set of lattice configurations and the symmetries of the system and are not aware that the system is in fact the Ising model. Our choice to look at the Binder cumulant should therefore be interpreted only as a "guess," and its only justification is the *a posteriori* observation that the Binder cumulant curves all display an abrupt change near the theoretical critical temperature.

We perform least-squares regression to fit a step function to each jackknife-sample Binder cumulant vs temperature data set. We then interpret the location of the jump discontinuity of the step function as a jackknife-sample estimate of T_c. However, if our data set includes temperatures $T_1, T_2, \dots, T_{100}$ and if we find the jump discontinuity to lie in the open interval $T_i, T_{i+1})$ for some i, then moving the jump discontinuity to any other temperature in (T_i, T_{i+1}) would result in a fit that is just as good as the original step function. This approach therefore allows us only to obtain interval estimates of the critical temperature. To obtain point estimates, we set up and solve a convex optimization problem in which we seek to minimize the jackknife standard deviation in the estimate subject to the constraints defined by the jackknife-sample interval estimates (see Appendix D for details). In this way, for each lattice size L and number of training-validation samples per temperature

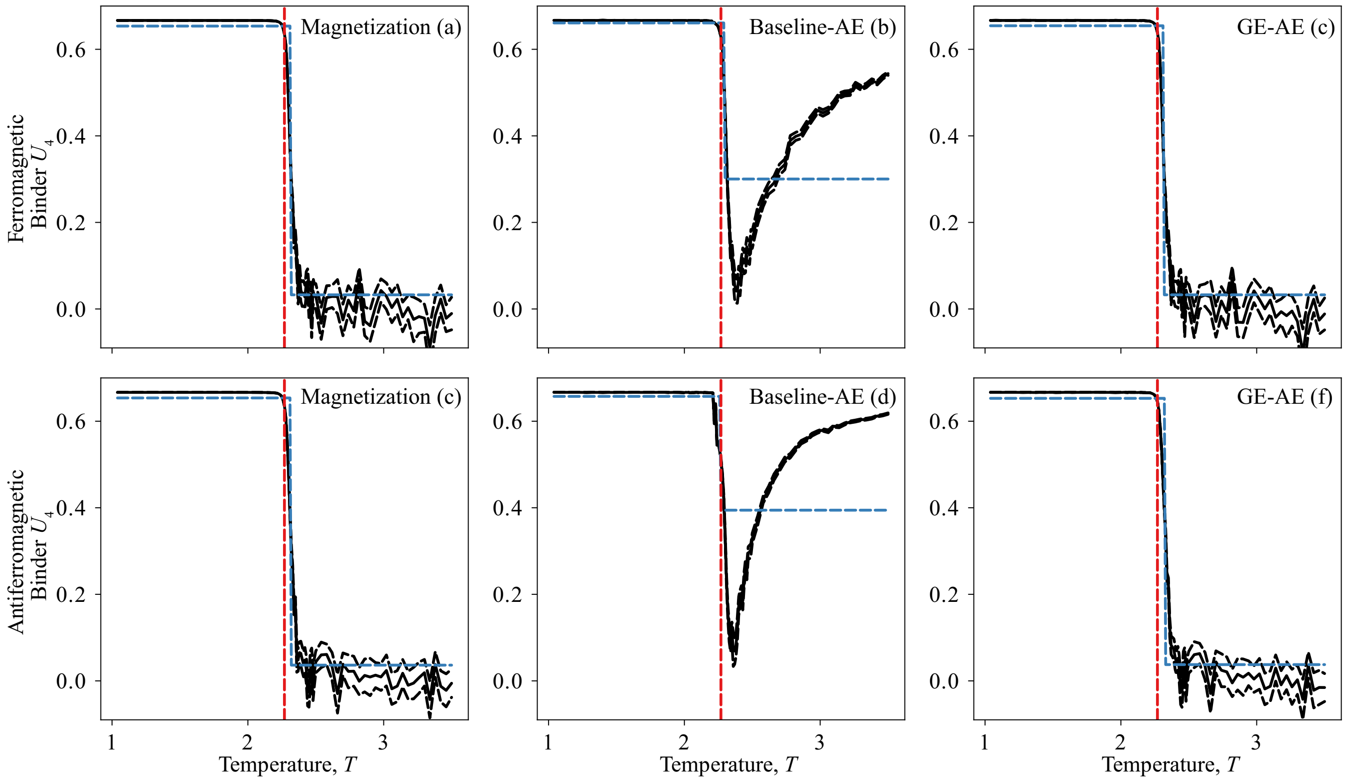

FIG. 6. Fourth-order Binder cumulants U_4 of the order observables at each temperature T on a lattice of size $L = 128$ for one example training data fold and initialization seed and using $N = 256$ training-validation samples per temperature. The top row shows results for the ferromagnetic model with the data derived from (a) the magnetization, (b) the baseline AE, and (c) the GE AE. The bottom row (d)–(f) shows corresponding results obtained for the antiferromagnetic model. The standard deviation curves (dashed) were obtained using the jackknife resampling method (these may be difficult to see since the standard deviations are small). In all cases, the Binder cumulant drops abruptly near the theoretical critical temperature (dashed red). Step functions (blue) are fit to the Binder cumulants using least squares, and the locations of their jump discontinuities are taken as estimates of the critical temperature.

N, we obtain a critical temperature estimate as the jackknife[2] mean averaged over all 24 trials (eight training data folds and three initialization seeds), along with a standard deviation (Figs. 7 and 8).

In both the ferromagnetic and antiferromagnetic cases, the baseline encoder consistently achieves lower error in its critical temperature estimates than do the GE encoders (Figs. 7 and 8). Moreover, in contrast to the GE encoders, the baseline encoder makes better use of more training data, as its error decreases with increasing training data set size. The GE encoders, on the other hand, achieve errors closer to that of magnetization, and their estimates are also more stable in terms of lower standard deviations. Multiscale training (i.e., all four lattice sizes in the training data set) results in an additional reduction in error.

We speculate that the baseline encoder achieves the lowest error in T_c estimation because it is a more flexible network in comparison to the GE encoder. As such, it may be able to express certain nonlinearities that the GE encoder cannot. From the proximity of the GE encoder to magnetization, we infer that its four hidden neurons have aligned such that the GE encoder is approximately a linear function of its input. In contrast, if the baseline encoder has learned a nonlinearity such that it squashes (resp. inflates) the value assigned to lattice configurations with low (resp. high) absolute magnetization, then the mean absolute value of the baseline encoder vs temperature curve will be more “bowed” compared to Onsager’s solution. Such bowing is indeed what we see (Fig. 5). This observation is consistent with previous works reporting

[2] We remark on an important detail in the jackknife calculation: It is common to reduce the bias in the jackknife estimate by combining the jackknife mean with the estimate obtained without resampling. The argument for this, however, relies on a Taylor expansion of the underlying estimator [51], and it turns out that our critical temperature estimator is not everywhere-differentiable; under a small perturbation of the Binder cumulant estimates, our critical temperature estimate either remains constant or changes abruptly if the jackknife-sample interval estimates change. Indeed, in our original T_c estimates, we found that the bias estimate was either zero or so extreme that it often pushed the critical temperature estimate outside the range of temperatures included in our data set. In contrast, when we did not reduce the bias, we obtained more stable results. Overall, since the bias estimate is known to scale as $\frac{1}{n}$ while the jackknife standard deviation scales as $\frac{1}{\sqrt{n}}$—so that the bias is typically much smaller than the standard deviation for a sufficiently large sample size n and can often be ignored [51]—we were confident that our extreme bias estimates were spurious. Therefore, we have not adjusted for them in the results presented (Figs. 7 and 8).

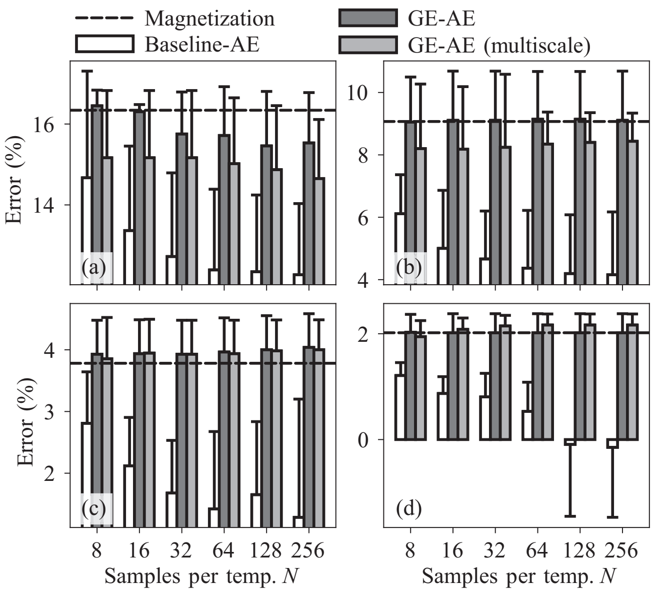

FIG. 7. Ferromagnetic critical temperature estimates (expressed as percent errors relative to the exact theoretical critical temperature) for lattice sizes (a) $L = 16$, (b) 32, (c) 64, and (d) 128. The error bars represent standard deviations that combine the standard deviation of means across 24 trials (eight training data folds and three initialization seeds) and the jackknife standard deviations for each fold and seed, with most of the variance coming from the former. The standard deviation associated with magnetization is 0. The baseline encoder consistently achieves the lowest error and improves with more training data. The GE encoders, on the other hand, are more similar to magnetization, with multiscale training reducing error slightly, and are also more stable in terms of their standard deviations.

that deeper and more flexible autoencoders incorrectly classify the Ising phase transition as first-order [42], and this could also explain why the baseline encoder estimates the critical temperature with greater accuracy, even though it does worse when extrapolated to the thermodynamic limit (see Sec. IV D).

D. Extrapolating the critical temperature estimates

Here we perform finite-size scaling analysis on the critical temperature estimates presented in the last section in order to obtain estimates at infinite lattice size, i.e., in the thermodynamic limit. For each training data fold and initialization seed, order observable, training-validation sample size, and each of the ferromagnetic and antiferromagnetic cases, we perform least-squares linear regression on the critical temperature estimates against inverse lattice size (r^2 value ≈1 for all fits). We plot an example of these fits for $N = 256$ training-validation samples per temperature in Fig. 9. The y intercepts of the linear fits are then taken to be the critical temperature estimates at $L^{-1} = 0$, i.e., the thermodynamic limit. We find that while the baseline encoder achieves the lowest error in its estimation of the critical temperature for individual finite lattice sizes (Figs. 7 and 8), the GE encoders are significantly more accurate once their estimates are extrapolated to infinite lattice size (Fig. 10).

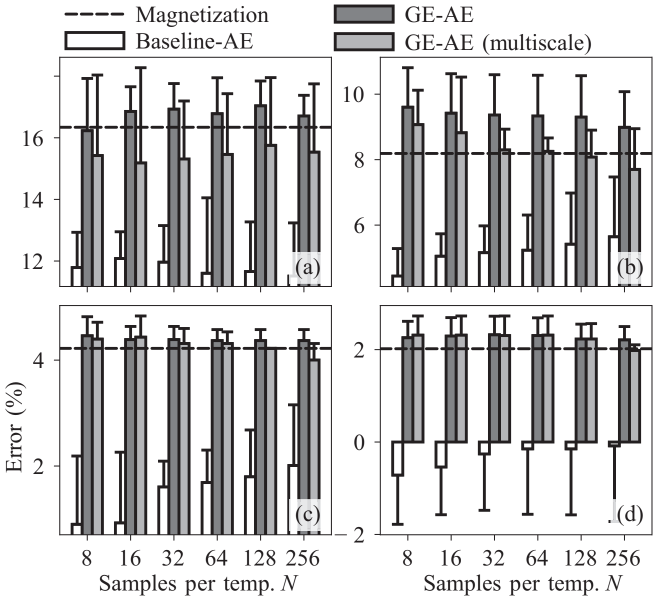

FIG. 8. Antiferromagnetic critical temperature estimates (expressed as percent errors relative to the exact theoretical critical temperature) for lattice sizes (a) $L = 16$, (b) 32, (c) 64, and (d) 128. The error bars represent standard deviations that combine the standard deviation of means across 24 trials (eight training data folds and three initialization seeds) and the jackknife standard deviations for each fold and seed, with most of the variance coming from the former. The standard deviation associated to magnetization is 0. The baseline encoder consistently achieves the lowest error and improves with more training data. The GE encoders, on the other hand, are more similar to magnetization, with multiscale training reducing error slightly, and are also more stable in terms of their standard deviations.

