arXiv:2403.09786

Phase determination with and without deep learning

Burak Çivitcioğlu,[1,*] Rudolf A. Römer,[2,†] and Andreas Honecker[1,‡]
[1] *Laboratoire de Physique Théorique et Modélisation, CNRS UMR 8089, CY Cergy Paris Université, 95302 Cergy-Pontoise, France*
[2] *Department of Physics, University of Warwick, Coventry, CV4 7AL, United Kingdom*

(Dated: March 14, 2024; revised October 5, 2024)

Detection of phase transitions is a critical task in statistical physics, traditionally pursued through analytic methods and direct numerical simulations. Recently, machine-learning techniques have emerged as promising tools in this context, with a particular focus on supervised and unsupervised learning methods, along with non-learning approaches. In this work, we study the performance of unsupervised learning in detecting phase transitions in the J_1-J_2 Ising model on the square lattice. The model is chosen due to its simplicity and complexity, thus providing an understanding of the application of machine-learning techniques in both straightforward and challenging scenarios. We propose a simple method based on a direct comparison of configurations. The reconstruction error, defined as the mean-squared distance between two configurations, is used to determine the critical temperatures. The results from the comparison of configurations are contrasted with that of the configurations generated by variational autoencoders. Our findings highlight that for certain systems, a simpler method can yield results comparable to more complex neural networks. This work contributes to the broader understanding of machine-learning applications in statistical physics and introduces an efficient approach to the detection of phase transitions using machine determination techniques.

I. INTRODUCTION

Identification of critical points separating distinct phases of matter is a central pursuit in condensed matter and statistical physics [1, 2]. This task requires a thorough understanding of the global behavior of the many-body system because phenomena may emerge that are very difficult to derive from microscopic rules [3]. Traditional analytic methods and numerical simulations have proven effective in understanding these complex systems [4], but they often come with limitations, particularly in high-dimensional parameter space [5].

Machine-learning (ML) methods, particularly supervised [6] and unsupervised learning techniques [7], have in the last years appeared in physics as a novel strategy to bypassing some of these limitations [8, 9]. They have been shown to yield promising predictions in identifying critical points or phases in parameter space [10–16], providing an alternative and potentially more efficient way of exploring complex systems. In particular supervised machine-learning methods have been shown to be capable of identifying different phases of a physical system [10–16]. Subsequently, several strategies, including but not limited to anomaly detection [17–24], were demonstrated to be able to reconstruct the outlines of a system's phase diagram within unsupervised learning and semi-unsupervised learning contexts. The potential to identify structural changes within a system attracted further attention to these techniques in modern scientific exploration [25, 26].

Among the various models studied in the context of machine learning and statistical physics, the Ising model on the square lattice has served as a benchmark [10, 17, 19–21, 23, 27–40] due to its simplicity and the ready availability of its exact solution [41–43]. Let us also mention related work on multi-layer [44] and Potts models [24, 45–49], where the latter include the Ising model as the $q = 2$ case. Percolation can be considered as the $q \to 1$ limit of the Potts model [50] and yields another class of models to which machine-learning techniques have been applied [51–56].

The J_1-J_2 Ising model incorporates competing interactions across the diagonals of the squares and presents a more challenging case than the aforementioned ones. Investigations of this model have a long history in statistical physics [57–84]. It was observed early on [57, 58] that, with J_1 denoting the nearest-neighbor interaction, the competing second-neighbor interaction J_2 gradually suppresses the ordering temperature, until it vanishes completely when $J_2 = |J_1|/2$. Furthermore, beyond this point, a new ordered phase called the "superantiferromagnetic phase" appears. The universality class of the transition into the superantiferromagnetic phase has been investigated early on [57, 60], but continues to attract attention [61, 63–66, 68–75, 79–81, 84] since its nature remains controversial. There is at least also one investigation of this model on the D-wave quantum annealer [85] and a small number of machine-learning investigations [22, 36].

In this work, we focus on the square-lattice J_1-J_2 Ising model, using machine-learning techniques to predict phase transitions and construct the phase diagram. We adopt the approach of detecting criticality based on the

[*] civitcioglu@aivancity.ai; Present address: aivancity School of AI & Data for Business & Society, 57 Av. du Président Wilson, 94230 Cachan, France
[†] R.Roemer@warwick.ac.uk
[‡] andreas.honecker@cyu.fr

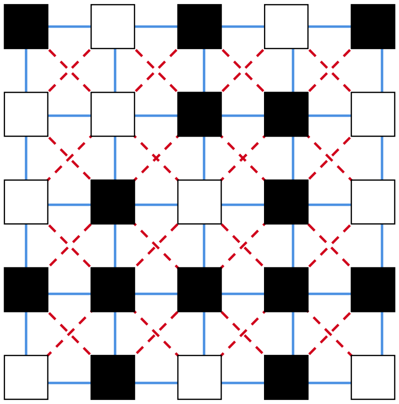

FIG. 1. Schematic representation of the J_1-J_2 Ising model on a 5×5 square lattice. The squares designate the classical spins with black and white corresponding to up and down states, respectively, chosen to illustrate one spin configuration. The solid blue and dashed red lines denote the interactions of nearest neighbors J_1 and next-nearest neighbors J_2, respectively.

reconstruction error ($\mathcal{E}$), defined as the mean-squared distance between two spin configurations, by comparatively using two machine determination methods [86]. The first method is the Variational Autoencoder (VAE), a type of neural network that reconstructs a given predicted state after being trained on a selected set of states [87]. We use the TensorFlow interface to implement our VAE [88]. The second machine-determination method is simpler, based on using a configuration comparison (CMP) and does not require training nor any bespoke machine-learning tools. It is an interpretable method [34], i.e. employs a fully explainable computational strategy. Such methods are generally used both as benchmarks and as practical alternatives to machine learning [89].

Our study aims to provide insights into the capabilities of these computational methods in the context of phase transition detection and we conclude that the simpler method (CMP) can achieve success rates that can be compared to that of the VAEs.

II. THE J_1-J_2 ISING MODEL

The J_1-J_2 Ising model serves as a simple but non-trivial system to illustrate phase transitions, especially those associated with magnetic behavior. As presented in Fig. 1, it adds the complexity of second-nearest-neighbor interactions to the traditional nearest-neighbor Ising model. The Hamiltonian of the J_1-J_2 Ising model is expressed as

$$H_{J_1 J_2} = -J_1 \sum_{\langle i,j \rangle} s_i\, s_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} s_i\, s_j\,. \tag{1}$$

Here, s_i represents the spin at site i, which can be either up ($+1$) or down (-1); $\langle i,j \rangle$ refers to nearest-neighbor pairs, $\langle\langle i,j \rangle\rangle$ denotes next-nearest neighbor pairs, while J_1, $J_2 \geq 0$ signify the interaction strengths between the nearest and next-nearest neighbors, respectively. The sign convention of the Hamiltonian in Eq. (1) leads to a ferromagnetic coupling for J_1 pairs while next-nearest neighbors prefer to align in an antiferromagnetic structure. Some previous investigations have used $J_1 < 0$, but this yields equivalent physics to the case $J_1 > 0$ considered here, see appendix A for further details. We present results in units of $|J_1| = 1$ where the absolute value emphasizes that the structure of the phase diagram is the same for both signs of J_1. We investigate square lattices of size $L \times L$ (linear extent L) and impose periodic boundary conditions.

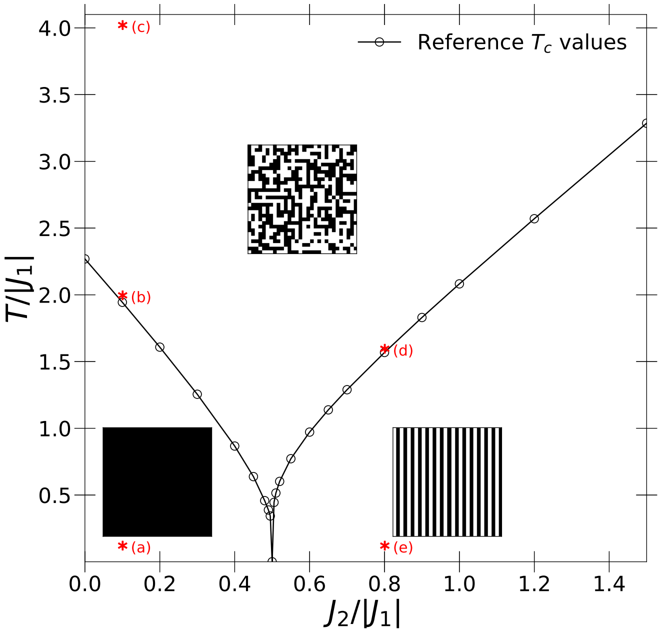

FIG. 2. Phase diagram of the J_1-J_2 Ising model on the periodic square lattice, with sample configurations from each of the three phases shown. The top paramagnetic configuration was obtained at $J_1 = 1$, $J_2 = 0.1$, and $T = 4 > T_c$. The bottom left ferromagnetic configuration corresponds to $J_1 = 1$, $J_2 = 0.1$, and $T = 0.1 < T_c$. The bottom right configuration was found for $J_1 = 1$, $J_2 = 0.8$, and $T = 0.1 < T_c$. This configuration illustrates the superantiferromagnetic phase. The reference T_c data is based on Ref. [66] and shown by circles connected by lines as guide for the eye. The five red stars ($*$) and their associated letters locate the (T, J_2) positions for selected configurations further detailed in Fig. 3.

A. Phase diagram

Figure 2 recalls the well-studied phase diagram of the J_1-J_2 Ising model. Here, we use the previously computed high-precision transition temperatures T_c from Ref. [66] for reference. Some more recent numerical investigations such as Refs. [68, 69, 71, 75, 80, 84] may provide more ac-

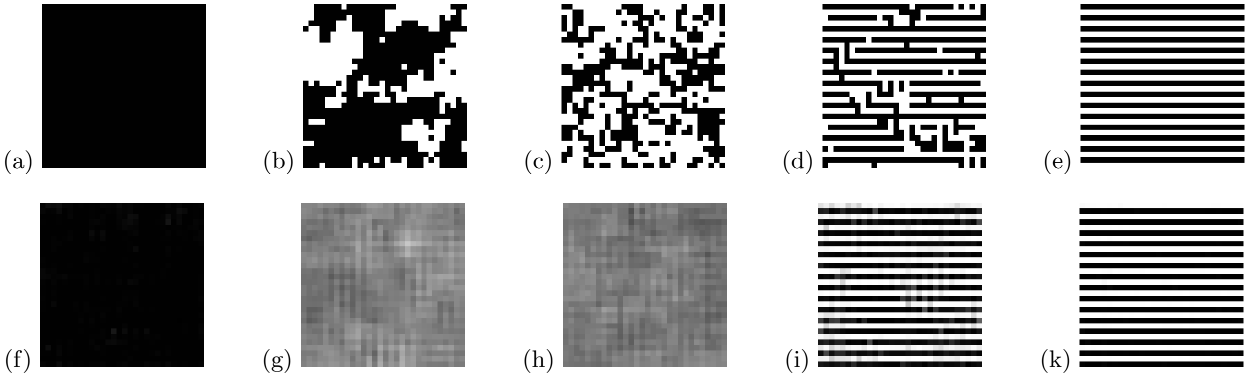

FIG. 3. (a-e) Five illustrative spin configurations of the J_1-J_2 Ising model on a periodic 30×30 square lattice for (T, J_2) values as indicated in Fig. 2. (f-k) Predicted spin configurations from (a-e), respectively, using the VAEs from the in-phase training cycles described in Sec. IV B. Panels (a, b, c) keep $J_2 = 0.1$ fixed while increasing T from (a) ferromagnetic at $T = 0.1$ to (b) $T = 1.975$ near the ferromagnetic-to-paramagnetic transition and (c) a configuration deep in the paramagnetic phase at $T = 4.0$. Panels (d, e) have $J_2 = 0.8$ and then decrease T from (d) $T = 1.575$ near the paramagnetic-to-superantiferromagnetic transition to (e) a superantiferromagnetic configuration at $T = 0.1$. In (f-k), the parameters are as in (a-e), respectively. In all cases, the black squares correspond to up spins while white is for down spins as in Fig. 1. In (f-k), the values in the interval $[-1, 1]$ are denoted by the gray squares in the panels.

curate estimates of the transition temperatures, but any potential differences are so small that they are irrelevant for the present purposes.

