Fine-tuning neural network quantum states

Riccardo Rende,[1,*] Sebastian Goldt,[1,†] Federico Becca,[2] and Luciano Loris Viteritti[2,‡]
[1]*International School for Advanced Studies (SISSA), Via Bonomea 265, I-34136 Trieste, Italy*
[2]*Dipartimento di Fisica, Università di Trieste, Strada Costiera 11, I-34151 Trieste, Italy*

(Received 20 March 2024; accepted 4 November 2024; published 16 December 2024)

Recent progress in the design and optimization of neural-network quantum states (NQSs) has made them an effective method to investigate ground-state properties of quantum many-body systems. In contrast to the standard approach of training a separate NQS from scratch at every point of the phase diagram, we demonstrate that the optimization of a NQS at a highly expressive point of the phase diagram (i.e., close to a phase transition) yields features that can be reused to accurately describe a wide region across the transition. We demonstrate the feasibility of our approach on different systems in one and two dimensions by initially pretraining a NQS at a given point of the phase diagram, followed by fine-tuning only the output layer for all other points. Notably, the computational cost of the fine-tuning step is very low compared to the pretraining stage. We argue that the reduced cost of this paradigm has significant potential to advance the exploration of strongly correlated systems using NQS, mirroring the success of fine-tuning in machine learning and natural language processing.

DOI: 10.1103/PhysRevResearch.6.043280

I. INTRODUCTION

Over the last decade, neural networks have emerged as the most important general-purpose machine learning tool [1,2]. Their versatility is evident in the most recent generation of neural networks based on the Transformer architecture [3], which was initially designed for natural language processing, and is now achieving state-of-the-art performance in fields as diverse as text generation [4,5], computer vision [6], and protein contact prediction [7]. The success of deep neural networks is generally attributed to their ability to learn relevant features directly from data [1,2]. This approach replaces the classical one, where one first designs a set of well-suited features to describe the inputs and then trains a simple machine learning algorithm (e.g., linear regression) to perform a given task. However, hand-crafting suitable features is feasible only for simple problems and becomes unfeasible when dealing with complicated tasks. By contrast, deep-learning methods learn good representations to solve the target task directly from raw data, outperforming hand-crafted representations in a variety of domains.

In the last few years, neural networks are also increasingly used in condensed matter physics to approximate low-energy states of many-body quantum systems, for both spin and fermionic models [8–17]. In this context, neural-network quantum states (NQSs) parametrize the amplitude of a variational state $|\Psi_\theta\rangle$ expanded in a proper basis $\{|\boldsymbol{\sigma}\rangle\}$, mapping input physical configurations $\boldsymbol{\sigma}$ to complex numbers $\langle\boldsymbol{\sigma}|\Psi_\theta\rangle = \Psi(\boldsymbol{\sigma};\theta)$. Within the variational Monte Carlo framework, the parameters θ of the state are optimized to minimize the variational energy $E_\theta = \langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle/\langle\Psi_\theta|\Psi_\theta\rangle$ [18]. Recently, a new parametrization of NQS which explicitly leverages the feature learning perspective has been introduced in Ref. [19]. In this viewpoint, a deep neural network is used to construct a map from the space of the physical configurations to abstract representations in a feature space, where the determination of the low-energy properties of the systems is simplified, and then a simple shallow network transform these representations into complex numbers. The resulting architecture achieved state-of-the-art performance on one of the most famous benchmark problems of the field [20].

In this work we address a key conceptual question that this approach raises: given a system that exhibits a phase transition, *do the representations that have learned to approximate the ground state near the transition point generalize to other points of the phase diagram?* This question holds significance not only from a theoretical perspective but also from a practical one, as it provides a concrete advantage of avoiding the need of optimizing the wave function from scratch for each point in the phase diagram.