E. Measuring the time efficiency

We compare the time efficiencies of the GE-autoencoder and baseline-autoencoder methods. Since we were able to exploit never-broken symmetries to reduce the network size of the GE autoencoder, we expect it to be significantly more efficient. Figure 11 reports the total computation times for the GE-autoencoder and baseline-autoencoder methods for each lattice size. These values include the time to generate all the data (i.e., run the MC simulation), all preprocessing time such as checkerboard averaging the lattice configurations, and the time needed to train and validate the autoencoder and to evaluate the trained encoder on the entire data set of lattice configurations. Importantly, the training-validation-evaluation time is a *sum* over all 24 trials (eight training data folds and three initialization seeds) to reflect the computation needed to obtain error bars on the critical temperature estimates. Whether we are looking at the ferromagnetic or antiferromagnetic case has no impact on execution time, as the autoencoder architectures and learning hyperparameter settings are identical in both cases. Moreover, as a consequence of allowing the minibatch size to proportionally vary with the training-validation sample size N, we found that the execution time depended very little on N. Thus, we report each execution time as an average over all sample sizes N and over the ferromagnetic vs antiferromagnetic cases. We find that the GE-autoencoder method is significantly faster

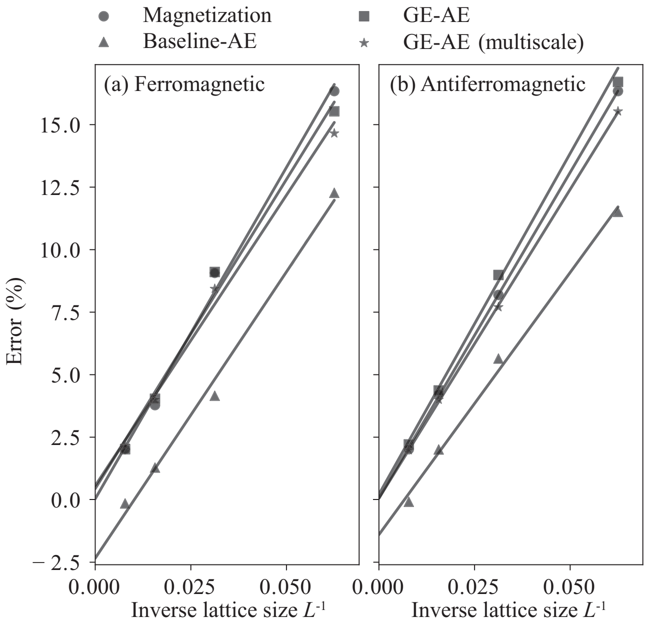

FIG. 9. Linear dependence of critical temperature estimates (expressed as percent errors relative to the exact theoretical critical temperature) on inverse lattice size using $N = 256$ training-validation samples per temperature and averaged over 24 trials (eight training data folds and three initialization seeds). Results for the ferromagnetic model are shown in (a), and results for the antiferromagnetic model are shown in (b). Although the baseline encoder achieves lower error for each lattice size individually, the linear extrapolation of its estimates to infinite lattice size (y intercept) is a worse estimate compared to the GE encoders.

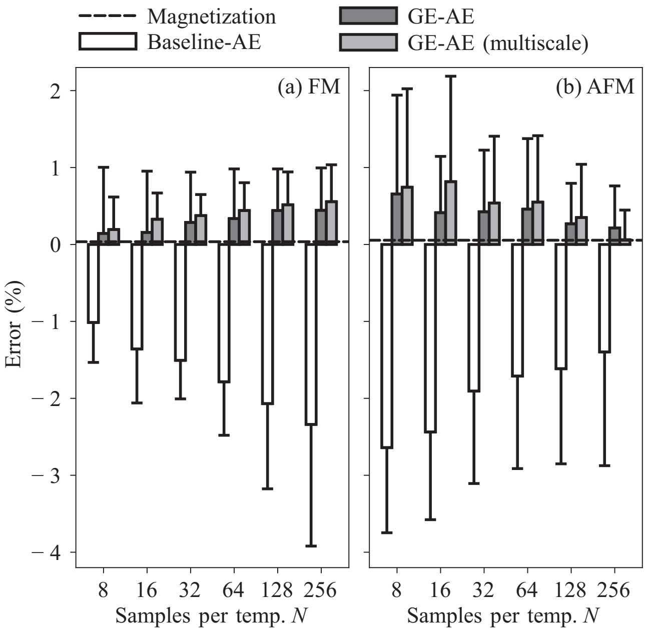

FIG. 10. Critical temperature estimates (expressed as percent errors relative to the exact theoretical critical temperature) extrapolated to infinite lattice size. Results for the ferromagnetic (FM) model are shown in panel (a), and results for the antiferromagnetic (AFM) model are shown in panel (b). The error bars represent standard deviations that combine the standard deviation of means across 24 trials (eight training data folds and three initialization seeds) and the jackknife standard deviations for each fold and seed, with most of the variance coming from the former. The standard deviation associated to magnetization is 0. Although the baseline encoder achieves lower error on finite lattices (Figs. 7 and 8), the GE encoder estimates extrapolate to have lower error in the thermodynamic limit.

than the baseline autoencoder. Moreover, multiscale training gives an additional boost in efficiency in the computation time needed to extrapolate estimates to infinite lattice size, as the multiscale GE autoencoder needs to be trained only once across the four finite lattice sizes. We therefore conclude that our GE-autoencoder method is indeed more efficient than the baseline-autoencoder method.

F. Detecting an external magnetic field

Finally, we investigate if the baseline-encoder and GE-encoder order observables can be used to detect the presence of a weak external magnetic field h in the ferromagnetic Ising model by adding a term to Eq. (1):

$$\mathcal{H}(\mathbf{x}) \to \mathcal{H}(\mathbf{x}) - h\sum_{\mathbf{i}} x_{\mathbf{i}}. \qquad (15)$$

We assume the magnetic field is uniform and note that it breaks the internal symmetry of the Ising model. We consider two temperatures, 2.0 and 2.5, slightly below and above the critical temperature, and we consider three field strengths $0.001J$, $0.01J$, and $0.1J$ (where $J = 1$ is the coupling constant in the Ising Hamiltonian). For each case, as well as the case of no external field at all, we use the Wolff algorithm with a "ghost site" [52] to simulate the Ising model with lattice size $L = 128$ in an external magnetic field; we generate $N = 2000$ sample lattice configurations for each temperature and field strength. If $x_1^0, \ldots, x_N^0$ (resp. $x_1, \ldots, x_N$) are the sample lattice configurations in the absence (resp. presence) of an external magnetic field, then for each encoder $\mathcal{O}$, we compute the statistic

$$D = \left|\frac{1}{N}\sum_{n=1}^{N}\mathcal{O}(x_n) - \frac{1}{N}\sum_{n=1}^{N}\mathcal{O}\left(x_n^0\right)\right|. \qquad (16)$$

Using the baseline encoder and GE encoder already fitted to data as described in previous sections, we obtain 24 measurements of D for each encoder (eight training data folds and three initialization seeds). We then define the "confidence score"

$$\xi = \frac{\langle D\rangle}{\sqrt{\langle D^2\rangle - \langle D\rangle^2}}. \qquad (17)$$

Intuitively, a value sufficiently far from zero indicates that the order parameter has shifted and hence there is an external symmetry-breaking field. We normalize by the standard deviation of D so that ξ is independent of the arbitrary scale learned by each encoder; it also boosts the score when the measurement of D is robust across the 24 trials. Figure 12 shows the ξ scores for the baseline encoder and GE encoder at each temperature and external field strength. In all cases, the GE encoder attains a significantly higher score than does the baseline encoder, meaning that it is more sensitive to and detects with greater confidence the presence of a weak

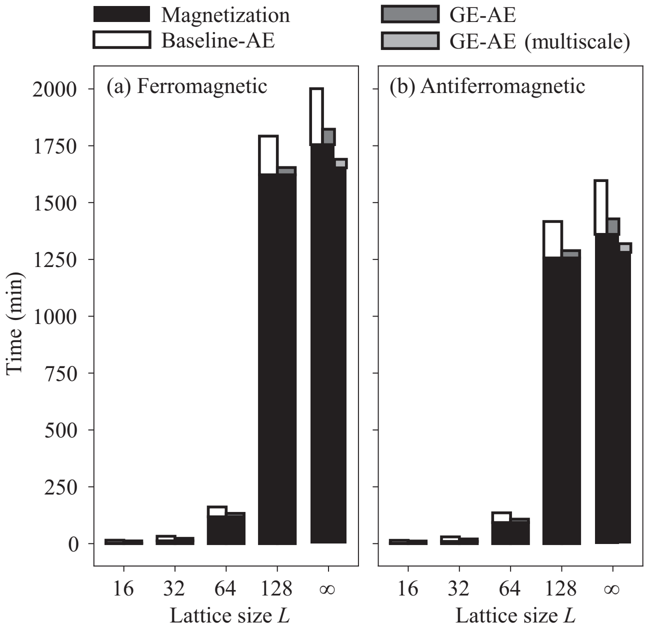

FIG. 11. Computation times to generate data (black) and train (not black) the autoencoders for each lattice size L. Results are shown for the (a) ferromagnetic and (b) antiferromagnetic cases. The time to measure magnetization is reported as equivalent to data generation time as these measurements were taken on the fly during the MC simulation. Each training time is a *sum* over all eight training folds and RNG seeds. The training times were then averaged over all training-validation sample sizes N and over both the ferromagnetic and antiferromagnetic cases, as these factors did not contribute to significant variation in execution time. The GE autoencoder is significantly more time-efficient (lower time) than the baseline autoencoder. The times reported for infinite lattice size are just sums of the times for the finite lattice sizes, although the multiscale GE autoencoder exhibits greater efficiency as it only requires a single network to be trained across all finite lattice sizes.

external magnetic field. At temperature 2.5, the baseline encoder does become more confident (increasing score) in its detection as the external field becomes stronger, which is to be expected; however, the GE encoder remains confident even in its detection of the weakest field.

V. ON VECTOR ORDER OBSERVABLES

A. A 2D order observable

We have thus far assumed that the order observable is scalar valued or equivalently that a sufficient choice for the latent dimension of the autoencoder networks is 1. In this section, we justify this assumption by instead assuming a 2D vector order observable; if the real 2D orthogonal representation $\psi : G \mapsto \mathrm{O}(2, \mathbb{R})$ by which the order observable transforms—as learned by the GE autoencoder—is equivalent to a 1D representation, then we may conclude that a 1D order observable is sufficient to describe the phase transition. The exposition in this section will also help to illustrate how our GE-autoencoder method can be extended to a somewhat more complex scenario.

We construct the 2D GE autoencoder in direct analogy to the 1D case. First, we calculate the subgroup of never-broken symmetries and find it to be $H = \langle \alpha^2, \beta^2 \rangle$, which is the subgroup generated by the even horizontal translations and even vertical translations (see Proposition 4 in Appendix E 1). This subgroup has $L^2/4$ symmetries for even lattice size L, and the corresponding quotient group G/H of possibly broken symmetries has $16L^2/(L^2/4) = 64$ elements. Compare this to the 1D case, in which there are L^2 never-broken symmetries and only 16 possibly broken symmetries.

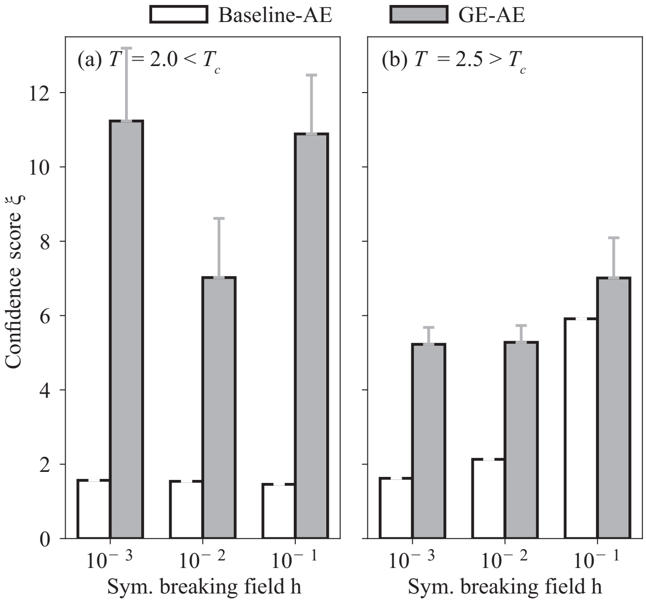

FIG. 12. "Confidence scores" ξ [see Eq. (17)] of the baseline encoder (white) and GE encoder (dark gray) at detecting the presence of an external magnetic field at temperatures (a) $T = 2 < T_c$ and (b) $T = 2.5 > T_c$. Results are shown for three different field strengths h. The error bars represent standard deviations estimated using 10 000 bootstrap draws from the 24 trials used to calculate ξ. The GE encoder consistently detects the external field with greater confidence than does the baseline encoder.