The J_1-J_2 Ising model exhibits three distinct phases that we illustrate, by one representative spin configuration each, in Fig. 2. The *paramagnetic* phase appears at sufficiently high temperatures, namely $T > T_c$ irrespective of the values of the interaction constants. This phase is characterized by the absence of long-range order. The next phase is the *ferromagnetic* one, characterized by a preference of the neighboring spins to align. In the $T = 0$ ground state, spins are perfectly aligned, yielding an energy per site of $e_{\text{ferro}} = -2\,J_1 + 2\,J_2$. The finite-temperature phase transition to this ferromagnetic phase starts at the exactly known value $T_{c,\text{Ising}}/\left|J_1\right| = 2/\ln(1+\sqrt{2}) \approx 2.269$ for $J_2 = 0$ [41] and is gradually suppressed by a competing $J_2 > 0$. The third and last phase is known as the *superantiferromagnetic* phase [58]. Here, the J_1 and J_2 interactions compete: J_1 prefers to align nearest-neighbor spins parallel while J_2 tries to enforce an antiparallel alignment of next-nearest neighbor spins. In the superantiferromagnetic state, either vertical or horizontal stripes composed of opposing spins are formed, thus satisfying all J_2 interactions and half of the J_1 interactions. At $T = 0$, this order is perfect, yielding an energy per site of $e_{\text{super}} = -2\,J_2$.

When $J_2 = \left|J_1\right|/2$ for $T = 0$, we find $e_{\text{ferro}} = e_{\text{super}}$, i.e., the ground-state energies become degenerate, corresponding to the transition point between ferromagnetic and superantiferromagnetic phases. Numerical investigations [66, 75, 80, 83] indicate that there is no finite-temperature phase transition exactly at $J_2 = \left|J_1\right|/2$ and that the critical temperature T_c is suppressed to $T_c = 0$ when approaching $J_2 = \left|J_1\right|/2$ from either ordered phase.

Figure 3 shows further examples of spin configurations, emphasizing changes upon approaching the phase transition. Panels (a) and (e) show configurations at low temperatures in the ferromagnetic and superantiferromagnetic phase, respectively. These are similar to the configurations already shown in Fig. 2, except that in the superantiferromagnetic case the stripes in the examples are rotated by 90°. Panels (b) and (d) of Fig. 3 show configurations at higher temperatures, closer to the critical temperature T_c. Here one observes fluctuations on top of the ordered background. Finally, Fig. 3(c) shows another example for a configuration in the high-temperature paramagnetic phase.

B. Monte Carlo Method

To generate the necessary input data for the machine determination models, we utilize the Metropolis algorithm, a well-established method in the realm of computational physics for simulating thermal systems [90–93].

In the present investigation, we initially focus on a system size of 30×30 with periodic boundary conditions. This choice is motivated by the machine-learning frameworks being tailored for images of similar size. Indeed, previous related work [22, 86] for the J_1-J_2 Ising model on the square and honeycomb lattices also employed $L = 30$. In order to assess the influence of the size of the system, we also investigate 60×60 and 120×120 square lattices, again with periodic boundary conditions.

Equilibration of the simulations can be difficult, in particular in the regime of $J_2 \approx \left|J_1\right|/2$ [66]. In order to ensure proper thermalization within a simple single-spin flip Metropolis scheme, we proceed as follows: we fix $J_2/|J_1|$ and start from a high initial temperature $T/|J_1| = 100$, where the spin configurations are essentially random. Next, we gradually lower the temperature to $T/|J_1| = 4$ over the course of 1000 Monte Carlo sweeps (MC sweeps, i.e., complete $L \times L$ spin updates). Then we start the data collection phase: for each T, we first thermalize for another 3000 MC sweeps. Next, we collect five spin configurations at a given T, spaced by 1000 MC sweeps between each measurement to ensure the statistical independence of the configurations. Finally, temperature is lowered by $\Delta T/|J_1| = 0.025$ and the procedure repeated, until we reach $T/|J_1| = 0.1$. This yields a set $\mathcal{T}$ of $|\mathcal{T}| = 157$ temperatures with $T/|J_1| \in [0.1, 4]$ for $T \in \mathcal{T}$. The procedure is repeated with different random numbers until we have $C = 40$ configurations for each temperature at the given value of $J_2/|J_1|$.

We then select another $J_2/|J_1| \in \mathcal{J}_2$ and collect spin configurations as above. Here, $\mathcal{J}_2$ denotes the set $\mathcal{J}_2 = \{0, 0.1, 0.2, 0.3, 0.4, 0.45, 0.48, 0.49, 0.495, 0.5, 0.505, 0.51, 0.52, 0.55, 0.6, 0.65, 0.7, 0.8, 0.9, 1, 1.2, 1.5\}$ with $|\mathcal{J}_2| = 22$ distinct values. In total, this results in a dataset containing $|\mathcal{T}|\times|\mathcal{J}_2|\times C = 157\times 22\times 40 = 138\,160$ independent configurations for a given system size. We shall denote this *global* dataset as ρ_G. The examples shown in Figs. 2 and 3 were taken from ρ_G. We note that the configurations are stored in an exact numeric form, and not as potentially lossy images, as the machine-learning context might suggest [94].

III. MACHINE-LEARNING APPROACHES

The datasets described in the preceding section will form the basis for two independent machine-determination approaches to the phase diagram of the J_1-J_2 Ising model. In principle, we would like to explore fully unsupervised approaches, but we note that some prior knowledge about the phase diagram goes into the generation of the underlying dataset, namely the relevant range of temperatures T and the required resolution of coupling ratios $J_2/|J_1|$.

We will employ two distinct computational methods: the first approach follows Refs. [22, 86] and uses a deep-learning model, viz. a variational autoencoder (VAE) that produces a predicted output spin configuration for each given input configuration. The second approach is much simpler, namely just a direct comparison of configurations, but has to the best of our knowledge not been implemented previously. The generated datasets will be used, as shown in Fig. 4 (to be discussed in more detail later in Sec. IV A), as the training data for the VAE and the reference configurations for the direct comparison of configurations. Consequently, the reconstruction errors are computed, from which the phases can be identified. Then as the final step in the workflow we estimate the T_c's for each J_2 value.

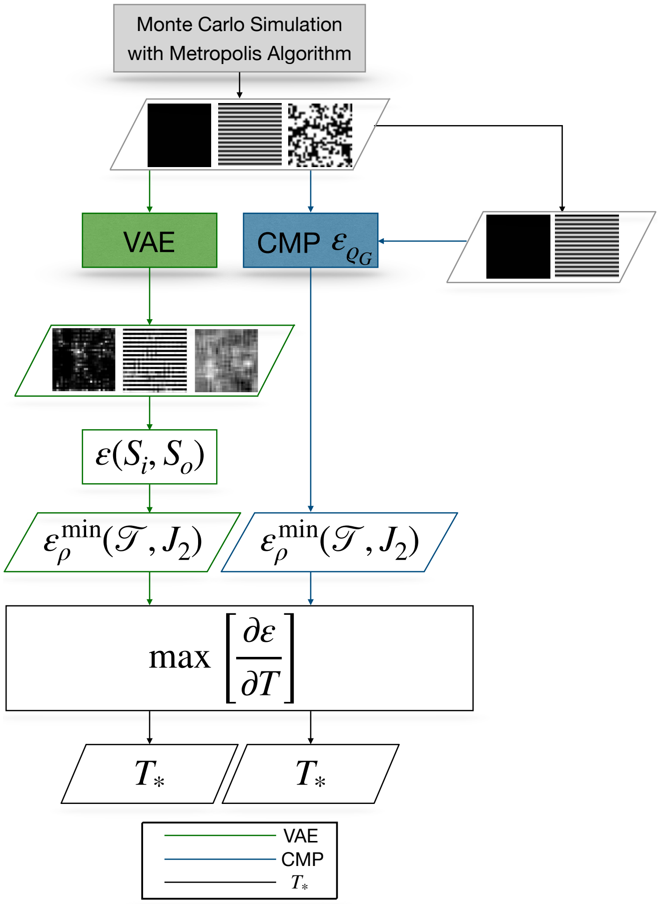
Monte Carlo Simulation
with Metropolis Algorithm
VAE
CMP $\mathcal{E}_{\varrho_G}$
$\varepsilon(S_i, S_o)$
$\varepsilon_\rho^{\min}(\mathcal{T}, J_2)$
$\varepsilon_\rho^{\min}(\mathcal{T}, J_2)$
$\max\left[\frac{\partial\varepsilon}{\partial T}\right]$
T_*
T_*
VAE
CMP
T_*

FIG. 4. Workflows of the two machine-"learning" approaches implemented in the present work: We begin with an input dataset that is processed in the first approach by a VAE to generate a reconstructed set of configurations. In the second approach, we compare the input images against some reference configurations (CMP). For both pathways, we calculate the "reconstruction error" $\mathcal{E}$ for each configuration, resulting in a set of reconstruction error distributions. By analyzing the derivative of the distribution, we finally pinpoint the temperature T_* with the largest derivative as the predicted critical temperature T_c for a given value of $J_2/|J_1|$.

A. Variational Autoencoder

A VAE is a relatively recent deep-learning architecture that combines standard compression techniques with the regularization strategies of machine learning, serving also as a generative model [87, 95]. In brief, it consists of an *encoding* multilayered neural network that, upon training with input data, produces output parameters for a variational distribution. These parameters characterize a low-dimensional probabilistic distribution space, known as *latent space*. The *decoding* part of the VAE then is again a deep neural network architecture that generates the reconstructed output data from the latent space, taking samples from the latent space rather than determin-

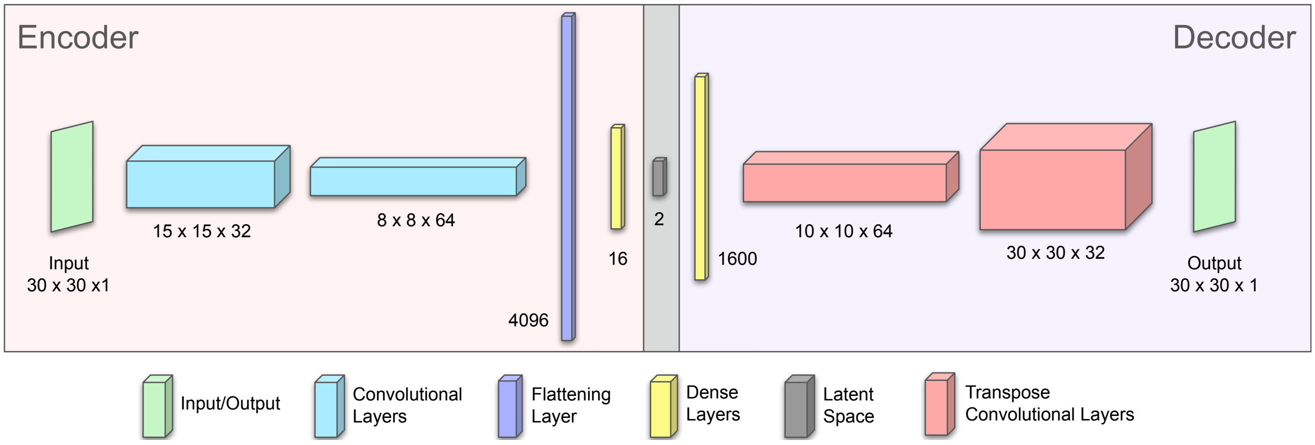

FIG. 5. Network architecture of the variational autoencoder for the 30×30 lattice. The encoder consists of two convolutional layers with 32 and 64 filters, ReLu activation function, and padding, followed by a flattening layer and a dense layer with 16 units. The dense layer outputs the mean and log-variance parameters to the latent space, from which the latent vector $\mathbf{z}$ is sampled. The decoder, starting with a dense layer and a reshape layer, uses transposed convolutional layers to reconstruct the input image from $\mathbf{z}$. The final layer applies a sigmoid activation function to produce output values between 0 and 1, which are then rescaled to the interval $[-1, 1]$.

istic points. Clearly, when the dimension d of the latent space is much smaller than the information content of the input data, this procedure will lead to some information loss. Hence, one is trying to construct en- and decoders such that upon encoding a maximum of information is kept while upon decoding a minimum of error is introduced into the output data. In order to optimally train the model to generate such a VAE, two loss measures are employed. The *reconstruction error* ε (see Sec. III B for details) quantifies the difference between input and output data upon training. In addition, the so-called Kullback-Leibler divergence [96] assures the regularization of the latent space, making it approximate standard normal distributions [87]. In practice, during training one minimizes a total loss ℓ that combines the reconstruction loss ℓ_ε in the final layer from ε as well a Kullback-Leibler loss ℓ_{KL} [87], such that $\ell = \ell_\varepsilon + c\,\ell_{\mathrm{KL}}$, where c a constant. We have tried different values of $c \in \{1 \ldots 10\}$ and observed that it does not yield any notable changes in the results, therefore we fix $c = 1$.

In the present case, our input data consists of spin configurations with dimensions $(L, L, 1)$ with a simple ± 1 binary value for each of the L^2 spins. This is similar to standard black-and-white images, in particular after normal batch-normalization operations in the encoding [97]. Hence, we are using in the following the same regularization strategy as used in VAEs for image reconstructions (ImageNet [98]). We have also checked that a latent space dimension up to $d = 8$ reproduces similar results, but we find that $d = 2$, as shown in Fig. 5, is sufficient and often better in distinguishing phases. We suggest that this is due to the relatively small number of phases for the J_1-J_2 model. We note that $d = 2$ has also been used with good accuracy in the MNIST ten-numerals-recognition challenge [99, 100].

Figure 5 shows the architecture of the VAE, inspired by Refs. [22, 86], for a 30×30 lattice. For the other input sizes $L = 60$ and 100, our network architecture has the same structure, but with size-adjusted parameters. In order to keep the parameters padding, kernel size and stride for $L = 120$ the same as for $L = 30$ and 60, we add a third convolutional layer. The network architecture for $L = 30$ begins with two convolutional layers with 32 and 64 filters, respectively, each with a 3×3 kernel size and a stride of 2. Both layers apply a rectifier linear unit or ReLu activation function that is defined as $\mathrm{ReLU}(x) = \max(0, x)$ and use padding to preserve spatial dimensions. After these convolutional layers, a flattening layer reshapes the 3D output into a 1D tensor, which then feeds into a dense layer with 16 units and again ReLU activation. This layer prepares the data for conversion into the latent space. The final two layers of the encoder output two parameters: the mean, μ, and the log-variance, $\log \sigma^2$, of the latent space. These parameters define a Gaussian distribution, from which we sample using the parameterization trick [87]. The decoder then aims to reconstruct the input spin configuration (image) from the compressed information in the latent space. The architecture of the decoder starts with a dense layer that up-samples to dimensions of $5 \times 5 \times 64$ $(= 1600)$. A reshape layer follows, converting the 1D tensor back into a 3D tensor. After this, two sets of transposed convolutional layers with 64 and 32 filters, respectively, are applied, each with a 3×3 kernel size and ReLU activation. The first has a stride of 2 and the second has a stride of 3, and both utilize padding. The final layer of the decoder uses another transposed convolutional layer with one filter to generate the output image, applying the sigmoid activation to ensure that the output values fall within the range between 0 and 1. More details on the structure and performance of the latent space can be found in Ref. [101].

B. Reconstruction error

The typical reconstruction error used in many image-based VAE applications is known as "mean-squared error" (MSE) and is defined as

$$\varepsilon(\mathbf{S}_\mathrm{i}, \mathbf{S}_\mathrm{o}) = \frac{1}{4L^2} \sum_{l=1}^{L^2} (s_{\mathrm{i},l} - s_{\mathrm{o},l})^2 \,, \tag{2}$$

where $\mathbf{S}_\mathrm{i} = \{s_{\mathrm{i},l}\}$ and $\mathbf{S}_\mathrm{o} = \{s_{\mathrm{o},l}\}$ correspond to the input and output configurations of the VAE, respectively. For two identical spin configurations, $\mathbf{S}_\mathrm{o} = \mathbf{S}_\mathrm{i}$, we obviously have $\varepsilon = 0$. For two opposite configurations, $\mathbf{S}_\mathrm{o} = -\mathbf{S}_\mathrm{i}$, we find $\varepsilon = 1$ while for two configurations with half the spins identical and half opposite, we have $\varepsilon = 0.5$. The latter value is also true when comparing two independent and identically distributed, i.e., random, spin configurations, at least when $L \to \infty$. We note that in (2), the factor 4 in the denominator assures that the ε are normalized similar to the standard results for MSEs where usually comparisons are for values ranging from 0 to 1, whereas in the present case the range is $[-1, 1]$. While the spin configurations computed for the J_1-J_2 model are restricted to values ±1, no such restriction is in place for the output generated by the VAE and the possible range of $s_{\mathrm{o},l}$ is $[-1, 1]$. Therefore, in principle all values $\in [0, 1]$ are possible when computing ε between a spin configuration $\mathbf{S}_\mathrm{i}$ computed from the J_1-J_2 model, and $\mathbf{S}_\mathrm{o}$, reconstructed via the VAE. In particular, if the VAE would produce a completely featureless configuration $s_{\mathrm{o},l} = 0$ for all l in the $L \times L$ lattice, then we would have $\varepsilon = 0.25$.

Often, we shall be interested in reconstruction errors originating from differently trained VAEs. For example, we might be interested in selecting a certain smaller region ρ of the $(\mathcal{T}, \mathcal{J}_2)$ parameter space as the space from where the input spin configurations $\mathbf{S}_\mathrm{i}$, used in the training of a VAE, originate. We shall then use ε_ρ to denote that particular training. Furthermore, when testing a particular VAE, we will do so at specific T and J_2 values. We note that while we use the term *testing* in the technical ML sense, its physics use is in producing reconstruction errors $\varepsilon_\rho(T, J_2)$ to allow phase reconstruction. In principle, there is one such $\varepsilon_\rho^{(c)}(T, J_2)$ value for each spin configuration c of the $C = 40$ configurations constructed at each point (T, J_2) as discussed in section II B. Hence it is at this point that a statistical analysis can be applied, e.g., construct mean and minimal estimates, i.e., $\langle\varepsilon_\rho\rangle(T, J_2) = \sum_{c=1}^{C} \varepsilon_\rho^{(c)}(T, J_2)/C$ and $\min(\varepsilon_\rho)(T, J_2) = \min\{\varepsilon_\rho^{(c)}(T, J_2) \mid c \in C\}$. We note that in computing these statistical estimates – and their standard errors used later – we do not also change the underlying input configurations as obtained from Monte-Carlo.

Sets of reconstruction errors shall be denoted $\mathcal{E}$, with

$$\mathcal{E}_\rho(J_2) = \big\{\{T, \varepsilon_\rho(T, J_2)\} \mid T \in \mathcal{T}\big\} \tag{3}$$

denoting the set of all reconstruction errors at constant J_2 computed for a VAE trained on region ρ. Here, when suppressing the explicit mention of the configuration label c, we then mean that all C configurations are elements of such a set. Further, we shall denote by $\mathcal{E}_\rho(\mathcal{T}, \mathcal{J}_2)$ the set of all $\mathcal{E}_\rho(J_2)$ with $J_2 \in \mathcal{J}_2$. Finally, we define

$$\mathcal{E}_\rho^{\min}(\mathcal{T}, J_2) \;=\; \big\{\min_{c\in C}\left[\varepsilon_\rho(T, J_2)\right] \mid T \in \mathcal{T}\big\}, \tag{4}$$

$$\mathcal{E}_\rho^{\min}(\mathcal{T}, \mathcal{J}_2) \;=\; \big\{\min_{c\in C}\left[\varepsilon_\rho(T, J_2)\right] \mid T \in \mathcal{T}, J_2 \in \mathcal{J}_2\big\}. \tag{5}$$

Average $\langle\cdot\rangle$ or maximal $\max[\cdot]$ values can be defined in the same way as above for the minimum.

C. Comparing individual spin configurations

When training the VAE, the information about the spin configurations in the region ρ is learned, i.e., nonlinearly encoded, in the set of parameters of the en-/decoding neural networks and latent space of the VAE. Then, when given an arbitrary input spin configuration at (T, J_2), the VAE will generate a new spin configuration according to the information imprinted on its parameters based on ρ, aiming to minimize $\varepsilon_\rho(T, J_2)$. Alternatively, one can also just use each spin configuration of ρ and then simply compute $\varepsilon_\rho(T, J)$ between a test configuration at (T, J_2) and each reference spin configuration. This leads to the set $\mathcal{E}_\rho(T, J_2)$ of "reconstruction errors" [102]. As in the case of the VAE, we can proceed to again define the reconstruction error sets analogously to (3), (4), and (5).

Without further optimization, such a direct comparison of individual spin configurations will scale with the number $|\rho|$ of configurations in ρ and each computation of the reconstruction error in (2) uses $O(L^2)$ operations.

IV. RECONSTRUCTION OF THE PHASE DIAGRAM USING A SINGLE VAE

We can now begin to use the VAE architecture to identify the phases of the J_1-J_2 model as a function of T and J_2 for constant $J_1 = 1$. For $J_2 = 0$, we are back to the nearest-neighbor Ising model with known critical temperature $T_{c,\mathrm{Ising}} \approx 2.269$ [41]. We can therefore confidently choose at least an initial temperature range of $0 \leq T \leq 4$ containing $T_{c,\mathrm{Ising}}$. From Sec. II, we also know that the ferromagnetic-to-superantiferromagnetic transition is at $J_2 = 1/2$. Hence we choose a range for J_2 from 0 to 1.5 (should we later see that these ranges do not suffice to capture all phases, we could further increase the maximal T and J_2 values). In the present case, we have chosen

training cycle	loss	value
global	ℓ_{KL}	13.1(3)
global	ℓ_ε	10072(15)
in-phase (F)	ℓ_{KL}	9(2)
in-phase (F)	ℓ_ε	8(4)
in-phase (S)	ℓ_{KL}	17(7)
in-phase (S)	ℓ_ε	20(20)
global	λ	**0.2075**(3)
in-phase (F)	λ	**0.0012**(2)
in-phase (S)	λ	**0.001**(1)

TABLE I. Losses obtained at epoch 500 for global and in-phase training cycles as discussed in Secs. IV A and IV B, respectively. The ℓ_ε and ℓ_{KL} are given separately to show their relative importance. The error is calculated as the standard error, given by $\sigma/\sqrt{n}$, where σ is the standard deviation of the mean, and $n = 10$ is the total number of trainings. The symbols (F) and (S) indicate the ferromagnetic and superantiferromagnetic in-phase training cycles, respectively. The last three (bold) λ values have been used in Fig. 6.

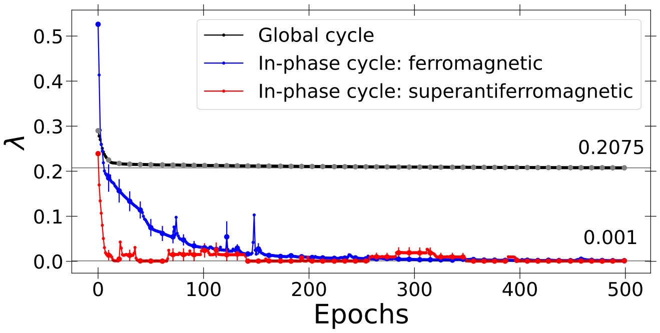

FIG. 6. Mean loss per site λ for 500 epochs when averaged over 10 independent trainings. The black markers represent the results for the global training cycle while blue and red represent in-phase training cycles for ferromagnetic training data and superantiferromagnetic training data, respectively. The gray horizontal lines indicate the final mean values of λ at epoch 500. Error bars are the usual standard error of the mean and shown, for clarity, at every 10th symbol only. The λ for the global training cycle is visibly higher than for the in-phase training, due to the selection of $\rho_{G,\mathrm{train}}$ with training samples chosen as specified in Sec. IV A.

these T and J_2 values to coincide with the range of available data as described in Sec. II where we denoted these ranges as $\mathcal{T}$ and $\mathcal{J}_2$.