Next, we constrain the GE autoencoder so that its encoder is invariant to the subgroup H of never-broken symmetries. Instead of constraining each weight matrix $\mathbf{w}_k$ (associated to the kth hidden neuron) to have a checkerboard pattern as in the 1D case, we constrain it to be a tiling of a 2×2 submatrix; the resulting weight matrix is then invariant to the action of the subgroup H, and hence so is the encoder. Applying this constrained encoder to an input $L \times L$ lattice configuration $\mathbf{x}$ is then equivalent to the following procedure: First, we construct a reduced 4D representation $\check{x} = (\check{x}_1, \check{x}_2, \check{x}_3, \check{x}_4)$ of $\mathbf{x}$ as follows:

$$\check{x}_1 = \frac{4}{L^2} \sum_{i_x \mathrm{even}, i_y \mathrm{even}} x_{\mathbf{i}},$$

$$\check{x}_2 = \frac{4}{L^2} \sum_{i_x \mathrm{even}, i_y \mathrm{odd}} x_{\mathbf{i}},$$

$$\check{x}_3 = \frac{4}{L^2} \sum_{i_x \mathrm{odd}, i_y \mathrm{even}} x_{\mathbf{i}},$$

$$\check{x}_4 = \frac{4}{L^2} \sum_{i_x \mathrm{odd}, i_y \mathrm{odd}} x_{\mathbf{i}}.$$

TABLE II. Estimated 2D latent representation ψ_g by which the learned 2D order observable (i.e., GE encoder) transforms, for $g \in \{\rho, \tau, \sigma\}$. Estimates were averaged over all 24 trials (eight training data folds and three initialization seeds) as well as over all lattice sizes L and training-validation sample sizes N as they showed little variation; the reported uncertainties are standard deviations. In both the ferromagnetic and antiferromagnetic cases, the learned representations are equivalent to the 1D representations presented in Sec. IV B.

	Ferromagnetic	Antiferromagnetic
ψ_ρ	$\begin{bmatrix} 1.0000(2) & 0.000(0) \\ 0.0000(9) & 1.000(0) \end{bmatrix}$	$\begin{bmatrix} -0.96(5) & 0.000(0) \\ 0.00(0) & 0.994(9) \end{bmatrix}$
ψ_τ	$\begin{bmatrix} 1.0000(2) & 0.000(0) \\ 0.0000(8) & 1.000(0) \end{bmatrix}$	$\begin{bmatrix} -0.96(5) & 0.000(0) \\ 0.00(0) & 0.994(9) \end{bmatrix}$
ψ_σ	$\begin{bmatrix} -1.0000(8) & 0.000(0) \\ 0.0000(0) & 1.000(2) \end{bmatrix}$	$\begin{bmatrix} -0.96(5) & 0.000(0) \\ 0.00(0) & 0.994(9) \end{bmatrix}$

The representation $\check{x}$ is just the 2×2 block average of $\mathbf{x}$. Then, we feed $\check{x}$ into a reduced unconstrained encoder $\check{\mathcal{O}} : [-1, 1]^4 \mapsto \mathbb{R}^2$. Finally, we use the 4D output of a reduced decoder $\check{\mathcal{D}}$ to reconstruct an $L \times L$ lattice configuration by tiling a 2×2 block in the horizontal and vertical directions.

Let $\psi : G \mapsto \mathrm{O}(2, \mathbb{R})$ be the 2D latent representation of G by which the 2D order observable transforms. We already know $\psi_g = I \forall g \in H$, where I is the 2×2 identity matrix. Our particular choice of the GE-autoencoder architecture as described above places an additional constraint on ψ:

$$\psi_\alpha \check{\mathcal{O}} = \psi_{\rho^2\tau} \check{\mathcal{O}}$$

(see Proposition 5 in Appendix E 1). Thus, we need only learn three of the four generators of ψ while training the GE autoencoder: ψ_ρ, ψ_τ, and ψ_σ. This result is analogous to Proposition 2 for the 1D case.

We train the 2D GE autoencoder with the loss function [Eq. (20)] given below in Sec. V B; it is a generalization of the loss function [Eq. (12)] and holds for any order observable dimension d. We train the 2D GE autoencoder with the same data sets and optimizer settings as in the 1D case. After training, we obtain the learned latent representation ψ of G using the estimator [Eq. (19)] given below in Sec. V B. Observe that the GE autoencoder is invariant under the transformation $\check{\mathcal{O}}(\cdot) \to A\check{\mathcal{O}}(\cdot)$ and $\check{\mathcal{D}}(\cdot, \cdot) \to \check{\mathcal{D}}(A^{-1}\cdot, \cdot)$ for any invertible 2×2 matrix A. Thus, without loss of generality, we transform the learned representation ψ into the eigenbasis of ψ_σ, so that ψ_σ is diagonal with sorted diagonal elements. In both the ferromagnetic and antiferromagnetic cases, the elements ψ_ρ, ψ_τ, and ψ_σ—and hence the entire representation ψ—are approximately diagonal (Table II). More precisely, we see that the learned representation ψ approximately admits the decomposition $\psi = \psi_1 \oplus \psi_2$ where ψ_1 is the representation learned in the 1D case and ψ_2 is the trivial representation. Thus, the learned 2D representation is equivalent to the 1D representation in Sec. IV B, and hence a 1D order observable is sufficient.

TABLE III. Subgroup H_d of never-broken symmetries of the Ising symmetry group G assuming various dimensions d of the order observable. Here $\beta = \rho\alpha\rho^{-1}$ is the horizontal translation generator. The subgroups were computed using our GAP implementation of Eq. (18) for lattice sizes $L \in \{4, 8, 16\}$.

d	$\lvert H_d \rvert$	$\lvert G/H_d \rvert$	H_d
1	L^2	16	$\langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle$
2	$L^2/4$	64	$\langle \alpha^2, \beta^2 \rangle$
3	$L^2/4$	64	$\langle \alpha^2, \beta^2 \rangle$
4	1	$16L^2$	$\{1\}$

B. A general procedure

We end this section with some remarks on a general procedure for calculating the subgroup of never-broken symmetries and applying the GE-autoencoder method. Consider an arbitrary statistical-mechanical system in thermal equilibrium with finite symmetry group G, and suppose we assume an order observable of the system to have dimension d (we can regard d as a hyperparameter that we select as part of the GE-autoencoder architecture). Then by Proposition 6 (see Appendix E 2), the subgroup H_d of never-broken symmetries can be calculated as

$$H_d = \bigcap_{\chi \in \hat{G} | \deg_{\mathbb{R}}(\chi) \leqslant d} \ker(\chi), \tag{18}$$

where $\hat{G}$ is the set of all (complex-)irreducible characters of G; $\ker(\chi)$ is the kernel of the character χ defined as the preimage set $\chi^{-1}[\deg(\chi)]$ of the degree $\deg(\chi) = \chi(1)$; and we define $\deg_{\mathbb{R}}(\chi)$ to be the degree of the smallest real character built out of χ:

$$\deg_{\mathbb{R}}(\chi) = \deg(\chi) \begin{cases} 1, & \text{if } \mathrm{I}_{\mathrm{FS}}(\chi) = 1 \\ 2, & \text{otherwise,} \end{cases}$$

where $\mathrm{I}_{\mathrm{FS}}(\chi)$ is the Frobenius-Schur indicator of χ. We can therefore quickly calculate the subgroup of never-broken symmetries once we have the character table of G.

An immediate corollary of Eq. (18) (and Proposition 6) is that we have the normal series $H_1 \trianglerighteq H_2 \trianglerighteq \cdots \trianglerighteq \{1\}$; i.e., the subgroup of never-broken symmetries shrinks with increasing dimension of the order observable and eventually becomes trivial, so that beyond some finite value of d, the GE autoencoder is no longer able to exploit any never-broken symmetries and is as large as the baseline autoencoder.

We implemented Eq. (18) in GAP[3] and used our implementation to calculate the subgroup of never-broken symmetries for the Ising symmetry group G for various dimensions d of the order observable (Table III). We note that a limitation of this computational approach is that it requires a numerical value for the lattice size L, and the computation time increases with L. We ran our code for lattice sizes $L \in \{4, 8, 16\}$ and observed that the presentations of the returned subgroups in terms of generators and relations were independent of lattice

[3] GAP is a computer algebra system for computational discrete algebra with particular emphasis on computational group theory [53].

size, allowing us to conclude empirically that these presentations hold for arbitrarily large even lattice size L.

The subgroups of never-broken symmetries of the Ising symmetry group for $d \in \{1, 2\}$ agree with the theoretical calculations in Props. 1,5 (Table III). For $d = 3$, we get the same subgroup as with $d = 2$. For $d = 4$, the subgroup of never-broken symmetries becomes trivial, so that the GE autoencoder is just as large as the baseline autoencoder; this remains the case for all $d \geqslant 4$ since the subgroups form a descending series with increasing d.

Once we have determined the subgroup H_d of never-broken symmetries, we constrain the GE autoencoder so that the encoder is H_d-invariant and the decoder returns an H_d-invariant lattice configuration. If the encoder and decoder each has one hidden layer of neurons, then the starting point for determining the appropriate parameter constraints is a recent classification of all invariant shallow neural networks [54]. Finally, once the GE autoencoder architecture is selected, we train the network by minimizing a loss function with symmetry regularization that generalizes Eq. (12) for 1D order observables to order observables of any dimension d. The loss function is given below.

Suppose we have a data set of N lattice configurations. For every $g \in G$, let $Z_g \in \mathbb{R}^{N\times d}$ whose nth row is $\check{O}(g^{-1}\check{x}_n)$; i.e., Z_g is the matrix of outputs of the GE encoder evaluated on the data set, where all lattice configurations were first transformed by g^{-1} (we use g^{-1} instead of g because the GE-encoder outputs are stacked as row vectors in Z_g). For the identity element $g = 1$, let $Z = Z_1$ for further brevity. Let P be the orthogonal projection operators onto the null space of Z:

$$P = I - Z^+ Z,$$

where Z^+ is the Moore-Penrose pseudoinverse of Z and I the identity matrix. Finally, let A be a $d \times d$ learnable matrix parameter, and define the following estimator of ψ_g:

$$\hat{\psi}_g = Z^+ Z_g + PA. \tag{19}$$

Then the general loss function for a GE autoencoder with latent dimension d has the form

$$\check{\mathcal{L}}(\check{\mathcal{O}}, \check{\mathcal{D}}, A) = \frac{1}{N}\sum_{n=1}^{N} L_{\mathrm{BCE}}(\check{\mathcal{D}}(\check{\mathcal{O}}(\check{x}_n), T_n), \check{x}_n) + \lambda \sum_{g\in\Gamma}(R_g + S_g), \tag{20}$$

where $\Gamma \subset G$ is the minimal set of symmetry generators necessary to check (e.g., for Ising symmetries, $\Gamma = \{\tau, \sigma\}$ for $d = 1$ and $\Gamma = \{\rho\tau\sigma\}$ for $d = 2$) and

$$R_g = \|I - \hat{\psi}_g^\top \hat{\psi}_g\|_F^2, \tag{21}$$

$$S_g = \frac{\|Z_g - Z\hat{\psi}_g\|_F^2}{\|Z_g\|_F^2}, \tag{22}$$

and where $\|\cdot\|_F^2$ is the squared Frobenius matrix norm (sum of squared matrix elements). These regularization terms are obtained by relaxing the hard equivariance constraint $Z_g = Z\psi_g$ to the minimization of $\|Z_g - Z\psi_g\|_F^2$. Specifically, $\hat{\psi}_g$ is the minimizer (i.e., linear least-squares estimator); R_g ensures $\hat{\psi}_g$ is (approximately) an orthogonal representation; and S_g ensures $\|Z_g - Z\hat{\psi}_g\|_F^2$ is minimized. Note the denominator in Eq. (22) is included so the optimizer does not simply rescale the GE encoder.

The above loss function is an approximate generalization of the loss function for the case of a 1D order observable [Eq. (12)]. Specifically, if $d = 1$, then R_g and S_g reduce to expressions equivalent to the first and second regularization terms in Eq. (12), with S_g matching the second regularization term exactly (see Appendix E 3 for details). Equation (12) includes a third regularization term whose original purpose was to ensure a nontrivial representation ψ is learned; however, based on a small sample of numerical tests, we believe that this third regularization term is not strictly necessary, and we have thus omitted its generalization from Eq. (20).

VI. DISCUSSION

We introduced the group-equivariant autoencoder (GE autoencoder), a deep neural network (DNN) architecture that can be used to locate phase transitions by detecting the associated spontaneous symmetry breaking (SSB). We demonstrated its efficacy for the 2D classical ferromagnetic and antiferromagnetic Ising models, finding that the GE autoencoder (1) accurately determines which symmetries are broken at each temperature and (2) estimates the critical temperature with greater accuracy and time efficiency than an SSB-agnostic autoencoder.

We also found the GE autoencoder to be more robust than the baseline autoencoder in an interesting sense. Recall that we deliberately selected nonlinear architectures for the autoencoders, even though (staggered) magnetization—the "true" order observable—is a linear function of the lattice configuration. This models the likely scenario in real applications where the DNN being used is more expressive than the unknown order observable we are seeking. Ideally, if the DNN is robust, then it should be able to reduce its expressivity to fit the target order observable. As discussed in Sec. IV C, we suspect that the baseline autoencoder performs better than the GE autoencoder on finite lattices but worse in the thermodynamic limit because it learned an inappropriate nonlinear order observable; that the GE autoencoder could accurately learn magnetization without overfitting attests to its superior robustness. This robustness is also reflected in the GE autoencoder's greater sensitivity to the presence of an external symmetry-breaking magnetic field (Sec. IV F).

There are several implementation details in our method that are worth noting. First, in our proof-of-principle example, we found that the T_c estimation by fitting a step function to the fourth Binder cumulant vs temperature curve works well, and it still allows for finite-size scaling analysis (see Sec. IV D). In contrast, the intersection of second Binder cumulant curves across lattice sizes is not guaranteed for autoencoders, and the intersection points of fourth Binder cumulant curves across lattice sizes are not guaranteed to be unique. Second, proper initialization and learning rate settings (see Appendix B) were critical for a fair comparison of the baseline autoencoder and GE autoencoder. Our approach to determining these settings could be useful for deep learning experiments in general, where one DNN is a constrained copy of another. Finally, in contrast to ML methods for phase detection that are entirely data-driven, our method includes group-theoretic considerations that allow us to exploit some of the structure available in the problem, namely, symmetries; given the benefits we have identified here, we think this practice should always be followed whenever possible.

Our work has several physical implications as well. For example, recent progress has been made in transfer learning from small to large lattice sizes implementing ideas of block decimation from the renormalization group as a way to more efficiently extrapolate to the thermodynamic limit [33]. The GE autoencoder, on the other hand, is scale-independent and could therefore be well suited for this transfer-learning task. Indeed, we found that the multiscale GE autoencoder is more time-efficient and slightly more accurate than the single-scale GE autoencoder. In future work, we aim to improve the multiscale GE autoencoder until it saves us from having to simulate the largest lattice size (e.g., $L = 128$) entirely. For example, researchers have used different generative ML methods to learn the distribution of lattice configurations of a system, allowing them to simulate the system more efficiently than with traditional MC methods [55–60]. Combining these methods with ideas introduced in this paper could allow us to generate large sample lattices based only on a data set of small to moderate lattices, possibly offering a significant speedup over MC methods.

The superior accuracy of the GE autoencoder could also translate into greater robustness against the sign problem. There is empirical evidence suggesting that DNNs could overcome the sign problem in the single-band Hubbard model to some extent [29]. Subsequent work, however, showed that the sign problem returns at more extreme temperatures and doping [30], although this work also demonstrates that transfer-learning from sign-problem-free regions to sign-problem-prone regions in the phase diagram could work for small amounts of doping. If the GE autoencoder is indeed more robust than previous DNN methods, then perhaps it could allow us to access even more extreme temperatures and doping levels.

Finally, knowledge of the SSB associated with a phase transition has value beyond a means to phase detection alone. For example, identifying the relevant symmetries could help elucidate the mechanism driving subtle phase transitions and thus offer a means to control them in practice to realize real-world applications.

Last, we end with remarks on some potential improvements to our approach to be addressed in the future. First, we would like to formalize, further develop, and better understand the general procedure presented in Sec. V B. For example, the computational approach described in Sec. V A for finding the subgroup of never-broken symmetries of an arbitrary finite symmetry group requires that we specify a value for the system size; understanding how the never-broken symmetries depend on system size would thus require us to test various system sizes and then search for a pattern, and it is unclear how feasible this would be for general systems. On the other hand, if we consider only a class of systems for which the dependence of the symmetry group on the system size takes a particular form, then it may be feasible to derive a specialized algorithm for calculating the never-broken symmetries for arbitrary system sizes.

Second, we would like to consider example systems with more complicated symmetry groups. This could include those with continuous internal symmetries and gauge symmetries; however, a first step would be to consider finite non-Abelian internal symmetry groups that must act on vector-valued lattice configurations.

Third and finally, we would like to better understand how to choose the representation by which the never-broken symmetries are imposed on the GE autoencoder. Relating to this is our finding that the baseline autoencoder estimates the critical temperature with greater accuracy than the GE autoencoder for individual lattice sizes (i.e., without finite-size scaling analysis). This aspect was discussed in Sec. IV C. As speculated there, this observation could be due to the greater flexibility of the baseline autoencoder. However, we suspect that this greater flexibility is also why the baseline autoencoder exhibits a slightly sharper transition in its order parameter vs temperature curve, while the GE autoencoder learns an order parameter very close to that of magnetization. We thus suspect that the baseline autoencoder is more prone to incorrectly classifying the Ising phase transition as first-order instead of second-order (as was observed in Ref. [42]) when compared to the GE autoencoder. If greater flexibility is desired, we believe the better way is to use a more flexible GE autoencoder; this could become important for cases where the nature of the phase transition (i.e. continuous or weakly first-order) is under dispute. The network complexity required for the GE autoencoder to match the critical-temperature-estimation performance of the baseline autoencoder is likely related to our choice of how we impose the never-broken symmetries. We intend to investigate this relationship in future work.

Much work remains to be done to establish the GE autoencoder as a mature and trusted machine learning model; however, the proof-of-principle application presented here provides optimism for its utility in addressing open problems related to the search for spontaneous symmetry breaking in the pseudogap phase of the high-temperature cuprates or at topological phase transitions.

ACKNOWLEDGMENTS

Work by D.A. and J.O. was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Award No. DE-SC0018175. Work by S.J. and A.D. was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award No. DE-SC0022311.

APPENDIX A: PROPOSITIONS

1. Subgroup of never-broken Ising symmetries

Recall from Sec. II A that a symmetry $g \in G$ is never broken if $\psi_g = 1$ [i.e., if g is in the kernel of ψ, denoted $\ker(\psi)$]. In the absence of any knowledge about the true representation ψ associated with the order parameter, we can deduce a subgroup of never-broken symmetries by finding all symmetries $g \in G$ such that $\psi_g = 1$ [i.e., $g \in \ker(\psi)$] for all representations $\psi : G \mapsto \{-1, 1\}$. The following proposition establishes such a subgroup for the Ising symmetry group.

Proposition 1. Let Ψ be the set of all real scalar representations $\psi : G \mapsto \{-1, 1\}$ of the Ising symmetry group G. Then $\bigcap_{\psi \in \Psi} \ker(\psi) = \langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle$.

Proof. Let $g \in G$ such that $g = h^2$ for some $h \in G$. Then for every $\psi \in \Psi$, we have

$$\psi_g = \psi_{h^2} = \psi_h^2 = (\pm 1)^2 = 1,$$

and hence $g \in \bigcap_{\psi \in \Psi} \ker(\psi)$; in particular, we have

$$\langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle \subseteq \bigcap_{\psi \in \Psi} \ker(\psi).$$

All that remains to show is the reverse inclusion. Let $g \in \bigcap_{\psi \in \Psi} \ker(\psi)$, so that $\psi_g = 1$ for every $\psi \in \Psi$. By the defining relations of G, g admits the expression $g = h\tau^{m_3}\sigma^{m_4}$, where h is a product of m_1 copies of α and m_2 copies of ρ in some order. Now since ψ takes values in an Abelian group, then for all $g_1, g_2 \in G$,

$$\psi_{g_1 g_2} = \psi_{g_1}\psi_{g_2} = \psi_{g_2}\psi_{g_1} = \psi_{g_2 g_1}.$$

Using this fact, we have

$$\begin{aligned}\psi_g &= \psi_h \psi_{\tau^{m_3}} \psi_{\sigma^{m_4}}\\ &= \psi_{\alpha^{m_1}\rho^{m_2}} \psi_{\tau^{m_3}} \psi_{\sigma^{m_4}}\\ &= \psi_\alpha^{m_1} \psi_\rho^{m_2} \psi_\tau^{m_3} \psi_\sigma^{m_4} = 1. \end{aligned} \tag{A1}$$

This holds for every $\psi \in \Psi$. Now each $\psi \in \Psi$ is completely determined by its values on the generators α, ρ, τ, and σ. Moreover, it is easy to verify that if we apply ψ to every defining relation of G, then the resulting equations are satisfied for all choices of ψ_α, ψ_ρ, ψ_τ, and ψ_σ. Thus, Ψ is precisely the set of homomorphisms $\psi : G \mapsto \{-1, 1\}$ determined by every combination of values in $\{-1, 1\}$ on the four generators of G. Let $\psi_i \in \Psi$ such that ψ_i is -1 on the ith generator and 1 on the other three generators. Evaluating Eq. (A1) at $\psi = \psi_i$, we get $(-1)^{m_i} = 1$, implying that each m_i is even. The element g thus takes the form $g = h\tau^{2n_3}\sigma^{2n_4} = h$, where h is a product of $2n_1$ copies of α and $2n_2$ copies of ρ in some order.

On the other hand, let $\beta = \rho\alpha\rho^{-1}$, and note that α and β commute. Then $G = (\langle \alpha, \beta \rangle \rtimes \langle \rho, \tau \rangle) \times \langle \sigma \rangle$, implying that each $g \in G$ admits the unique representation

$$g = \alpha^{q_1}\beta^{q_2}\rho^{q_3}\tau^{q_4}\sigma^{q_5}.$$

If $\psi_g = 1$, then we have already deduced that g must be a product of α's and ρ's, and hence $g = \alpha^{q_1}\beta^{q_2}\rho^{q_3}$. Noting that $\beta^{q_2} = \rho\alpha^{q_2}\rho^{-1}$, this expression of g has $q_1 + q_2$ copies of α and $1 - 1 + q_3 = q_3$ copies of ρ. Equating these to the previously obtained exponents in the expression of g, we have $q_1 + q_2 = 2n_1$ and $q_3 = 2n_2$, implying that q_3 is even and that q_1 and q_2 have the same parity. Now using the fact that α and β commute, g admits the form

$$\begin{aligned} g &= \alpha^{q_1 - q_2}\alpha^{q_2}\beta^{q_2}\rho^{q_3}\\ &= \alpha^{q_1 - q_2}(\alpha\beta)^{q_2}\rho^{q_3}\\ &= (\alpha^2)^{n_1 - q_2}(\alpha\beta)^{q_2}(\rho^2)^{n_2}, \end{aligned}$$

implying $g \in \langle \alpha^2, \rho^2, \alpha\beta \rangle$. Finally, note that

$$\begin{aligned} \alpha\beta &= \alpha\rho\alpha\rho^{-1}\\ &= \alpha\rho\alpha\rho^3\\ &= \alpha\rho\alpha\rho\rho^2\\ &= (\alpha\rho)^2\rho^2. \end{aligned}$$

Since $\rho^2 \in \langle \alpha^2, \rho^2, \alpha\beta \rangle$, then $\langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle$, and therefore $g \in \langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle$, establishing

$$\bigcap_{\psi \in \Psi} \ker(\psi) \subseteq \langle \alpha^2, \rho^2, (\alpha\rho)^2 \rangle. \qquad \blacksquare$$

Intuitively, even for a general observable dimension d, the observable $\mathcal{O} : \mathbb{X} \mapsto \mathbb{R}^d$ reduces dimensionality from L^2 to d where L may be arbitrarily large in the thermodynamic limit. If d is less than the minimum dimensionality required for ψ to be a faithful representation, then some subgroup of G will necessarily be modded out by ψ.