A. Global training cycle

In order to train the *single* VAE, we now choose from each of the $|\mathcal{T}| \times |\mathcal{J}_2| = 3454$ pairs (T, J_2) one of the C configurations randomly. In this way, configurations from the whole range of T and J_2 values are included in the training cycle. We shall call this dataset $\rho_{G,\mathrm{train}} \subset \rho_G$. We train for 500 epochs with a batch size of $|B| = 64$, and achieve a Kullback-Leibler loss of $\ell_{\mathrm{KL}} = 13.1 \pm 0.3$ and an MSE loss of $\ell_\varepsilon = 10072 \pm 15$, see also the first two lines of Table I. It may be useful to convert this to a per-site MSE λ. To this end, we first consider the batch size and the number of spins per configuration $30 \times 30 = 900$ which results in $3454/64 \approx 54$ configurations per epoch. The total number of spins in training is thus $54 \times 900 = 48600$. The per-site MSE total loss divided by the total number of spins during the training, is $\lambda = (\ell_\varepsilon + \ell_{\mathrm{KL}})/\left(L^2 \frac{|\mathcal{T}|\times|\mathcal{J}_2|}{|B|}\right)$. We find that ℓ_ε corresponds to $\lambda = 0.2075 \pm 0.0003$ for $L = 30$. As λ is a more intuitive metric, we use λ to show the evolution of the loss during training epochs in Fig. 6. One observes good convergence of the training for a total of 500 epochs.

The workflow after the training is shown in Fig. 4. We now test how the trained VAE can reconstruct configurations using the test dataset $\rho_{G,\mathrm{test}}$. Here, $\rho_{G,\mathrm{test}} = \rho_G \setminus \rho_{G,\mathrm{train}}$, i.e., the full dataset with $\rho_{G,\mathrm{train}}$ removed. The size of the test dataset is $|\rho_{G,\mathrm{test}}| = |\rho_G| - |\rho_{G,\mathrm{train}}| = 138160 - 3454 = 134706$. For each pair (T, J_2) we then have $C - 1 = 39$ generated spin configurations and can compute $\varepsilon(T, J_2)$ for each. In order to use the VAE to reconstruct the phase diagram, we now choose a particular $J_2' \in \mathcal{J}_2$. We then compute $\varepsilon(T, J_2')$ for all $T \in \mathcal{T}$. At each (T, J_2') there will be a distribution of 39 ε values. When T and J_2 are far away from phase boundaries, the 39 $\varepsilon(T, J_2')$ values will follow a roughly similar behavior in each phase. On the other hand, close to phase boundaries, there will be a large variation in $\varepsilon(T, J_2')$. One could hence in principle compute $\langle\varepsilon(T, J_2')\rangle$ to detect a phase change. We have found that $\min_{c\in C} \varepsilon(T, J_2')$ works even better for the J_1-J_2 model.

Figure 7 shows results for $J_2 = 0$ and 0.4. As expected, for both J_2, we see that $\varepsilon \approx 0$ when $T \ll T_c(J_2)$ while for $T \gg T_c(J_2)$ we find $\varepsilon \approx 0.25$. The temperature range where ε changes is already reasonably close to $T_c(J_2)$ for $L = 30$ and we note that, upon increasing L, the curves become sharper with the kink approaching the exact value in the thermodynamic limit.

Figure 8(a) shows the emerging phase diagram. We can see that $\mathcal{E}^{\min}_{\rho_G}(\mathcal{T}, \mathcal{J}_2)$ separates into two distinct regions. Let us emphasize that up to this point, we have not used any a priori information besides the two limiting transition temperatures as outlined above. In particular, we have not yet used information about the spatial configurations of the spins in each of the phases. It is hence noteworthy to find that the border between the two identified regions already is very close to the known phase boundaries. The apparent steps in the VAE estimates for the phase boundaries arise in fact from the underlying discrete set of values of T and in particular J_2 investigated whereas the agreement for actual values, as denoted by the circles, is in fact at the resolution limit of the figure.

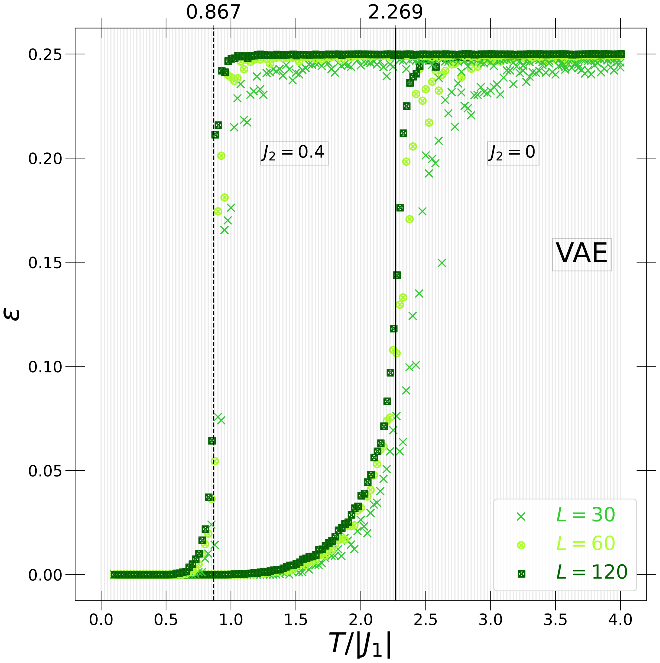

FIG. 7. Reconstruction error $\mathcal{E}^{\min}_{\rho_G}(T, \mathcal{J}_2)$ obtained by a single VAE for $J_2 = 0$ and 0.4 shown for $L = 30$ (×, green), 60 (⨂, light green) and 120 (⊠, dark green). The 157 gray vertical lines represent the temperatures in $\mathcal{T}$. The black solid vertical line indicates the exact $T_c(J_2 = 0) = 2.269$ and the black dashed line shows the numerical estimate of $T_c(J_2 = 0.4) = 0.867$ from Ref. [66].

B. In-phase training cycle

Having identified two distinct regions in Sec. IV A, we can now repeat the training of the VAE deep in one of the two phases. In Figs. 8(b,c) we indicate two distinct such regions for low T, following the general shape of the boundary between the two regions which seems to have a clear separation into low- and high-J_2 sub-regions. We follow the same procedure as for Fig. 8(a), but with the much more restricted training data regions $\rho_{\text{low-}J_2}$ and $\rho_{\text{high-}J_2}$. In order to have a reasonable amount of training data, we now use all 40 values for each (T, J_2) in each training region. For the results underlying Fig. 8(b), this amounts to 1440 training configurations in $\rho_{\text{low-}J_2}$, while for Fig. 8(c), we have 1800 configurations in $\rho_{\text{high-}J_2}$. Loss function convergence values are presented in Table I. We note that these low-temperature configurations turn out to be well ordered, as illustrated by the examples in Figs. 3(a,e). One could thus also use synthetic configurations in this case. We nevertheless prefer to use Monte Carlo data deep inside the ordered phases, as illustrated in Figs. 8(b,c), in order to minimize prior knowledge going into the reconstruction of the phase diagram.

From Fig. 8(b) we see that the trained network identifies again two distinct regions. Now, the low-T, low-J_2 region is clearly separated from the rest of the (T, J_2) plane. Similarly, Fig. 8(c) establishes a low-T, high-J_2 region. We note that in both cases, the ε values in the low/high-J_2 regions are close to zero, while in the other regions we have $\varepsilon \approx 0.5$. This value suggests that in both cases, the out-of-region configurations have about 50% of spins different, in agreement with the known phases as presented in Fig. 2. We can therefore conclude that the low-T region identified in Fig. 8(a) consists of two distinct regions.

C. Discussion of reconstruction error

Overall, the combination of global and in-phase learning has indeed led to the identification of three separate regions. These regions agree very well with the previously established phases shown in Fig. 2. The ε values of 0, 0.25, and 0.5 indicate best, random, and worst reconstruction possible, respectively, compatible with the spin configurations in each phase. Clearly, the regions with $\varepsilon \approx 0$ correspond to the ordered ferro- and super-antiferromagnetic phases. When using the VAEs trained in both of these phases, we find $\varepsilon \approx 0.5$ when testing in the disordered paramagnetic phase, clearly showing the difference between ordered and disordered phases.

The value of $\varepsilon \approx 0.25$, however, should not emerge when simply comparing the configuration of up (+1) and down (−1) spin states as shown in Figs. 3(a–e). However, it is compatible with the VAE reconstruction of spin configurations similar to the undifferentiated “gray” in Figs. 3(g+h). This tells us that even though we can train the VAEs in the disordered phase, this does not lead to any predictive power in the disordered phase. Put differently: if we had chosen a third region to train our VAEs, namely a high-T region in the disordered phase, we would not have been able to identify the phase boundaries to the two ordered phases.

V. RECONSTRUCTION OF THE PHASE DIAGRAM USING MULTIPLE VAES

Another approach is training *multiple* VAEs for different regions ρ spanning across the (T, J_2) plane. Let $\rho_{\mathcal{T}'}(J_2)$ denote a region at constant J_2 with varying temperature in $\mathcal{T}' = \{0.1, \ldots, 4\}$. $\mathcal{T}'$ consists of 40 evenly spaced temperatures with $\Delta T = 0.1$ and $\mathcal{T}' \subset \mathcal{T}$. In Fig. 8(d), we indicate as example $\rho_{\mathcal{T}'}(J_2 = 1.2)$. Additionally, we define by $\rho_{\mathcal{J}_2}(T)$ regions of constant T but containing the 22 elements of $\mathcal{J}_2$. An example of such a region is again given in Fig. 8(d), namely for $T = 0.25$.

Then for a given (T, J_2) pair, we compute $\varepsilon_\rho(T, J_2)$. In order to remove the bias of low ε for in-region trainings, we average over the reconstruction errors computed from all ρ, i.e.,

$$\langle \varepsilon \rangle(T, J_2) = \frac{\sum_{\rho \in \rho_{\mathcal{T}'}(J_2), \rho_{\mathcal{J}_2}(T)} \varepsilon_\rho(T, J_2)}{|\rho_{\mathcal{T}'}(J_2)| + |\rho_{\mathcal{J}_2}(T)|}. \tag{6}$$

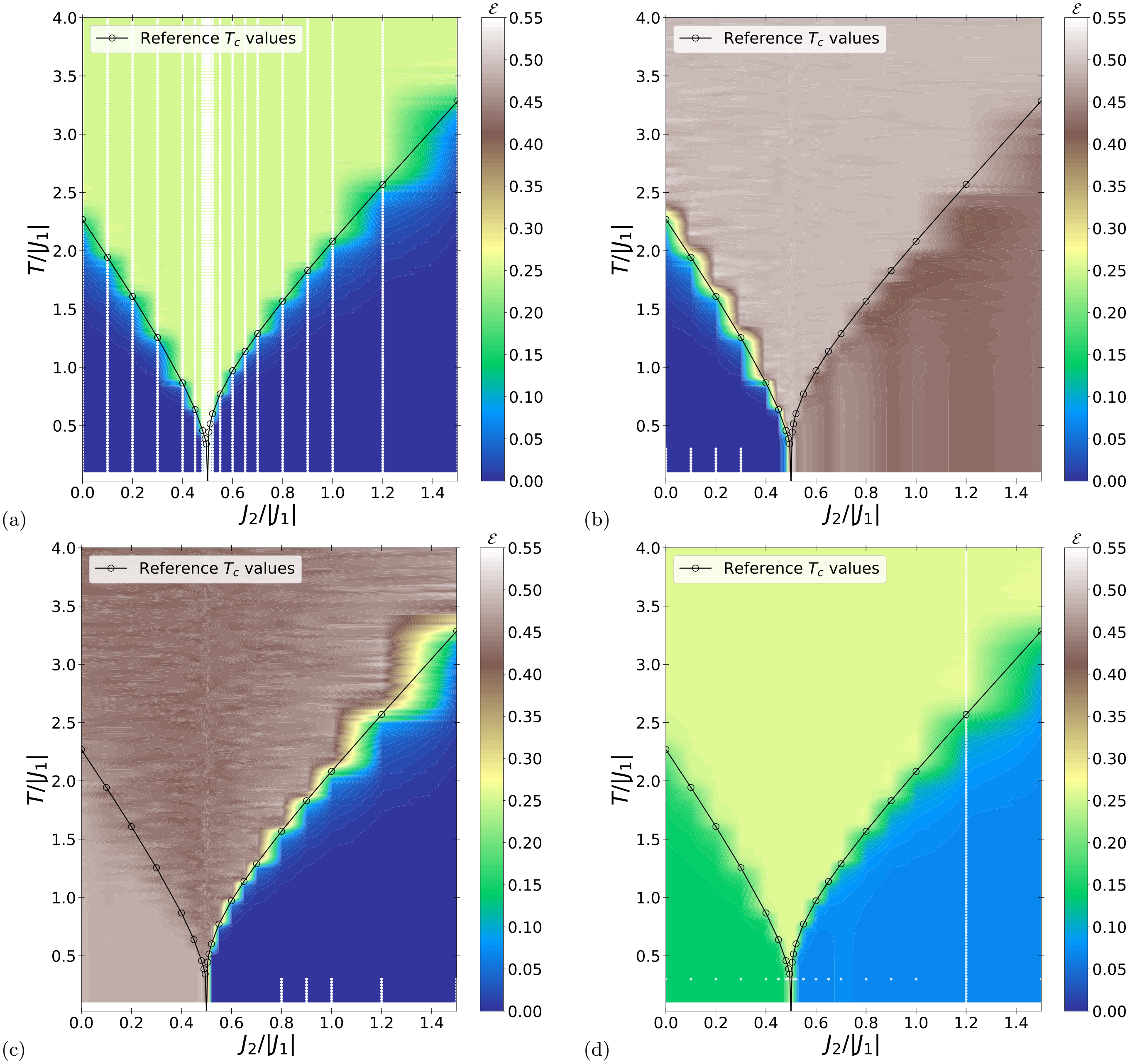

FIG. 8. $\mathcal{E}_\rho^{\min}(\mathcal{T}, \mathcal{J}_2)$ for the VAE-based reconstruction of the J_1-J_2 model's phase diagram from Secs. IV and V. The results correspond to $L = 30$. For the single VAE approach of Sec. IV, panel (a) shows the global training cycle of Sec. IV A with $\rho = \rho_G$, (b) represents the in-phase learning of Sec. IV B from the low-J_2 region $\rho_{\text{low-}J_2}$ and (c) gives results for the in-phase learning from the high-J_2 region $\rho_{\text{high-}J_2}$. The (T, J_2) data points of various training regions are indicated by small white dots for each (T, J_2) pair (usually these are closely spaced and hence appear as vertical lines). Lastly, panel (d) shows the results of the multiple VAE approach of Sec. V. Here, the white dots give examples for the constant J_2 (the vertical line of dots) and the constant T trainings (horizontal line of dots) explained in Sec. V A. In all panels, $\circ$ symbols connected by black lines denote the reference phase boundaries of Ref. [66].

As we shall show below, this also allows the identification of three separate regions. Although the method corresponds to training $|\mathcal{J}_2| + |\mathcal{T}'| = 62$ VAEs as outlined below, this is in practice not much more involved than the single-VAE method of Sec. IV and can, arguably, be seen as less biased since we do not select a-priori specific regions to train for (cf. Sec. IV B). Before proceeding, let us streamline the notation again and denote the training set of the 62 VAEs by $\rho_{\Delta G}$.

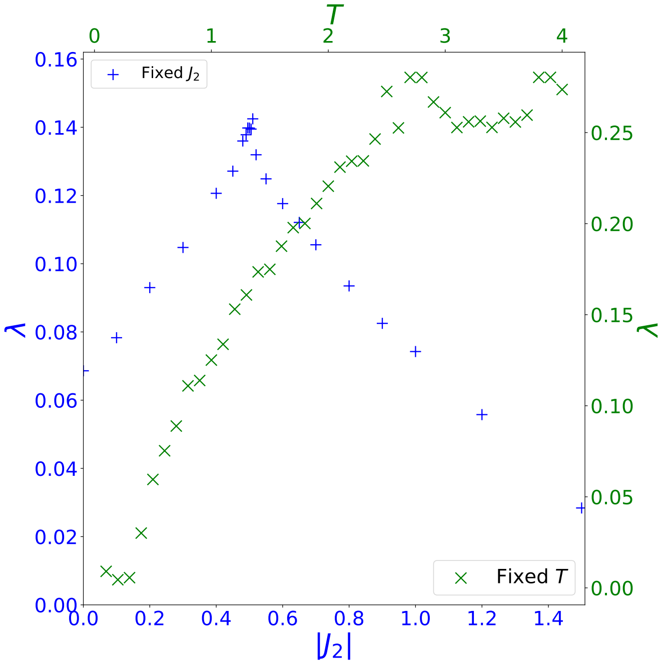

FIG. 9. Loss per spin λ obtained at epoch 500 for multi-VAE training cycles as discussed in Sec. V. Blue (+) represents the loss per spin λ at fixed-J_2 training cycle, and green (×) are the losses λ for a fixed-T training cycle.

A. Constant J_2 and T training cycles

In the first instance, we train 22 VAEs, corresponding to each of the 22 $J_2 \in \mathcal{J}_2$. The training is done for each VAE on the 40 temperatures in $\mathcal{T}'$ outlined above, using the $C = 40$ available configurations at each (T, J_2). In Fig. 8(d), we indicate this by the vertical line of white dots at $J_2 = 1.2$; the other 21 lines for the remaining J_2 values are not shown for clarity. Next, we train additional VAEs by selecting a fixed T among the possible values $0, 0.1, \ldots, 4$ and train a VAE with varying J_2 values. In Fig. 8(d), this is indicated by the horizontal line of white dots at $T = 0.25$. Since we have 40 available T's to use, this results in 40 further trained VAEs.

Figure 9 shows the corresponding losses λ at the end of a training cycle. The precise value of λ depends on details of the training cycle such as the number of configurations used. Nevertheless, at a qualitative level, we can interpret these results as follows. Deep inside an ordered phase, the VAE learns configurations well such that the loss λ is close to 0. On the other hand, the VAE is unable to learn a disordered configuration and rather returns an average gray for these, compare Fig. 3(g+h). This amounts to a loss $\lambda \approx 0.25$. When the training dataset contains a mixture of ordered and disordered configurations, the overall loss seems to be a weighted average of these two limits. The results in Fig. 9 thus reflect the fraction of the disordered configurations in the corresponding cut at fixed J_2 or T, respectively.

B. Result for averaged VAEs

Armed with the 62 trained VAEs, we can now go to each of the $|\mathcal{T}'| \times |\mathcal{J}_2| = 157 \times 22$ points in the (T, J_2) plane, compute $\varepsilon_{\rho_{\mathcal{T}'}}(T, J_2)$ for each of the 40 configurations and then average to create $\mathcal{E}_{\rho_{\mathcal{T}'}}(\mathcal{T}, \mathcal{J}_2)$. The result is shown in Fig. 8(d). As in the other panels of Fig. 8, we find a distinction between different phases. The changes from one phase to the next, when plotted for constant J_2, are very similar to Fig. 7. We find that there are three distinct regions, namely one with $\varepsilon \approx 0$, a second one with ≈ 0.25 and the third showing $\varepsilon \approx 0.17$. These regions match the superantiferromagnetic, the ferromagnetic, and the paramagnetic phases in Fig. 2 very well.

VI. CONSTRUCTING THE PHASE DIAGRAM BY COMPARING CONFIGURATIONS

Now we will investigate if one can shortcut the technical complications of the VAE and perform a direct comparison of configurations (CMP) instead. We proceed analogously to Sec. IV A and again select one of the C configurations randomly from each of the $|\mathcal{T}| \times |\mathcal{J}_2| = 3454$ pairs (T, J_2) shown in Fig. 8. We shall denote this reference set as ϱ_G. Instead of training as in the VAE, the CMP approach simply computes an averaged reconstruction error $\langle \varepsilon_{\varrho_G} \rangle (T, J_2)$ for a given test pair (T, J_2), where the mean is found by averaging over all ε for each data point in ϱ_G. In order to simplify again the notation, and since the average is somewhat implicit in the use of the subscript ϱ_G, we shall proceed by using $\varepsilon_{\varrho_G}(T, J_2)$ to denote this averaged reconstruction error.

A. Global comparison cycle

As in Sec. IV A, we have 39 data points available to compute $\varepsilon_{\varrho_G}(T, J_2)$ for each pair (T, J_2). We again choose a $J_2' \in \mathcal{J}_2$ and compute $\varepsilon_{\varrho_G}(T, J_2')$ for all $T \in \mathcal{T}$. This yields 39 values for each (T, J_2). Studying again $\varepsilon^{\min}_{\varrho_G}(T, J_2')$, Figure 10 shows results for $J_2' = 0$ and 0.4. The result is qualitatively remarkably close to that of Fig. 7, showing a change from low $\varepsilon^{\min}_{\varrho_G} \sim 0$ to values close to 0.5 at high temperatures. The values of $\varepsilon^{\min}_{\varrho_G}$ are very similar and as before, going from $L = 30$ to 60, and finally to 120 makes the change of ε_{ϱ_G} more pronounced when passing from the low-T region to the high-T one. Figure 11(a) presents an overview of the results of this approach for the full J_2-T regime for the $L = 30$ system. Overall, we can distinguish two regions: a low-temperature one with low reconstruction error ε and a high-temperature one where ε approaches 0.5.

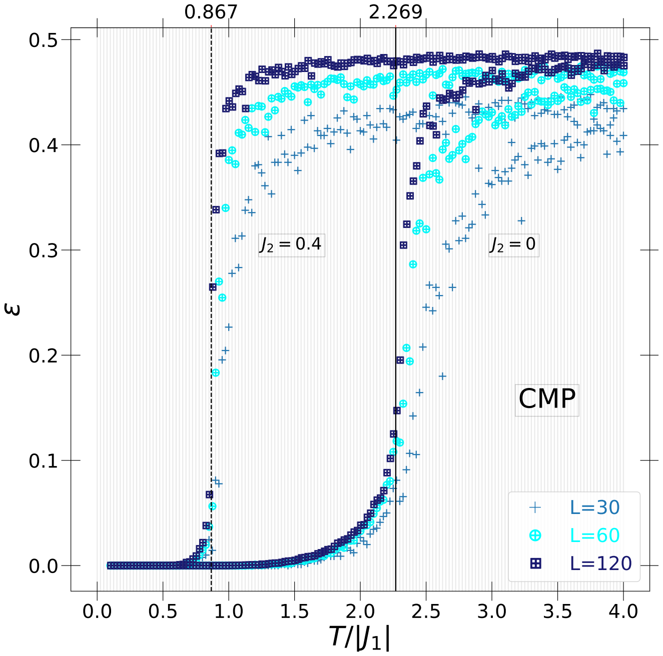

FIG. 10. Reconstruction error $\mathcal{E}^{\min}_{\rho G}(T, \mathcal{J}_2)$ obtained by direct comparison of configurations for $J_2 = 0$ and 0.4 shown for $L = 30$ (+, blue), $L = 60$ (⨁, light blue), and $L = 120$ (⊞, dark blue). As in Fig. 7, the vertical gray and black (solid and dashed) lines show the temperature resolution and positions of T_c for $J_2 = 0$ and 0.4, respectively.