It can be shown that $\mathrm{SCB}(L) \trianglelefteq G$, and hence we have the quotient group

$$\begin{aligned} G/H &= \langle \alpha H, \rho H, \tau H, \sigma H \rangle\\ &= \{\alpha^{m_1}\rho^{m_2}\tau^{m_3}\sigma^{m_4}H : m_i \in \{0, 1\}\},\\ H &= \mathrm{SCB}(L). \end{aligned} \tag{A2}$$

This quotient group represents the reduction in the number of symmetries we will need to check for SSB; indeed, $|G| = 16L^2$ and $|H| = L^2$, so that $|G/H| = 16$.

2. Constraint on the representations of spatial symmetries

The form of the reduced encoder [Eq. (10)] places additional constraints on the spatial symmetries in the quotient group [Eq. (A2)], as specified in the following proposition.

Proposition 2. Suppose $\check{\mathcal{O}} : [-1, 1]^2 \mapsto \mathbb{R}$ is nonzero for at least one lattice configuration. For every $\psi \in \Psi$ such that $\check{\mathcal{O}}(g\check{x}) = \psi_g \check{\mathcal{O}}(\check{x})$ for all Ising symmetries $g \in G$, we have $\psi_\alpha = \psi_\rho = \psi_\tau$.

Proof. Clearly α (downward translation), ρ (90° rotation), and τ (reflection) map black squares to white squares and vice versa on an $L \times L$ checkerboard where L is even. Thus, for all $g \in \{\alpha, \rho, \tau\}$,

$$\begin{aligned} g(\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}}) &= (\check{x}_{\mathrm{W}}, \check{x}_{\mathrm{B}}),\\ \check{\mathcal{O}}(g(\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})) &= \check{\mathcal{O}}((\check{x}_{\mathrm{W}}, \check{x}_{\mathrm{B}})),\\ \psi_g\check{\mathcal{O}}((\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})) &= \check{\mathcal{O}}((\check{x}_{\mathrm{W}}, \check{x}_{\mathrm{B}})), \end{aligned}$$

and hence,

$$\psi_\alpha\check{\mathcal{O}}((\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})) = \psi_\rho\check{\mathcal{O}}((\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})) = \psi_\tau\check{\mathcal{O}}((\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}})).$$

Evaluating this on a lattice configuration $\mathbf{x}$ for which $\check{\mathcal{O}}$ is nonzero, we obtain $\psi_\alpha = \psi_\rho = \psi_\tau$ as desired. $\blacksquare$

The key idea is that $g(\check{x}_{\mathrm{B}}, \check{x}_{\mathrm{W}}) = (\check{x}_{\mathrm{W}}, \check{x}_{\mathrm{B}})$ for all $g \in \{\alpha, \rho, \tau\}$. The upshot is that we now need only estimate ψ_σ and one of ψ_α, ψ_ρ, and ψ_τ from the data.

3. Unsupervised learning of symmetries

The following proposition states that under suitable conditions, if an unsupervised model is fit to a data set containing symmetries, then the fit model will be invariant to those symmetries at least when restricted to the data set.

Proposition 3. Let G be a finite group and $\mathcal{X} = \{\mathbf{x}_n \in \mathbb{R}^m\}_{n=1}^N$ a G-invariant data set; i.e., $g\mathbf{x}_n \in \mathcal{X}$ for every $g \in G$

and $\mathbf{x}_n \in \mathcal{X}$. Let $\mathcal{F}$ be a convex set of functions $f : \mathcal{X} \mapsto \mathbb{R}^m$ such that $gfg^{-1} \in \mathcal{F}$ for every $f \in \mathcal{F}$ and $g \in G$. Let $L_{\text{metric}} : \mathbb{R}^m \times \mathcal{X} \mapsto \mathbb{R}$ be a function such that

(1) $L_{\text{metric}}(g\mathbf{x}, g\mathbf{x}') = L_{\text{metric}}(\mathbf{x}, \mathbf{x}')$ for all $g \in G$ and $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^m \times \mathcal{X}$, and

(2) L_{metric} is strictly convex in its first argument.

Then every global minimizer $f_* \in \mathcal{F}$ of the loss function $\mathcal{L} : \mathcal{F} \mapsto \mathbb{R}$ given by

$$\mathcal{L}(f) = \frac{1}{N}\sum_{n=1}^{N} L_{\text{metric}}(f(\mathbf{x}_n), \mathbf{x}_n)$$

is G-equivariant on the data set $\mathcal{X}$.

Proof. First we establish that the loss function $\mathcal{L}$ is both strictly convex and G-invariant. For each $n = 1, \dots, N$, it is easy to show that the map from $\mathcal{F}$ to $\mathbb{R}$ given by $f \to L_{\text{metric}}(f(\mathbf{x}_n), \mathbf{x}_n)$ is strictly convex. [Note that this relies on the hypothesis that the domain of the functions in $\mathcal{F}$ is the data set $\mathcal{X}$, so that for $f_1, f_2 \in \mathcal{F}$, $f_1 = f_2$ if and only if $f_1(\mathbf{x}_n) = f_2(\mathbf{x}_n)$ for all $n = 1, \dots, N$.] Moreover, since $\mathcal{L}$ is a convex combination of such maps, then it is also strictly convex. Now for every $f \in \mathcal{F}$ and $g \in G$, we have

$$\begin{aligned}
\mathcal{L}(gfg^{-1}) &= \frac{1}{N}\sum_{n=1}^{N} L_{\text{metric}}(gf(g^{-1}\mathbf{x}_n), \mathbf{x}_n) \\
&= \frac{1}{N}\sum_{n=1}^{N} L_{\text{metric}}(gf(g^{-1}\mathbf{x}_n), g(g^{-1}\mathbf{x}_n)) \\
&= \frac{1}{N}\sum_{n=1}^{N} L_{\text{metric}}(gf(\mathbf{x}_n), g\mathbf{x}_n) \quad (g^{-1}\mathbf{x}_n \to \mathbf{x}_n) \\
&= \frac{1}{N}\sum_{n=1}^{N} L_{\text{metric}}(f(\mathbf{x}_n), \mathbf{x}_n) \\
&= \mathcal{L}(f),
\end{aligned}$$

where we used the fact that the data set $\mathcal{X}$ is G-invariant in the reindexing step. Thus, $\mathcal{L}$ is G-invariant, where each $g \in G$ acts on $\mathcal{L}$ by conjugation of its argument.

Now let $f_* \in \mathcal{F}$ be a global minimizer of $\mathcal{L}$. By G invariance, gf_*g^{-1} is also a global minimizer for every $g \in G$, and by convexity, $\frac{f_*+gf_*g^{-1}}{2}$ is a global minimizer for each $g \in G$ as well. On the other hand, convexity and G invariance together imply that

$$\begin{aligned}
\mathcal{L}\left(\frac{f_* + gf_*g^{-1}}{2}\right) &\leqslant \frac{\mathcal{L}(f_*) + \mathcal{L}(gf_*g^{-1})}{2} \\
&= \frac{\mathcal{L}(f_*) + \mathcal{L}(f_*)}{2} \\
&= \mathcal{L}(f_*),
\end{aligned}$$

but since f_* and $\frac{f_*+gf_*g^{-1}}{2}$ are both global minimizers, then this inequality is in fact an equality:

$$\mathcal{L}\left(\frac{f_* + gf_*g^{-1}}{2}\right) = \frac{\mathcal{L}(f_*) + \mathcal{L}(gf_*g^{-1})}{2}.$$

Finally, since $\mathcal{L}$ is strictly convex, then this necessitates $f_* = gf_*g^{-1}$ or equivalently $f(g\mathbf{x}_n) = gf(\mathbf{x}_n)$ for all $g \in G$ and $\mathbf{x}_n \in \mathcal{X}$, thereby proving the claim. ■

Note that the functions in $\mathcal{F}$ are restricted to the data set $\mathcal{X}$; the minimizer f_* is therefore not guaranteed to be G equivariant off the data set.

For our particular case, the function L_{metric} is the binary cross-entropy L_{BCE} [Eq. (8)]. By its form in Eq. (8), L_{BCE} is clearly invariant under the spin-flip symmetry $(\hat{\mathbf{x}}, \mathbf{x}) \to (-\hat{\mathbf{x}}, -\mathbf{x})$, and it is manifestly invariant under all spatial symmetries as it is a sum over the lattice sites; thus, L_{BCE} is G invariant where G is the Ising symmetry group. The strict convexity of L_{BCE} is evident; since $x_{\mathbf{i}} = \pm 1$, then each summand in Eq. (8) is either $-\log(\frac{1+\hat{x}_{\mathbf{i}}}{2})$ or $-\log(\frac{1-\hat{x}_{\mathbf{i}}}{2})$, and in either case, its second derivative with respect to $\hat{x}_{\mathbf{i}}$ is strictly positive, so that each summand is strictly convex. The set $\mathcal{F}$ is the set of functions (restricted to our data set) expressible as autoencoders of fixed depth but arbitrary widths; we allow for arbitrary widths to ensure that any convex combination of autoencoders is again expressible as a single autoencoder in $\mathcal{F}$. In practice, however, we make the assumption that the global minimizer $f_* \in \mathcal{F}$ can be accessed by the single autoencoder architecture we proposed. There are additional caveats to Proposition 3: First, our data set of lattice sets is probably only approximately G-invariant; second, actually finding f_* is nontrivial since the map from its network parameters to the autoencoder f is in general nonconvex. Nevertheless, we take Proposition 3 as strong motivation to make the assumption that our autoencoder $f = \mathcal{D} \circ \mathcal{O}$ is G-invariant.

APPENDIX B: NETWORK INITIALIZATION AND LEARNING RATES

Here we provide details on the network parameter initialization and learning rate settings needed for a fair comparison of the baseline autoencoder and GE autoencoder. The GE autoencoder is a small network that is equivalent to the larger baseline autoencoder with the parameters of its first encoding layer and last decoding layer constrained to a checkerboard pattern (ignoring the symmetry regularization terms in the GE-autoencoder loss function). The idea is to initialize and set the learning rates of the baseline autoencoder such that, if the checkerboard constraint were imposed and maintained on the baseline autoencoder, then it would be and would remain functionally equivalent to the GE autoencoder at initialization time and throughout training.

1. Initialization

We initialize the weight matrix and bias vector of each layer of the GE autoencoder with IID values sampled under a uniform distribution over $[-\frac{1}{\sqrt{h_{\text{in}}}}, \frac{1}{\sqrt{h_{\text{in}}}}]$, where h_{in} is the input dimension of the layer. If (u_k, v_k) are the initial weights of the first encoding layer of the GE-autoencoder incident to the kth hidden neuron, then we initialize the weights of the first encoding layer of the baseline autoencoder according to Eq. (9). We initialize the weight matrix and bias vector of the last decoding layer of the baseline autoencoder by simply tiling the initialized parameters of the last decoding layer of the GE autoencoder according to a checkerboard pattern. All remaining parameters of the baseline autoencoder are structurally equivalent to those of the GE autoencoder, and we thus initialize them to be equal to the corresponding initialized parameters in the GE autoencoder.

2. Learning rate

We set the learning rate of all parameters of the GE autoencoder to $\eta_{\mathrm{GE}} = 0.001$. We proceed to deduce the appropriate learning rate η for the weights in the first encoding layer of the baseline autoencoder; we do this for the weights on "black squares" [$i_x + i_y$ even; see Eq. (9)]; the argument for "white squares" is analogous. Assuming full-batch gradient descent for simplicity and ignoring symmetry regularization, the update rules for the weights in the first encoding layers of the baseline autoencoder and GE autoencoder are

$$\Delta w_{k,\mathbf{i}} = -\eta \frac{\partial \mathcal{L}}{\partial w_{k,\mathbf{i}}} \text{and} \Delta u_k = -\eta_{\mathrm{GE}} \frac{\partial \mathcal{L}}{\partial u_k}.$$

However, by Eq. (9), we have $\Delta w_{k,\mathbf{i}} = \frac{2}{L^2}\Delta u_k$ and hence

$$\eta \frac{\partial \mathcal{L}}{\partial w_{k,\mathbf{i}}} = \frac{2}{L^2}\eta_{\mathrm{GE}} \frac{\partial \mathcal{L}}{\partial u_k}.$$

Applying the chain rule to the right side, we have

$$\begin{aligned}\eta \frac{\partial \mathcal{L}}{\partial w_{k,\mathbf{i}}} &= \frac{2}{L^2}\eta_{\mathrm{GE}} \sum_{j_x+j_y \text{even}} \frac{\partial \mathcal{L}}{\partial w_{k,\mathbf{j}}} \frac{\partial w_{k,\mathbf{j}}}{\partial u_k} \\ &= \frac{2}{L^2}\eta_{\mathrm{GE}} \left(\frac{2}{L^2} \sum_{j_x+j_y \text{even}} \frac{\partial \mathcal{L}}{\partial w_{k,\mathbf{j}}} \right).\end{aligned}$$

The quantity in parentheses is the average gradient over black squares. By averaging both sides over $\mathbf{j}$ with $j_x + j_y$ even, we can obtain such an average gradient on both sides; cancelling then leaves us with the desired learning rate:

$$\eta = \frac{2}{L^2}\eta_{\mathrm{GE}}.$$

As an example, for lattice size $L = 128$, the learning rate of the weights in the first encoding layer of the baseline autoencoder should be set to approximately 1.2×10^{-7}, a value unlikely to be found by hand. By the same argument as above, we can show that the learning rate of the parameters in the last decoding layer of the baseline autoencoder should be set equal to η_{GE}. Finally, all remaining parameters of the baseline autoencoder are structurally equivalent to those of the GE autoencoder, and hence we set all of their learning rates equal to η_{GE} as well.