B. In-phase comparison cycle

Following the reasoning of Sec. IV B, we again define two low-T regions as indicated in Figs. 11(b+c). We equip the CMP with the restricted data region, e.g., $\varrho_{\text{low-}J_2}$ and $\varrho_{\text{high-}J_2}$, and as before use the full 40 spin configurations for each (T, J_2) in $\varrho_{\text{low-}J_2}$ and $\varrho_{\text{high-}J_2}$. Figure 11(b) shows that two regions can be identified; a low-T, low-J_2 region has separated from the rest of the (T, J_2) plane. For Fig. 11(c), a similar observation allows to differentiate a low-T, high-J_2 region in the (T, J_2) plane. The ε values in the low/high-J_2 regions in Figs. 11(b+c) are close to zero, while in the other regions we have $\varepsilon \approx 0.5$. The results for the CMP approach are hence also in good agreement with the known phases as presented in Fig. 2 and we can therefore again conclude that the low-T region identified in Fig. 11(a) consists of two distinct regions.

We can combine Figs. 11(b+c) into one by considering their element-wise difference. Figure 11(d) shows the resulting phase diagram where, as in Fig. 8(d), all three phases can be distinguished clearly.

C. Discussion of reconstruction error

The VAE returned a reconstruction error $\varepsilon \approx 0.25$ in the disordered high-temperature phase (compare Figs. 7 and 8(a+d)). The CMP procedure yields instead a twice larger value $\varepsilon \approx 0.5$ (see Figs. 10 and 11). The explanation is very simple: while the VAE reproduces the gray images shown in Figs. 3(g+h) for the disordered phase, the direct calculation of ε only uses the discrete ±1 spin values. Thus, while the $\varepsilon = 0.25$ of the VAE corresponds to the squared distance of either black or white pixels to an average gray, the value $\varepsilon = 0.5$ found in the CMP procedure corresponds to the average of half of the pixels in two random images being identical and the other half is the opposite of each other. We also note that in Fig. 11, one has to be quite far from the phase boundaries to find $\varepsilon \approx 0.5$ to a good accuracy while mostly $\varepsilon \lesssim 0.5$. This behavior suggests that the configuration comparison is sensitive to the difference of each spin in any two configurations being compared. Close to the phase boundary, the system undergoes a rapid change between two phases, causing a rapid change in the values of spin representation, resulting in $\varepsilon \lesssim 0.5$.

VII. IDENTIFYING THE PHASE BOUNDARIES

Figures 7 and 10 indicate that different phases are distinguished well by different values of their reconstruction errors ε. Furthermore, we note that the sharpness of the difference in ε becomes more pronounced when increasing L. This suggests that it is possible to obtain good estimates for the phase boundaries when computing the points T, J_2 at which these changes in ε occur. In the following, we shall restrict ourselves to computing the temperature values T_* at which the phases change, using the data as shown in Figs. 7 and 10, but for all $\mathcal{J}_2$. We will estimate T_* by computing the value of temperature at the maximum of the derivative,

$$T_*(J_2) = \underset{T}{\operatorname{argmax}} \left[\frac{\partial \mathcal{E}(J_2)}{\partial T} \right]. \tag{7}$$

In order to reduce numerical fluctuations in the data, we use a simple cubic spline fit [103]. For error estimation, the $\mathcal{E}(J_2)$ data is split into two frames with alternating T values. We then calculate T_{*1} and T_{*2} in both frames and use half the absolute difference $|T_{*1} - T_{*2}|/2$ as error estimate for T_*.

Figure 12 shows the results of this analysis for the 30×30, 60×60, and 120×120 lattices. Panel (a) shows that the T_* values estimated by the derivative (7) for the VAE-based ε are indeed close to the known values for T_c and the agreement gets better when increasing L from 30 to 120, i.e., the deviations seem to be mainly due to the VAE being applied to small lattices. Furthermore, the error estimates highlight that deviations from the reference data primarily arise from finite-size effects, although not exclusively. The $L = 30$ data in panel (a) shows that the extent of these deviations varies with J_2. The T_* estimates appear closer to the reference T_c values in instances that are less noisy. Figure 12(b) shows the

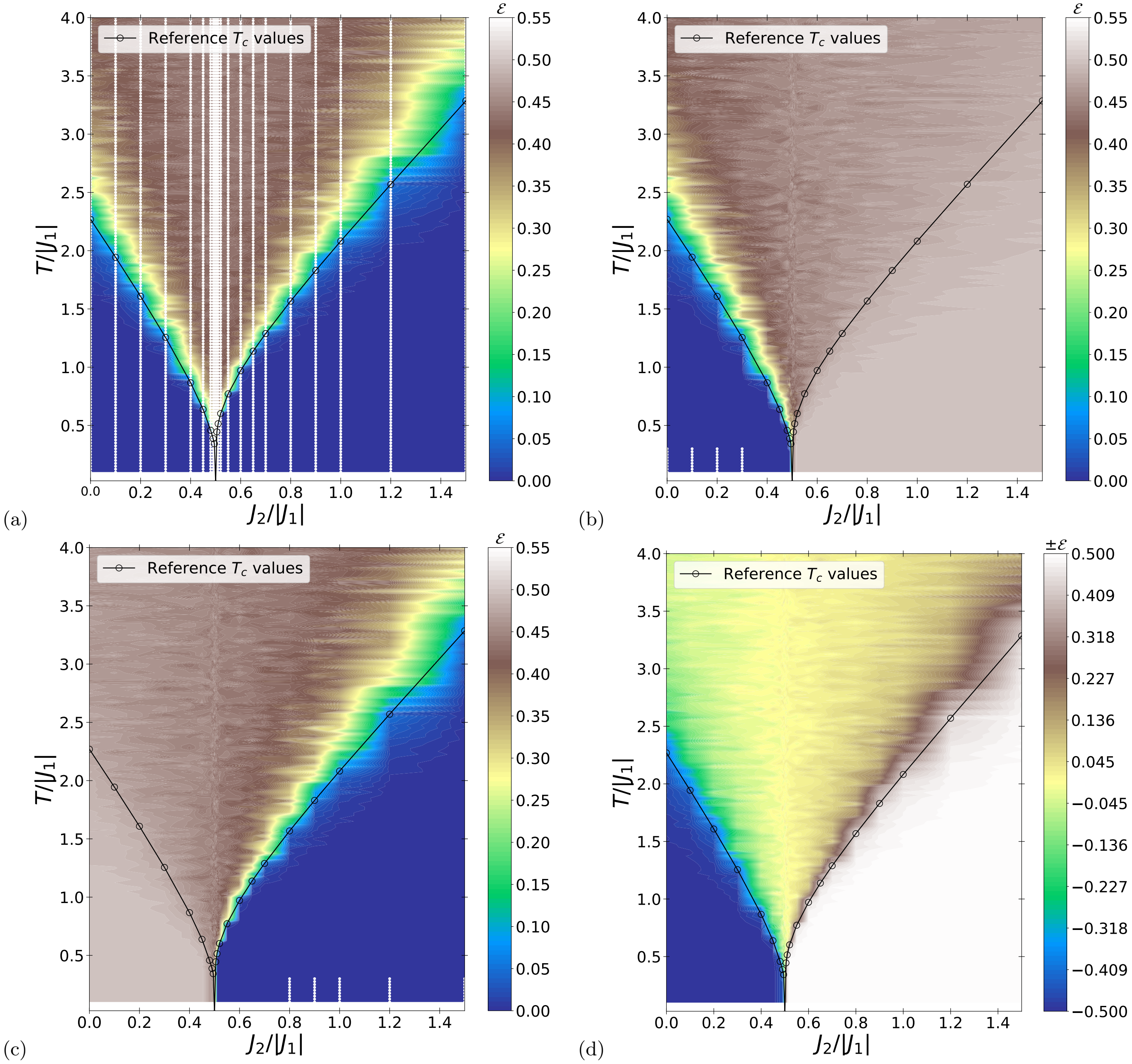

FIG. 11. Reconstruction error $\mathcal{E}^{\min}_{\rho G}(\mathcal{T},\mathcal{J}_2)$ for the CMP-based reconstruction of the phase diagram for the J_1-J_2 Ising model with $L = 30$. (a) Contour plot of all the $\mathcal{E}_{\rm CMP}(J_2)$ computed with the global comparison cycle, described in Sec. VI A, made by randomly selecting reference configurations from the entire dataset. (b) Contour plot of all the $\mathcal{E}_{\rm CMP}(J_2)$ computed with the in-phase comparison cycle, described in Sec. VI B, made by choosing reference configurations from (T, J_2) tuples such that $J_2 \in \{0.0, 0.1, 0.2, 0.3\}$ and $T \in \{0.1 \ldots 0.3\}$, also represented as white points. (c) Contour plot of all the $\mathcal{E}_{\rm CMP}(T, J_2)$ computed with the in-phase comparison cycle, made by choosing reference configurations from (T, J_2) tuples such that $J_2 \in \{0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$ and $T \in \{0.1 \ldots 0.3\}$ (VI B). (d) Element-wise difference of (b) and (c), i.e., (b)−(c). As in Fig. 8, the $\circ$ symbols connected by black lines denote the reference phase boundaries of Ref. [66] in each panel.

influence of the VAE training region on the T_* predictions for the fixed size $L = 30$. One observes that, at fixed L, the high-J_2 in-phase training cycle reproduces the reference T_c values best.

For the CMP approach, the differences between T_* and T_c are similar to the VAE: upon increasing L, the T_* values become closer to their T_c targets (Fig. 12(c)), while the in-phase-based CMP method results in the best agreement of T_* with T_c (Fig. 12(d)). In Fig. 12(c) one again observes not only better accuracy for the larger val-