APPENDIX C: FUNCTIONAL COMPARISON TO MAGNETIZATION AND ONSAGER'S SOLUTION

Here we regard the order observables (magnetization, baseline encoder, and GE encoder) as elements of a Hilbert space and measure the distance from each of them to magnetization. Given any Euclidean vector space $\mathbb{R}^n$ with inner product $\langle \cdot, \cdot \rangle$, we define a measure of error $\nu : \mathbb{R}^n \times \mathbb{R}^n \mapsto [0, 1]$ as follows:

$$\nu(\mathbf{v}, \mathbf{w}) = \sin^2\theta = 1 - \left(\frac{\langle \mathbf{v}, \mathbf{w} \rangle}{\|\mathbf{v}\| \|\mathbf{w}\|} \right)^2,$$

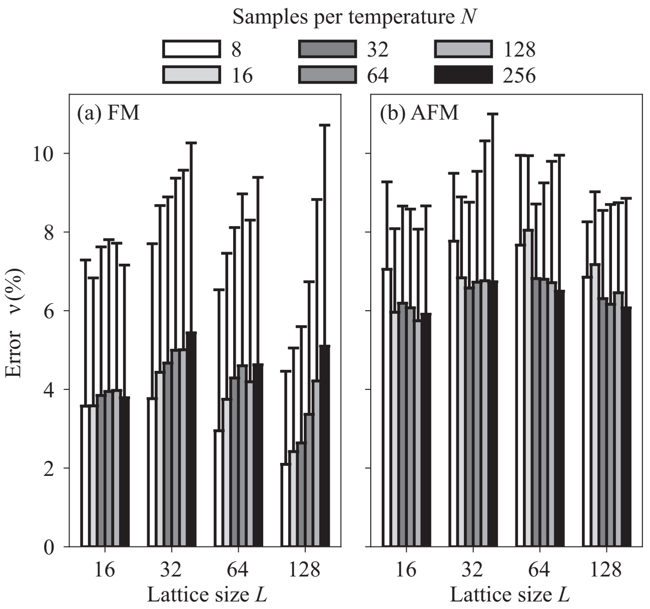

FIG. 13. Measure of error between the baseline encoder and magnetization applied to the (a) ferromagnetic (FM) and (b) antiferromagnetic (AFM) models. Error bars indicate the standard deviation across 24 trials (eight training data folds and three initialization seeds). The baseline encoder is distinct from magnetization (positive error ν) and becomes increasingly distinct with increasing number of training-validation samples in the ferromagnetic case.

where θ is the angle between vectors $\mathbf{v}$ and $\mathbf{w}$. As ν takes values in [0, 1], we will express it as a percentage. Note that $\nu(\mathbf{v}, \mathbf{w})$ is independent of the signs and norms of $\mathbf{v}$ and $\mathbf{w}$—a desirable property for our purpose. We evaluate each order observable $\mathcal{O}$ on a test set of $n = 204\,800$ lattice configurations (2048 configurations sampled at each of 100 temperatures) to obtain a vector of measurements $\hat{\mathcal{O}} \in \mathbb{R}^n$. We then evaluate $\nu(\hat{\mathcal{O}}, \hat{\mathcal{O}}_M)$ for each order observable $\mathcal{O}$, where $\mathcal{O}_M$ is magnetization. We trivially have $\nu(\hat{\mathcal{O}}_M, \hat{\mathcal{O}}_M) = 0$, but we also obtain an error around 0.0%–0.1% or 0.001 for both the single-scale and multiscale GE encoders across all lattice sizes and training-validation sample sizes, indicating that the GE encoders learn a function very similar to magnetization. In contrast, the baseline-encoder results in a larger error and is thus more distinct from magnetization (Fig. 13); in the ferromagnetic case, this error even increases with training-validation sample size, meaning that the baseline encoder moves away from magnetization as it sees more data. This is also consistent with the divergence between the baseline encoder and magnetization in terms of their critical temperature estimates (Fig. 7).

We now compare the order parameter $\langle|\mathcal{O}|\rangle$ derived from each order observable (Fig. 5) to Onsager's solution M_{ONS} [Eq. (14)]. We evaluate the order parameters and Onsager's solution on the 100 temperatures in our data set to obtain vectors in $\mathbb{R}^{100}$. We then evaluate $\nu(\langle|\mathcal{O}|\rangle, M_{\mathrm{ONS}})$ independently for each jackknife-sample of the order parameter $\langle|\mathcal{O}|\rangle$. We perform least-squares regression on the calculated error ν against inverse lattice size, finding an approximately linear relationship with the baseline encoder having a slightly weaker linear dependence (in terms of r^2 value) than

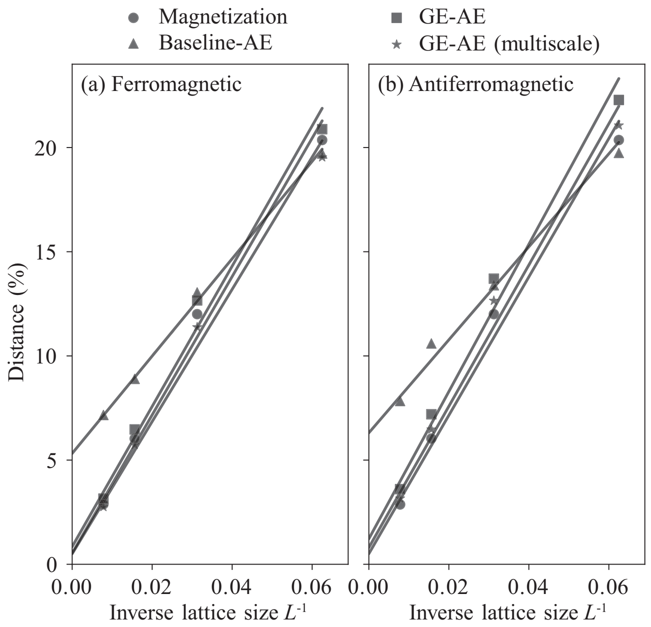

FIG. 14. Measure of error between the order parameters derived from each order observable and Onsager's solution using $N = 256$ training-validation samples per temperature and averaged over 24 trials (eight training data folds and three initialization seeds). Results for the ferromagnetic model are shown in panel (a), and results for the antiferromagnetic model are shown in panel (b). The error exhibits an approximately linear dependence on inverse lattice size, with stronger linearity for magnetization and the GE encoders. At infinite lattice size ($L^{-1} = 0$), magnetization and the GE encoders converge more closely to Onsager's solution than does the baseline encoder.

magnetization and the GE encoders; we visualize this for the maximal case of $N = 256$ training-validation samples per temperature (Fig. 14). Extrapolating to infinite lattice size (i.e., the thermodynamic limit), we find that the order parameters derived from magnetization and the GE encoders almost converge to Onsager's solution, while the baseline encoder converges further away from Onsager's solution.

APPENDIX D: OBTAINING POINT ESTIMATES OF THE CRITICAL TEMPERATURE

Here we provide the details of obtaining a point estimate of the critical temperature given interval estimates as described in Sec. IV C. Let $T_1 < T_2 < \cdots < T_M$ be the temperatures at which we ran MC simulations to generate our data set, and suppose we have N MC-sampled lattice configurations from each of these temperatures. Let $t_0 \in [T_{m_0}, T_{m_0+1}]$ be the interval estimate of the critical temperature obtained from the Binder cumulant curve based on all available data, in the way described in Sec. IV C, and let $t_i \in [T_{m_i}, T_{m_i+1}]$ be the interval estimate of the critical temperature obtained from the ith jackknife-sample Binder cumulant curve, for $i = 1, \dots, N$. We seek the critical temperature point estimate that is optimally stable; we do so by minimizing the jackknife variance subject to the interval estimates:

$$\min_{t_0,\dots,t_N} \sum_{i=1}^{N} (t_i - t_0)^2 \text{s.t.}$$

$$T_{m_i} \leqslant t_i \leqslant T_{m_i+1}, \text{ for } i = 0, \dots, N.$$

This problem can be expressed more elegantly with vector notation. Let $\mathbf{t}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^{N+1}$ with elements t_i, T_{m_i}, and T_{m_i+1} respectively, and define the matrix $\mathbf{A} \in \mathbb{R}^{(N+1)\times(N+1)}$ with elements

$$\begin{aligned} A_{00} &= N, \\ A_{i0} &= A_{0i} = -1, \\ A_{ii} &= 1, \text{ for } i = 1, \dots, N, \end{aligned}$$

and zero for all remaining elements. Then the above optimization problem can be expressed as

$$\min_{\mathbf{t}} \mathbf{t}^{\top}\mathbf{A}\mathbf{t}\text{s.t.}\mathbf{a} \leqslant \mathbf{t} \leqslant \mathbf{b}.$$

The matrix $\mathbf{A}$ admits the factorization $\mathbf{A} = \mathbf{B}^{\top}\mathbf{B}$, where $\mathbf{B} \in \mathbb{R}^{N\times(N+1)}$ with block structure $\mathbf{B} = [\mathbf{1} \mid \mathbf{I}]$, where $\mathbf{1}$ is an N-dimensional vector of 1's and $\mathbf{I}$ is the $N \times N$ identity matrix. The matrix $\mathbf{B}$ clearly has rank N, and hence $\mathbf{A}$ is a symmetric positive semidefinite matrix of corank 1. The above constrained optimization problem is thus a convex quadratic program, which we efficiently solve numerically using the coneqp solver available in CVXOPT [61].

Let $\mathbf{t}_{\text{sol}}$ be the numerical solution obtained. Since $\mathbf{A}$ has corank 1, then the solution set of the optimization problem is at most a line segment (one degree of freedom). It is easy to see that if $\mathbf{t}_{\text{sol}}$ is an interior solution, then $\mathbf{t}_{\text{sol}} \pm \varepsilon\mathbf{1}$ is a solution as well for sufficiently small $\varepsilon > 0$, where $\mathbf{1}$ is an $(N+1)$-dimensional vector of 1's. If we set

$$\varepsilon_1 = \min(\mathbf{t}_{\text{sol}} - \mathbf{a})\text{and}\varepsilon_2 = \min(\mathbf{b} - \mathbf{t}_{\text{sol}}),$$

then the complete solution set is

$$\{(1-\alpha)(\mathbf{t}_{\text{sol}} - \varepsilon_1\mathbf{1}) + \alpha(\mathbf{t}_{\text{sol}} + \varepsilon_2\mathbf{1}) : 0 \leqslant \alpha \leqslant 1\}.$$

We select the midpoint $\mathbf{t}_*$ as the vector of jackknife critical temperature point estimates.

The jackknife mean and variance of the critical temperature estimate are then

$$\text{mean} = \overline{t_*} + B,$$

$$\text{variance} = \sum_{i=1}^{N} (t_{*i} - \overline{t_*})^2 + \frac{B^2}{N},$$

where $\overline{t_*} = \frac{1}{N}\sum_{i=1}^{N} t_{*i}$ is the mean of the jackknife samples (not including the overall estimate based on all N samples), and $B = N(t_0 - \overline{t_*})$ is a term added to reduce the bias in the jackknife mean, at the cost of incurring additional variance. Note that the above expression for the variance is equivalent to the objective function of the optimization problem we solved. As already discussed in Sec. IV C, our estimates of B were unstable due to the nondifferentiability of our underlying critical temperature estimator; we therefore set $B = 0$ in the above equations to obtain our final expressions for the mean critical temperature point estimate and variance.

APPENDIX E: VECTOR ORDER OBSERVABLES

1. A 2D order observable

Before stating Proposition 4, we need the following lemma, which characterizes the real orthogonal square and fourth roots of the 2×2 identity matrix.

Lemma 1. Let A be a 2×2 real orthogonal matrix such that $A^4 = I$, where I denotes the 2×2 identity matrix.

(a) If $A^2 = I$, then

$$A = \pm I \quad \text{or} \quad A = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}, \quad \theta \in [0, 2\pi).$$

(b) If $A^2 \neq I$, then

$$A = \pm\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

(c) $A^2 = \pm I$.

Proof. Since A is orthogonal, then it has one of the two following forms for some real θ:

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \cos\theta & \sin\theta \end{bmatrix} \text{or} \tag{*}$$

$$A = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}. \tag{**}$$

To prove (a), suppose $A^2 = I$. If A has the form (**), then it is easy to check that $A^2 = I$ holds trivially. Suppose instead A has the form (*). Then $A^2 = I$ necessitates

$$\cos^2\theta - \sin^2\theta = 1,$$
$$\cos^2\theta - (1 - \cos^2\theta) = 1,$$
$$\cos^2\theta = 1,$$

and hence $\sin^2\theta = 0$. By (*), this implies $A = \pm I$ as claimed in (a).

To prove (b), suppose $A^2 \neq I$. Since $A^4 = (A^2)^2 = I$, then by (a), we must have either $A^2 = -I$ or A^2 has the following form for some real ϕ:

$$A^2 = \begin{bmatrix} \cos\phi & \sin\phi \\ \sin\phi & -\cos\phi \end{bmatrix}. \tag{***}$$

First, suppose A^2 has the form (***) and A the form (*). Then equating the diagonal terms of (*) squared and (***), and doing similar with the off-diagonal terms, we obtain the equations

$$\cos^2\theta - \sin^2\theta = \pm\cos\phi,$$
$$2\cos\theta\sin\theta = \pm\sin\phi.$$

These equations imply

$$\cos^2\theta - \sin^2\theta = \pm 0,$$
$$2\cos\theta\sin\theta = \pm 0,$$

and hence $\cos\theta = \sin\theta = 0$, which is impossible.

Suppose on the other hand A^2 has the form (***) and A the form (**). Equating the diagonal terms of (***) squared and (**), we obtain

$$(\pm\cos\theta)^2 + \sin^2\theta = \pm\cos\phi,$$
$$1 = \pm\cos\phi,$$

which is also impossible. Thus, A^2 cannot have the form (***).

We now turn to the case $A^2 = -I$. If $A^2 = -I$ and A has the form (**), then $\cos^2\theta + \sin^2\theta = -1$, which is impossible. On the other hand, if $A^2 = -I$ and A has the form (*), then

$$\cos^2\theta - \sin^2\theta = -1,$$
$$\cos^2\theta - (1 - \cos^2\theta) = -1,$$
$$2\cos^2\theta - 1 = -1,$$
$$\cos^2\theta = 0,$$

and hence $\sin^2\theta = 1$. The form (*) thus implies the expression for A claimed in (b).

Finally, to prove (c), simply observe if $A^2 = I$, then we are done; otherwise, A has the form given in (b), whose square is $-I$. ■

We now state and prove Proposition 4, which gives the subgroup of never-broken symmetries for a 2D order observable. Let $\beta = \rho\alpha\rho^{-1}$; i.e., the generator of horizontal translations. Let $O(2, \mathbb{R})$ be the group of 2×2 real orthogonal matrices.

Proposition 4. Let Ψ be the set of all real 2D representations $\psi : G \mapsto O(2, \mathbb{R})$ of the Ising symmetry group G. Then $\bigcap_{\psi\in\Psi} \ker(\psi) = \langle\alpha^2, \beta^2\rangle$.

Proof. Let I be the 2×2 identity matrix, and let $\psi \in \Psi$. Since $\rho^4 = 1$ by definition, then $\psi_\rho^4 = I$. By Lemma 1 (c), $\psi_\rho^2 = \pm I$. The defining relation $\alpha\rho^2 = \rho^2\alpha^{-1}$ thus implies

$$\psi_\alpha\psi_\rho^2 = \psi_\rho^2\psi_\alpha^{-1},$$
$$\psi_\alpha = \psi_\alpha^{-1},$$
$$\psi_\alpha^2 = I.$$

Ergo, $\alpha^2 \in \ker(\psi)$. We can similarly show $\beta^2 \in \ker(\psi)$, and hence

$$\langle\alpha^2, \beta^2\rangle \leqslant \bigcap_{\psi\in\Psi} \ker(\psi).$$

All that remains is to prove the reverse inclusion.

Let Φ be the set of all real scalar representations $\phi : G \mapsto \{-1, 1\}$, and define the set

$$\Psi_1 = \{g \mapsto \phi_g I : \phi \in \Phi\}.$$

Then by Proposition 1, we have

$$\bigcap_{\psi\in\Psi} \ker(\psi) \leqslant \bigcap_{\psi\in\Psi_1} \ker(\psi) \leqslant \langle\alpha^2, \rho^2, (\alpha\rho)^2\rangle.$$

If we can show

$$\rho^2, \alpha\beta \notin \bigcap_{\psi\in\Psi} \ker(\psi), \tag{*}$$

then this will imply

$$\bigcap_{\psi\in\Psi} \ker(\psi) \leqslant \langle\alpha^2, \beta^2\rangle,$$

which will then establish the claim. Consider $\psi \in \Psi$ defined such that

$$\psi_\alpha = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\psi_\rho = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Note these are valid representations as $\psi_\alpha \psi_\rho^2 = \psi_\rho^2 \psi_\alpha^{-1}$. Then we have $\psi_\rho^2 = -I$ as well as

$$\begin{aligned}\psi_\alpha \psi_\beta &= \psi_\alpha \psi_\rho \psi_\alpha \psi_\rho^{-1}\\ &= -I,\end{aligned}$$

so that $\rho^2, \alpha\beta \notin \ker(\psi)$. This establishes (*) and hence the proposition. ■

Recall the description of the GE-autoencoder architecture in Sec. V A, where the input into the network is a 4D block average $\check{x}$ of a lattice configuration $\mathbf{x}$. The following proposition states that our particular choice of architecture places an additional constraint on the representation ψ.

Proposition 5. The GE encoder described in Sec. V A satisfies $\psi_\alpha \check{\mathcal{O}} = \psi_{\rho^2\tau} \check{\mathcal{O}}$.

Proof. Let $\check{x} = (\check{x}_1, \check{x}_2, \check{x}_3, \check{x}_4)$ be the block average of a lattice configuration $\mathbf{x}$ over nonoverlapping 2×2 blocks. Then

$$\begin{aligned}\alpha\check{x} &= (\check{x}_3, \check{x}_4, \check{x}_1, \check{x}_2),\\ \rho\check{x} &= (\check{x}_2, \check{x}_4, \check{x}_1, \check{x}_3),\\ \tau\check{x} &= (\check{x}_2, \check{x}_1, \check{x}_4, \check{x}_3).\end{aligned}$$

Based on these permutations, it is easy to verify $\alpha\check{x} = \rho^2\tau\check{x}$. Thus,

$$\begin{aligned}\check{\mathcal{O}}(\alpha\check{x}) &= \check{\mathcal{O}}(\rho^2\tau\check{x}),\\ \psi_\alpha \check{\mathcal{O}} &= \psi_{\rho^2\tau} \check{\mathcal{O}},\end{aligned}$$

completing the proof. ■

2. Never-broken symmetries in terms of characters

The following proposition gives a way to compute the subgroup of never-broken symmetries of an arbitrary finite group G in terms of its character table.

Proposition 6. Let G be a finite group. Let Ψ_d be the set of all real orthogonal representations of G with degree d. Let $\hat{G}$ be the set of all irreducible characters of G. Then

$$\bigcap_{\psi \in \Psi_d} \ker(\psi) = \bigcap_{\chi \in \hat{G} | \deg_{\mathbb{R}}(\chi) \leqslant d} \ker(\chi).$$

Proof. Let $\chi \in \hat{G}$ such that $\deg_{\mathbb{R}}(\chi) \leqslant d$. Let ψ be the irreducible representation with character χ. Let $\psi_{\mathbb{R}}$ be the smallest real representation built out of ψ:

$$\psi_{\mathbb{R}} = \begin{cases} \psi, & \text{if} \quad \mathrm{I}_{\mathrm{FS}}(\chi) = 1\\ \psi \oplus \overline{\psi}_\chi, & \text{if} \quad \mathrm{I}_{\mathrm{FS}}(\chi) = 0\\ \psi \oplus \psi, & \text{if} \quad \mathrm{I}_{\mathrm{FS}}(\chi) = -1. \end{cases}$$

Note $\deg(\psi_{\mathbb{R}}) = \deg_{\mathbb{R}}(\chi) \leqslant d$. Let $n = d - \deg(\psi_{\mathbb{R}}) \geqslant 0$ and ψ_1 the trivial representation. Then construct the representation

$$\psi_\chi = \psi_{\mathbb{R}} \oplus \bigoplus_{i=1}^{n} \psi_1.$$

The kernel of this representation is clearly $\ker(\psi_\chi) = \ker(\psi_{\mathbb{R}}) = \ker(\chi)$. Moreover, $\deg(\psi_\chi) = d$ so that $\psi_\chi \in \Psi_d$. Since a real representation $\psi_\chi \in \Psi_d$ can be constructed for every $\chi \in \hat{G} \mid \deg_{\mathbb{R}}(\chi) \leqslant d$, then

$$\bigcap_{\psi \in \Psi_d} \ker(\psi) \leqslant \bigcap_{\chi \in \hat{G} | \deg_{\mathbb{R}}(\chi) \leqslant d} \ker(\chi). \qquad (*)$$

All that remains is to prove the reverse inclusion as well.

Let $\psi \in \Psi_d$. Then ψ admits the decomposition

$$\psi = \psi_1 \oplus \cdots \oplus \psi_k,$$

where each ψ_i is a real-irreducible representation (i.e., irreducible over $\mathbb{R}$). For each $i \in \{1, \ldots, k\}$, there exists a (complex-)irreducible representation ψ_i' such that $\psi_i = \psi_i'$, $\psi_i = \psi_i' \oplus \overline{\psi'}_i$, or $\psi_i = \psi_i' \oplus \psi_i'$. Since $\ker(\psi_i') = \ker(\overline{\psi'}_i)$, then in any of these three cases, we have $\ker(\psi_i) = \ker(\psi_i')$. Thus,

$$\ker(\psi) = \bigcap_{i=1}^{k} \ker(\psi_i) = \bigcap_{i=1}^{k} \ker(\psi_i').$$

Now let χ_i and χ_i' be the characters of ψ_i and ψ_i' respectively. Clearly, $\deg_{\mathbb{R}}(\chi_i') = \deg(\chi_i) \leqslant d$ since $\deg(\psi_i) \leqslant \deg(\psi) = d$ for each $i \in \{1, \ldots, k\}$. Thus,

$$\ker(\psi) = \bigcap_{i=1}^{k} \ker(\chi_i'),$$

where each χ_i' is irreducible and $\deg_{\mathbb{R}}(\chi_i') \leqslant d$. This establishes the reverse inclusion of (*). ■

3. Symmetry regularization

Here we relate the general symmetry regularization terms R_g and S_g [Eqs. (21) and (22)] to the first and second regularization terms in Eq. (12) for 1D order observables. This will provide insight into how Eq. (20) generalizes Eq. (12). We use the notation as in Eqs. (19)–(22).

First, however, we simplify the expression for S_g [Eq. (22)] for arbitrary order dimension d.

Lemma 2. The regularization term S_g [Eq. (22)] admits the expression

$$S_g = 1 - \frac{\|ZZ^+Z_g\|_F^2}{\|Z_g\|_F^2}.$$

Proof. Recalling the expression [Eq. (19)] for the linear least-squares estimator $\hat{\psi}_g$, the squared residual is

$$\begin{aligned}\|Z_g - Z\hat{\psi}_g\|_F^2 &= \|Z_g - Z(Z^+Z_g + PA)\|_F^2\\ &= \|Z_g - ZZ^+Z_g\|_F^2\\ &= \|(I - ZZ^+)Z_g\|_F^2.\end{aligned}$$

Since $I - ZZ^+$ is an orthogonal projection operator, then by the Pythagorean theorem we obtain

$$\|Z_g - Z\hat{\psi}_g\|_F^2 = \|Z_g\|_F^2 - \|ZZ^+Z_g\|_F^2,$$

and thus

$$S_g = 1 - \frac{\|ZZ^+Z_g\|_F^2}{\|Z_g\|_F^2},$$

completing the proof. ■

We now derive the simplified expressions for the regularization terms R_g and S_g for 1D order observables.