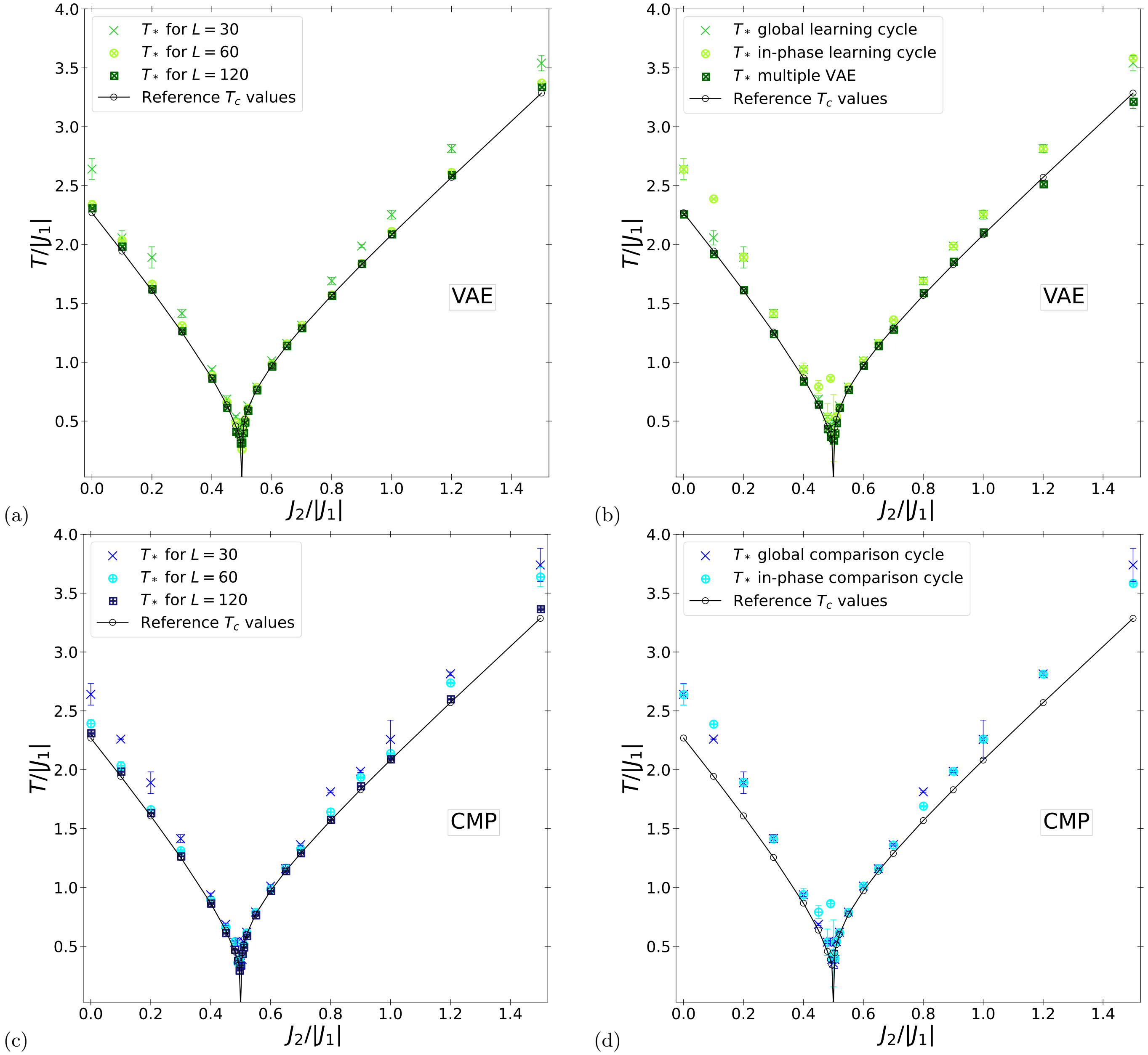

FIG. 12. Comparison of predictions for phase boundaries as determined by the VAE (a+b) and CMP (c+d) approaches. (a) Results from VAE global training cycle (Sec. IV A) for different system sizes $L = 30$ (×, green), 60 (⨂, light green), and 120 (⊠, dark green). (b) Results for $L = 30$ from VAE global training cycle (Sec. IV A, ×, green), in-phase training cycle (Sec. IV B, ⊙, light green), and using multiple VAEs (Sec. V, $*$, dark green). (c) Different L for in-phase CMP (Sec. VI B) with $L = 30$ (⊕, blue), 60 (⨁, light blue), and $L = 120$ (⊞, dark blue). (d) Comparison of global CMP (blue (+)) and in-phase CMP cycles (light blue (⊙)) with $L = 30$. In all panels, ◦ symbols connected by black lines denote the reference phase boundaries of Ref. [66].

ues of L, but also smaller statistical errors, like for the VAE in Fig. 12(a).

VIII. CONCLUSIONS

We have explored two unbiased machine-"learning" approaches to the detection of phase diagrams using the example of the J_1-J_2 Ising model on the square lattice. We found that both the variational autoencoder (VAE) and a direct comparison of configurations (CMP) can successfully detect all three phases exhibited by this model, where the main factor limiting accuracy of the location of the phase boundaries appears to be the size of the lattices employed in these investigations, both via finite-size effects and a stronger influence of statistical noise for smaller systems.

Specifically, for the VAEs presented here, we found in Sec. IV that, aiming for an unbiased reconstruction of the phases, a sequence of combinations of VAEs works best. Construction of this sequence still requires human input, so it is not yet a fully automatic "machine-led" process.

The use of VAEs to determine phases from just the spin configurations suggests that these themselves should contain sufficient information to identify phases. Our second approach using just a simple comparison of configurations establishes that this indeed is possible. In this approach, we replace the training phase of the VAE with a memory of spin configurations. We find that, for the relatively small system sizes considered here, both approaches give comparable accuracy.

Both VAE and CMP approaches yield, at least so far, limited accuracy when trying to determine the exact position of the transition points between phases. It seems that here the VAE approach is somewhat better thanks to its built-in capability to interpolate, but there is still considerable room for improvement in both strategies. One might for example in the CMP approach replace the simple MSE with more sophisticated choices such as a zero normalized cross-correlation [104] – which of course could also be used as a quantity to measure loss for the VAEs.

On reflection, both methods should be best used to determine the bulk of the phases and not so much to characterize the transition regions. As such, a more exploratory strategy suggests itself: knowing the critical temperature of the Ising model, $T_c(J_2 = 0)$, and the $T = 0$ transition at $J_2 = 1/2$ in the J_1-J_2 model, one might want to find a starting point (T, J_2) in the $T < T_c$ regime with (a) $J_2 \ll 1/2$ and (b) $J_2 \gg 1/2$. Then, e.g., for (a), one should explore locally around that starting point, with either VAE or CMP, and find other (T', J_2') such that $\varepsilon(T, J_2) \sim \varepsilon(T', J_2')$. This would explore the phase close to the starting (T, J_2). Then repeat the same for region (b). Clearly, when $\varepsilon(T, J_2) \nsim \varepsilon(T', J_2')$, one comes close to the phase boundaries. Such a strategy, when using the CMP, would only need to store the starting configuration at (T, J_2) with very little memory consumption. One could even include the newly found configurations belonging to the same phase when comparing with further configurations. With regard to a first possible application of such a strategy we note that the honeycomb lattice is topologically equivalent to a square lattice with some bonds being switched off [22, 86]. Consequently, it would be interesting to see if such a strategy is able to connect the present study to investigations of the honeycomb lattice [22, 86] by suitably varying some of the coupling constants.

If the aim instead would be to indeed use ML to determine the phase boundaries, then a more detailed finite-size scaling analysis is called for. This is beyond the aims of the present study. Still, already Refs. [10, 38, 105] presented some scaling results. However, Ref. [106] highlights that such scaling studies can suffer from large uncertainties and systematic deviations for values of critical exponents.

Also, the VAE is capable of providing additional information beyond the capabilities of the CMP. First, the generative nature of the VAE allows to construct at least approximate spin states without using the MC method. In the exploratory strategy outlined above, this means that one might be able to reduce computationally challenging MC calculations to a small training set deep in each phase and then explore the phases with VAE-generated states instead of MC-equilibrated ones. Particularly for low-T states, where equilibration is computationally intensive, this might provide a speed advantage. Second, the size of the latent space, i.e., its dimension, can provide additional information which can be useful to study other aspects of statistical models.

In conclusion, our investigations indicate that in previous work based on the reconstruction error [22, 86] a neuronal network is not fundamentally required, but that the essential idea behind uncovering the structure of the phase diagram, without manually defining an order parameter for each phase, is to actually *look* at the configurations, and this process can be automated even without resorting to a neuronal network [107]. Implementation of a direct comparison of configurations is straightforward, could still be optimized beyond the present implementation, and avoids possible complications inherent to VAEs such as the need to ensure proper training of the neuronal network.

Let us mention similar caveats that some of the present authors have recently found in the context of percolation. Not only does a neural-network approach fail to correctly reproduce the sample-to-sample fluctuations of the correlation length ξ in a supervised-learning context [54, 55], but instead of learning the physics of a *global* spanning cluster in classification of percolating versus non-percolating configurations, the network seems to rather learn how to guess this property via the proxy of the density of occupied sites in the system.

These findings call for further investigations to understand the real value of neuronal networks, in particular those designed for image-recognition tasks, when these are applied to the study of phase transitions.

ACKNOWLEDGMENTS

We thank D. Bayo (Warwick) for helpful discussions. RAR gratefully acknowledges funding by the CY Initiative of Excellence (grant "Investissements d'Avenir" ANR-16-IDEX-0008) and hospitality during various short-term stays at CY Advanced Studies. RAR also gratefully acknowledges support from the University of Warwick Research Technology Platform (RTP Scientific Computing). UK research data statement: No new data was generated.

Appendix A: The sign of J_1

Here we show that changing the sign of J_1 in the Hamiltonian (1) yields equivalent physics. It is convenient to split the site index i into two integer coordinates x, $y = 1$, …, L. Then consider the following transformation of spin variables

$$s'_{x,y} = (-1)^{x+y}\, s_{x,y}\,. \tag{A1}$$

When rewritten in terms of these new variables, the Hamiltonian (1) becomes

$$\begin{aligned} H_{J_1 J_2} &= +J_1 \sum_{x,y=1}^{L} s'_{x,y}\left(s'_{x+1,y} + s'_{x,y+1}\right) \\ &\quad +J_2 \sum_{x,y=1}^{L} s'_{x,y}\left(s'_{x+1,y+1} + s'_{x+1,y-1}\right)\,. \end{aligned} \tag{A2}$$

Clearly, the sign of J_1 has changed while the one of J_2 has remained unchanged. This implies that energy-related observables are independent of the sign of J_1, including the phase diagram. However, the precise nature of the configuration is changed; for example, the ferromagnetic state in the conventions of the main text is mapped to the antiferromagnetic one in the primed variables.

[1] N. Ashcroft and N. Mermin, *Solid State Physics* (Saunders College Publishing, Fort Worth, 1976).

[2] H. T. Diep, *Frustrated Spin Systems*, 3rd ed. (World Scientific, 2020).

[3] P. W. Anderson, More is different, Science **177**, 393 (1972).

[4] Y.-C. Yu, Y.-Y. Chen, H.-Q. Lin, R. A. Römer, and X.-W. Guan, Dimensionless ratios: Characteristics of quantum liquids and their phase transitions, Phys. Rev. B **94**, 195129 (2016).

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

[6] E. Alpaydin, *Introduction to Machine Learning, fourth edition*, Adaptive Computation and Machine Learning series (MIT Press, 2020).

[7] G. Hinton and T. J. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation* (The MIT Press, 1999).

[8] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. **810**, 1 (2019).

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

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

[11] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine learning phases of strongly correlated fermions, Phys. Rev. X **7**, 031038 (2017).

[12] A. Tanaka and A. Tomiya, Detection of phase transition via convolutional neural networks, J. Phys. Soc. Jpn. **86**, 063001 (2017).

[13] P. Huembeli, A. Dauphin, and P. Wittek, Identifying quantum phase transitions with adversarial neural networks, Phys. Rev. B **97**, 134109 (2018).

[14] X.-Y. Dong, F. Pollmann, and X.-F. Zhang, Machine learning of quantum phase transitions, Phys. Rev. B **99**, 121104(R) (2019).

[15] A. Canabarro, F. F. Fanchini, A. L. Malvezzi, R. Pereira, and R. Chaves, Unveiling phase transitions with machine learning, Phys. Rev. B **100**, 045129 (2019).

[16] K. Shinjo, K. Sasaki, S. Hase, S. Sota, S. Ejima, S. Yunoki, and T. Tohyama, Machine learning phase diagram in the half-filled one-dimensional extended Hubbard model, J. Phys. Soc. Jpn. **88**, 065001 (2019).

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

[18] K. Kottmann, P. Huembeli, M. Lewenstein, and A. Acín, Unsupervised phase discovery with deep anomaly detection, Phys. Rev. Lett. **125**, 170603 (2020).

[19] 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**, 226 (2020).

[20] F. D'Angelo and L. Böttcher, Learning the Ising model with generative neural networks, Phys. Rev. Res. **2**, 023266 (2020).

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

[22] I. Corte, S. Acevedo, M. Arlego, and C. Lamas, Exploring neural network training strategies to determine phase transitions in frustrated magnetic models, Comput. Mater. Sci. **198**, 110702 (2021).

[23] J. Wang, W. Zhang, T. Hua, and T.-C. Wei, Unsupervised learning of topological phase transitions using the Calinski-Harabaz index, Phys. Rev. Res. **3**, 013074 (2021).

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

[25] N. Walker, K.-M. Tam, B. Novak, and M. Jarrell, Identifying structural changes with unsupervised machine learning methods, Phys. Rev. E **98**, 053305 (2018).

[26] S. A. Pathak, K. Rahir, S. Holt, M. Lang, and H. Fangohr, Machine learning based classification of vector field configurations, AIP Advances **14**, 025004 (2024).

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

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

[29] 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).

[30] P. Suchsland and S. Wessel, Parameter diagnostics of phases and phase transition learning by neural networks, Phys. Rev. B **97**, 174435 (2018).

[31] 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).

[32] 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).

[33] S. Goel, Learning Ising and Potts models with latent variables, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 108, edited by S. Chiappa and R. Calandra (PMLR, 2020) pp. 3557–3566.

[34] J. Arnold, F. Schäfer, M. Žonda, and A. U. J. Lode, Interpretable and unsupervised phase classification, Phys. Rev. Res. **3**, 033052 (2021).

[35] B. Çivitcioğlu, R. A. Römer, and A. Honecker, Machine learning the square-lattice Ising model, J. Phys: Conf. Ser. **2207**, 012058 (2022).