Proposition 7. Let $d = 1$. Then the regularization terms R_g and S_g [Eqs. (21) and (22)] admit the expressions

$$R_g = \left(1 - \frac{\|Z_g\|^2}{\|Z\|^2}(1 - S_g)\right)^2,$$

$$S_g = 1 - \left(\frac{Z^\top Z_g}{\|Z\|\|Z_g\|}\right)^2.$$

Proof. We first consider S_g. By Lemma 2,

$$S_g = 1 - \frac{\|ZZ^+Z_g\|^2}{\|Z_g\|^2},$$

where we replaced the Frobenius matrix norm with the usual vector norm since Z and Z_g are now N-dimensional column vectors. Since ZZ^+ is an orthogonal projection operator, then $(ZZ^+)^\top(ZZ^+) = ZZ^+$ so that

$$S_g = 1 - \frac{Z_g^\top ZZ^+Z_g}{\|Z_g\|^2}.$$

Now unless the GE encoder is exactly zero on every sampled lattice configuration, Z is a nonzero column vector and thus full rank. Its pseudoinverse is thus the row vector

$$Z^+ = \frac{Z^\top}{\|Z\|^2}.$$

Substituting this into our expression for S_g, we obtain

$$S_g = 1 - \frac{Z_g^\top ZZ^\top Z_g}{\|Z\|^2\|Z_g\|^2}$$
$$= 1 - \left(\frac{Z^\top Z_g}{\|Z\|\|Z_g\|}\right)^2,$$

as claimed.

We now move to R_g. Since Z is full rank, then its null space is trivial so that $P = 0$. The estimator $\hat{\psi}_g$ [Eq. (19)] thus simplifies to

$$\hat{\psi}_g = Z^+Z_g + 0$$
$$= \frac{Z^\top Z_g}{\|Z\|^2}$$
$$= \frac{\|Z_g\|}{\|Z\|}\frac{Z^\top Z_g}{\|Z\|\|Z_g\|}.$$

Substituting this into Eq. (21) and noting $\hat{\psi}_g$ is a scalar, we have

$$R_g = (1 - \hat{\psi}_g^2)^2$$
$$= \left(1 - \frac{\|Z_g\|^2}{\|Z\|^2}\left(\frac{Z^\top Z_g}{\|Z\|\|Z_g\|}\right)^2\right)^2$$
$$= \left(1 - \frac{\|Z_g\|^2}{\|Z\|^2}(1 - S_g)\right)^2,$$

completing the proof. ■

Observe that the expressions for R_g and S_g in Proposition 7 match the first and second regularization terms in Eq. (12), except for the factor $(1 - S_g)$ in R_g. Still, in the optimal case $S_g = 0$, the expression for R_g matches the first regularization term in Eq. (12).

[1] N. W. Ashcroft and N. D. Mermin, *Solid State Physics* (Holt-Saunders, New York, 1976).

[2] S. Friedli and Y. Velenik, *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction* (Cambridge University Press, Cambridge, UK, 2017).

[3] H. Gomez, M. Bures, and A. Moure, A review on computational modelling of phase-transition problems, Phil. Trans. R. Soc. A. **377**, 20180203 (2019).

[4] Á. Nagy, M. Calixto, and E. Romera, A density functional theory view of quantum phase transitions, J. Chem. Theory Comput. **9**, 1068 (2013).

[5] L.-A. Wu, M. S. Sarandy, D. A. Lidar, and L. J. Sham, Linking entanglement and quantum phase transitions via density-functional theory, Phys. Rev. A **74**, 052335 (2006).

[6] A. Nagy and E. Romera, Quantum phase transitions via density-functional theory: Extension to the degenerate case, Phys. Rev. A **88**, 042515 (2013).

[7] C. Shahi, J. Sun, and J. P. Perdew, Accurate critical pressures for structural phase transitions of group IV, III–V, and II–VI compounds from the SCAN density functional, Phys. Rev. B **97**, 094111 (2018).

[8] R. J. Maurer, C. Freysoldt, A. M. Reilly, J. G. Brandenburg, O. T. Hofmann, T. Björkman, S. Lebègue, and A. Tkatchenko, advances in density-functional calculations for materials modeling, Annu. Rev. Mater. Res. **49**, 1 (2019).

[9] J. M. Haile, *Molecular Dynamics Simulation: Elementary Methods* (John Wiley & Sons, New York, 1992).

[10] R. Sasaki, Y. Hayashi, and S. Kawauchi, Acceleration of liquid-crystalline phase transition simulations using selectively scaled and returned molecular dynamics, J. Chem. Inf. Model. **60**, 3499 (2020).

[11] K. V. Shanavas and S. M. Sharma, Molecular dynamics simulations of phase transitions in argon-filled single-walled carbon nanotube bundles under high pressure, Phys. Rev. B **79**, 155425 (2009).

[12] M. Imada, A. Fujimori, and Y. Tokura, Metal-insulator transitions, Rev. Mod. Phys. **70**, 1039 (1998).

[13] D. C. Johnston, The puzzle of high temperature superconductivity in layered iron pnictides and chalcogenides, Adv. Phys. **59**, 803 (2010).

[14] G. R. Stewart, Unconventional superconductivity, Adv. Phys. **66**, 75 (2017).

[15] B. Keimer, S. A. Kivelson, M. R. Norman, S. Uchida, and J. Zaanen, From quantum matter to high-temperature superconductivity in copper oxides, Nature (London) **518**, 179 (2015).

[16] Y. Zhou, K. Kanoda, and T.-K. Ng, Quantum spin liquid states, Rev. Mod. Phys. **89**, 025003 (2017).

[17] L. Savary and L. Balents, Quantum spin liquids: A review, Rep. Prog. Phys. **80**, 016502 (2017).

[18] E. Fradkin, S. A. Kivelson, and J. M. Tranquada, Colloquium: Theory of intertwined orders in high temperature superconductors, Rev. Mod. Phys. **87**, 457 (2015).

[19] T. A. Maier, M. Jarrell, T. C. Schulthess, P. R. C. Kent, and J. B. White, Systematic Study of d-Wave Superconductivity in the 2D Repulsive Hubbard Model, Phys. Rev. Lett. **95**, 237001 (2005).

[20] B.-X. Zheng, C.-M. Chung, P. Corboz, G. Ehlers, M.-P. Qin, R. M. Noack, H. Shi, S. R. White, S. Zhang, and G. K.-L. Chan, Stripe order in the underdoped region of the two-dimensional Hubbard model, Science **358**, 1155 (2017).

[21] H.-C. Jiang and T. P. Devereaux, Superconductivity in the doped Hubbard model and its interplay with next-nearest hopping t', Science **365**, 1424 (2019).

[22] F. Haldane, Continuum dynamics of the 1-D Heisenberg antiferromagnet: Identification with the O(3) nonlinear sigma model, Phys. Lett. A **93**, 464 (1983).

[23] T. Kennedy and H. Tasaki, Hidden symmetry breaking and the Haldane phase in $S = 1$ quantum spin chains, Commun. Math. Phys. **147**, 431 (1992).

[24] P. W. Anderson, Plasmons, gauge invariance, and mass, Phys. Rev. **130**, 439 (1963).

[25] X. G. Wen, Topological orders in rigid stages, Int. J. Mod. Phys. B **04**, 239 (1990).

[26] S. A. Kivelson and S. Lederer, Linking the pseudogap in the cuprates with local symmetry breaking: A commentary, Proc. Natl. Acad. Sci. USA **116**, 14395 (2019).

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

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

[29] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).

[30] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[31] S. J. Wetzel and M. Scherzer, Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory, Phys. Rev. B **96**, 184410 (2017).

[32] A. Morningstar and R. G. Melko, Deep learning the Ising model near criticality, J. Mach. Learn. Res. **18**, 1 (2018).

[33] S. Efthymiou, M. J. S. Beach, and R. G. Melko, Super-resolving the Ising model with convolutional neural networks, Phys. Rev. B **99**, 075113 (2019).

[34] N. Walker, K.-M. Tam, and M. Jarrell, Deep learning on the 2-dimensional Ising model to extract the crossover region with a variational autoencoder, Sci. Rep. **10**, 13047 (2020).

[35] S. Johnston, E. Khatami, and R. Scalettar, A perspective on machine learning and data science for strongly correlated electron problems, Carbon Trends **9**, 100231 (2022).

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2012), pp. 1097–1105.

[37] G. Antipov, S.-A. Berrani, N. Ruchaud, and J.-L. Dugelay, Learned vs. hand-crafted features for pedestrian gender recognition, in *Proceedings of the 23rd ACM International Conference on Multimedia* (ACM, New York, 2015), pp. 1263–1266.

[38] H. Liang, X. Sun, Y. Sun, and Y. Gao, Text feature extraction based on deep learning: A review, J. Wireless Comm. Netw. **2017**, 1186 (2017).

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

[40] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).

[41] K. Ch'ng, N. Vazquez, and E. Khatami, Unsupervised machine learning account of magnetic transitions in the Hubbard model, Phys. Rev. E **97**, 013306 (2018).

[42] C. Alexandrou, A. Athenodorou, C. Chrysostomou, and S. Paul, The critical temperature of the 2D-Ising model through deep learning autoencoders, Eur. Phys. J. B **93**, 1140 (2020).

[43] D. Yevick, Variational autoencoder analysis of Ising model statistical distributions and phase transitions, Eur Phys J. B **95**, 56 (2022).

[44] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).

[45] D. Kingma and M. Welling, Auto-encoding variational Bayes, in *International Conference on Learning Representations* (OpenReview.net., 2014).

[46] Code to reproduce all results in this paper can be found at https://github.com/dagrawa2/ssb_detection_ising. The permanent link is https://doi.org/10.5281/zenodo.6055507.

[47] H.-O. Georgii, *Gibbs Measures and Phase Transitions* (de Gruyter, Berlin, New York, 2011).

[48] L. Onsager, Crystal statistics. I. A two-dimensional model with an order-disorder transition, Phys. Rev. **65**, 117 (1944).

[49] S. L. Smith, P. Kindermans, C. Ying, and Q. V. Le, Don't decay the learning rate, increase the batch size, in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings* (OpenReview.net, 2018).

[50] K. Binder, D. Heermann, L. Roelofs, A. J. Mallinckrodt, and S. McKay, Monte Carlo simulation in statistical physics, Comput. Phys. **7**, 156 (1993).

[51] P. Young, *Everything You Wanted to Know about Data Analysis and Fitting but were Afraid to Ask* (Springer Cham, Heidelberg, 2015).

[52] A. Coniglio, F. de Liberto, G. Monroy, and F. Peruggi, Exact relations between droplets and thermal fluctuations in external field, J. Phys. A: Math. Gen. **22**, L837 (1989).

[53] GAP, GAP–Groups, Algorithms, and Programming, Version 4.11.1, https://www.gap-system.org (2021).

[54] D. Agrawal and J. Ostrowski, A classification of *G*-invariant shallow neural networks, Adv. Neural Info. Proc. Syst. **35**, 13679 (2022).

[55] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning Monte Carlo method, Phys. Rev. B **95**, 041101(R) (2017).

[56] H. Shen, J. Liu, and L. Fu, Self-learning Monte Carlo with deep neural networks, Phys. Rev. B **97**, 205140 (2018).

[57] S. Li, P. M. Dee, E. Khatami, and S. Johnston, Accelerating lattice quantum Monte Carlo simulations using artificial neural networks: Application to the Holstein model, Phys. Rev. B **100**, 020302(R) (2019).

[58] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019).

[59] Y. Nagai, M. Okumura, and A. Tanaka, Self-learning Monte Carlo method with Behler-Parrinello neural networks, Phys. Rev. B **101**, 115111 (2020).

[60] C. Chen, X. Y. Xu, J. Liu, G. Batrouni, R. Scalettar, and Z. Y. Meng, Symmetry-enforced self-learning Monte Carlo method applied to the Holstein model, Phys. Rev. B **98**, 041102(R) (2018).

[61] M. S. Anderson, J. Dahl, and L. Vandenberghe, CVXOPT python software for convex optimization, http://cvxopt.org (2022).