[36] P. Basu, J. Bhattacharya, D. P. S. Jakka, C. Mosomane, and V. Shukla, Machine learning of Ising criticality with spin-shuffling (2023), arXiv:2203.04012 [cond-mat.stat-mech].

[37] D. W. Tola and M. Bekele, Machine learning of nonequilibrium phase transition in an Ising model on square lattice, Condensed Matter **8**, 83 (2023).

[38] V. Chertenkov, E. Burovski, and L. Shchur, Finite-size analysis in neural network classification of critical phenomena, Phys. Rev. E **108**, L032102 (2023).

[39] A. Naravane and N. Mathur, Semi-supervised learning of order parameter in 2D Ising and XY models using conditional variational autoencoders (2023), arXiv:2306.16822 [cond-mat.stat-mech].

[40] G. G. Pavioni, M. Arlego, and C. Lamas, Minimalist neural networks training for phase classification in diluted Ising models, Comput. Mater. Sci. **235**, 112792 (2024).

[41] H. A. Kramers and G. H. Wannier, Statistics of the two-dimensional ferromagnet. Part I, Phys. Rev. **60**, 252 (1941).

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

[43] B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model* (Harvard University Press, Cambridge, MA and London, England, 1973).

[44] W. Rządkowski, N. Defenu, S. Chiacchiera, A. Trombettoni, and G. Bighin, Detecting composite orders in layered models via machine learning, New J. Phys. **22**, 093026 (2020).

[45] K. Fukushima and K. Sakai, Can a CNN trained on the Ising model detect the phase transition of the q-state Potts model?, Prog. Theor. Exp. Phys. **2021**, 061A01 (2021).

[46] D. Giataganas, C.-Y. Huang, and F.-L. Lin, Neural network flows of low q-state Potts and clock models, New J. Phys. **24**, 043040 (2022).

[47] A. Tirelli, D. O. Carvalho, L. A. Oliveira, J. P. de Lima, N. C. Costa, and R. R. dos Santos, Unsupervised machine learning approaches to the q-state Potts model, Eur. Phys. J. B **95**, 189 (2022).

[48] Y.-H. Tseng and F.-J. Jiang, Detection of Berezinskii-Kosterlitz-Thouless transitions for the two-dimensional q-state clock models with neural networks, Eur. Phys. J. Plus **138**, 1118 (2023).

[49] Y.-H. Tseng and F.-J. Jiang, Learning the phase transitions of two-dimensional Potts model with a pre-trained one-dimensional neural network, Results in Physics **56**, 107264 (2024).

[50] F. Y. Wu, The Potts model, Rev. Mod. Phys. **54**, 235 (1982).

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

[52] W. Yu and P. Lyu, Unsupervised machine learning of phase transition in percolation, Physica A **559**, 125065 (2020).

[53] J. Shen, W. Li, S. Deng, and T. Zhang, Supervised and unsupervised learning of directed percolation, Phys. Rev. E **103**, 052140 (2021).

[54] D. Bayo, A. Honecker, and R. A. Römer, Machine learning the 2D percolation model, J. Phys: Conf. Ser. **2207**, 012057 (2022).

[55] D. Bayo, A. Honecker, and R. A. Römer, The percolating cluster is invisible to image recognition with deep learning, New J. Phys. **25**, 113041 (2023).

[56] S. Patwardhan, U. Majumder, A. D. Sarma, M. Pal, D. Dwivedi, and P. K. Panigrahi, Machine learning as an accurate predictor for percolation threshold of diverse networks (2023), arXiv:2212.14694v2 [physics.soc-ph].

[57] R. H. Swendsen and S. Krinsky, Monte Carlo renormalization group and Ising models with $n \geq 2$, Phys. Rev. Lett. **43**, 177 (1979).

[58] D. P. Landau, Phase transitions in the Ising square lattice with next-nearest-neighbor interactions, Phys. Rev. B **21**, 1285 (1980).

[59] K. Binder and D. P. Landau, Phase diagrams and critical behavior in Ising square lattices with nearest- and next-nearest-neighbor interactions, Phys. Rev. B **21**, 1941 (1980).

[60] D. P. Landau and K. Binder, Phase diagrams and critical behavior of Ising square lattices with nearest-, next-nearest-, and third-nearest-neighbor couplings, Phys. Rev. B **31**, 5946 (1985).

[61] J. L. Morán-López, F. Aguilera-Granja, and J. M. Sanchez, First-order phase transitions in the Ising square lattice with first- and second-neighbor interactions, Phys. Rev. B **48**, 3519 (1993).

[62] H. J. W. Zandvliet, The 2D Ising square lattice with nearest- and next-nearest-neighbor interactions, Europhys. Lett. **73**, 747 (2006).

[63] A. Malakis, P. Kalozoumis, and N. Tyraskis, Monte Carlo studies of the square Ising model with next-nearest-neighbor interactions, Eur. Phys. J. B **50**, 63 (2006).

[64] J. L. Monroe and S.-Y. Kim, Phase diagram and critical exponent ν for the nearest-neighbor and next-nearest-neighbor interaction Ising model, Phys. Rev. E **76**, 021123 (2007).

[65] R. A. dos Anjos, J. Roberto Viana, and J. Ricardo de Sousa, Phase diagram of the Ising antiferromagnet with nearest-neighbor and next-nearest-neighbor interactions on a square lattice, Phys. Lett. A **372**, 1180 (2008).

[66] A. Kalz, A. Honecker, S. Fuchs, and T. Pruschke, Phase diagram of the Ising square lattice with competing interactions, Eur. Phys. J. B **65**, 533 (2008).

[67] H. Zandvliet and C. Hoede, Spontaneous magnetization of the square 2D Ising lattice with nearest- and weak next-nearest-neighbour interactions, Phase Trans. **82**, 191 (2009).

[68] A. Kalz, A. Honecker, and M. Moliner, Analysis of the phase transition for the Ising model on the frustrated square lattice, Phys. Rev. B **84**, 174407 (2011).

[69] S. Jin, A. Sen, and A. W. Sandvik, Ashkin-Teller criticality and pseudo-first-order behavior in a frustrated Ising model on the square lattice, Phys. Rev. Lett. **108**, 045702 (2012).

[70] A. Kalz and A. Honecker, Location of the Potts-critical end point in the frustrated Ising model on the square lattice, Phys. Rev. B **86**, 134410 (2012).

[71] S. Jin, A. Sen, W. Guo, and A. W. Sandvik, Phase transitions in the frustrated Ising model on the square lattice, Phys. Rev. B **87**, 144406 (2013).

[72] A. Bobák, T. Lučivjanský, M. Borovský, and M. Žukovič, Phase transitions in a frustrated Ising antiferromagnet on a square lattice, Phys. Rev. E **91**, 032145 (2015).

[73] M. Ramazanov, A. Murtazaev, and M. Magomedov, Thermodynamic, critical properties and phase transitions of the Ising model on a square lattice with competing interactions, Solid State Commun. **233**, 35 (2016).

[74] P. Patil and A. W. Sandvik, Hilbert space fragmentation and Ashkin-Teller criticality in fluctuation coupled Ising models, Phys. Rev. B **101**, 014453 (2020).

[75] H. Li and L.-P. Yang, Tensor network simulation for the frustrated J_1-J_2 Ising model on the square lattice, Phys. Rev. E **104**, 024118 (2021).

[76] Y. Hu and P. Charbonneau, Numerical transfer matrix study of frustrated next-nearest-neighbor Ising models on square lattices, Phys. Rev. B **104**, 144429 (2021).

[77] H. J. W. Zandvliet, Phase diagram of the square 2D Ising lattice with nearest neighbor and next-nearest neighbor interactions, Phase Trans. **96**, 187 (2023).

[78] V. A. Abalmasov and B. E. Vugmeister, Metastable states in the J_1-J_2 Ising model, Phys. Rev. E **107**, 034124 (2023).

[79] H. Watanabe, Y. Motoyama, S. Morita, and N. Kawashima, Non-monotonic behavior of the Binder parameter in discrete spin systems, Prog. Theor. Exp. Phys. **2023**, 033A02 (2023).

[80] K. Yoshiyama and K. Hukushima, Higher-order tensor renormalization group study of the $J_1 - J_2$ Ising model on a square lattice, Phys. Rev. E **108**, 054124 (2023).

[81] A. A. Gangat, Weak first-order phase transitions in the frustrated square lattice $J_1 - J_2$ classical Ising model, Phys. Rev. B **109**, 104419 (2024).

[82] V. A. Abalmasov, Free energy and metastable states in the square-lattice J_1-J_2 Ising model, SciPost Phys. **16**, 151 (2024).

[83] J. H. Lee, S.-Y. Kim, and J. M. Kim, Frustrated Ising model with competing interactions on a square lattice, Phys. Rev. B **109**, 064422 (2024).

[84] S.-W. Li and F.-J. Jiang, A comprehensive study of the phase transitions of the frustrated J_1-J_2 Ising model on the square lattice, Prog. Theor. Exp. Phys. **2024**, 053A06 (2024).

[85] H. Park and H. Lee, Frustrated Ising model on D-wave quantum annealing machine, J. Phys. Soc. Jpn. **91**, 074001 (2022).

[86] S. Acevedo, M. Arlego, and C. A. Lamas, Phase diagram study of a two-dimensional frustrated antiferromagnet via unsupervised machine learning, Phys. Rev. B **103**, 134422 (2021).

[87] D. P. Kingma and M. Welling, Auto-encoding variational Bayes (2022), arXiv:1312.6114v11 [stat.ML].

[88] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[89] J. Arnold and F. Schäfer, Replacing neural networks by optimal analytical predictors for the detection of phase transitions, Phys. Rev. X **12**, 031044 (2022).

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

[91] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, 1999).

[92] B. A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis with Web-Based Fortran Code* (World Scientific Publishing Company, 2004).

[93] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, 4th ed. (Cambridge University Press, 2014).

[94] L. Li, Application of deep learning in image recognition, J. Phys: Conf. Ser. **1693**, 012128 (2020).

[95] J. Chou, Generated loss and augmented training of MNIST VAE (2019), arXiv:1904.10937 [cs.LG].

[96] S. Kullback and R. A. Leibler, On Information and Sufficiency, The Annals of Mathematical Statistics **22**, 79 (1951).

[97] T. Luhman and E. Luhman, High fidelity image synthesis with deep VAEs in latent space (2023), arXiv:2303.13714 [cs.CV].

[98] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in *2009 IEEE conference on computer vision and pattern recognition* (IEEE, 2009) pp. 248–255.

[99] A. Asperti, D. Evangelista, and E. Loli Piccolomini, A survey on variational autoencoders from a green AI perspective, SN Computer Science **2**, 301 (2021).

[100] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, Hyperspherical variational auto-encoders, in *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, edited by A. Glober-

son and R. Silva (Association For Uncertainty in Artificial Intelligence (AUAI), 2018) pp. 856–865.

[101] B. Çivitcioğlu, *Phase determination with and without deep learning for the J_1-J_2 Ising model*, Ph.D. thesis, CY Cergy Paris Université (2024).

[102] We note that in the image comparison context, there is no "reconstuction" such that Eq. (2) should rather be considered as a measure for the difference of two configurations. By slight abuse of terminology we will nevertheless continue to call this quantity "reconstruction error".

[103] G. Phillips and P. Taylor, *Theory and Applications of Numerical Analysis* (Elsevier Science, 1996).

[104] A. Papoulis, *The Fourier integral and its applications*, McGraw-Hill electronic sciences series (McGraw-Hill New York, New York, 1962).

[105] Z. Li, M. Luo, and X. Wan, Extracting critical exponents by finite-size scaling with convolutional neural networks, Phys. Rev. B **99**, 075418 (2019).

[106] H. Théveniaut and F. Alet, Neural network setups for a precise detection of the many-body localization transition: Finite-size scaling and limitations, Phys. Rev. B **100**, 224202 (2019).

[107] In fact, this is reminiscent of the Edwards-Anderson order parameter for spin glasses [108]. It is just remarkable that this idea seems not to have been applied to clean systems before the advent of ML.

[108] S. F. Edwards and P. W. Anderson, Theory of spin glasses, J. Phys. F: Met. Phys. **5**, 965 (1975).