II. METHODS

Mirroring the feature learning perspective, we represent the NQS as the composition of two functions [19]:

$$\begin{aligned} z &= V(\boldsymbol{\sigma};\phi), \\ \log[\Psi(\boldsymbol{\sigma};\theta)] &= f(z;W), \end{aligned} \tag{1}$$

*Contact author: rrende@sissa.it
†Contact author: sgoldt@sissa.it
‡Contact author: lucianoloris.viteritti@phd.units.it

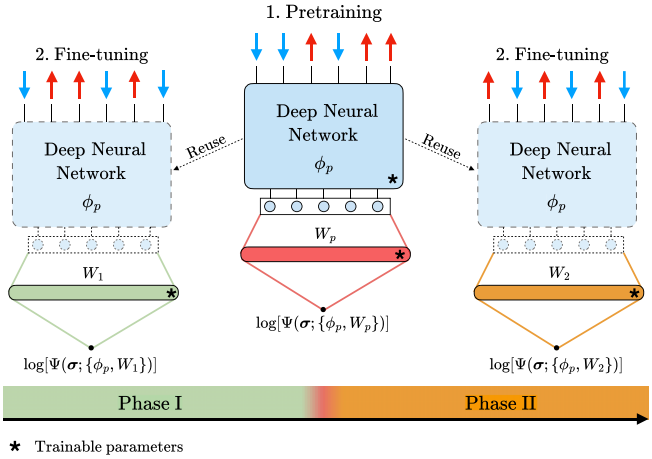

FIG. 1. Graphical representation of the pretraining and fine-tuning procedures. Initially, during the pretraining, the entire architecture is trained in proximity to the transition point of a given system, yielding a set of parameters $\theta_p = \{\phi_p, W_p\}$. Subsequently, in the fine-tuning stage, the parameters of the deep neural network ϕ_p are fixed, while the optimization process focuses exclusively on the weights of the shallow network W at various points across the phase diagram.

where we have partitioned the variational parameters into two blocks $\theta = \{\phi, W\}$. The function $V(\boldsymbol{\sigma};\phi)$ is parameterized through a *deep* neural network, mapping physical configurations $\boldsymbol{\sigma}$ into a feature space, thereby generating for each input configuration $\boldsymbol{\sigma}$ a hidden representation $\boldsymbol{z}(\boldsymbol{\sigma}) \in \mathbb{R}^d$, with d the dimension of the feature space (an adjustable hyperparameter of the network). Conversely, $f(\boldsymbol{z};W)$ is a *shallow* fully connected neural network used to generate a single scalar value $f(\boldsymbol{z};W) \in \mathbb{C}$ from hidden representations $\boldsymbol{z}$.

Given a system undergoing a phase transition, we want to investigate whether the representations learned near the transition point may be generalized to other points of the phase diagram. We perform the following experiment in two steps, as illustrated in Fig. 1:

(1) We *pretrain* the entire network, optimizing it to approximate the ground state at a single point of the phase diagram situated in the vicinity of the phase transition. The pretraining stage yields a set of optimized parameters $\{\phi_p, W_p\}$.

(2) Using the features constructed by the deep network (thus fixing its variational parameters ϕ_p) we *fine-tune* the model by optimizing only the parameters W of the output layer to approximate the ground states in the other points of the phase diagram, on both sides of the phase transition.

The pretraining of the architecture is carried out near the critical point, where long-range correlations are present and may be established in the body of the NQS. Then the last (shallow) output layer, which is fine-tuned in a different point in the phase diagram, can either reinforce correlations and establish true long-range order or weaken them and yield to a short-range state (or even keep the state critical). On the other hand, in trivial phases, where only a few configurations have non-zero amplitudes, the ability of pretrained networks to generalize away from these phases is limited (see Appendix A).

We apply this procedure on finite systems and measure physical properties (e.g., order parameters) of various systems exhibiting, in the thermodynamic limit, phase transitions of different nature. In all cases, the features extracted during the pretraining stage, close to transition points, lead to excellent results after fine-tuning at the other points of the phase diagram [21]. This result reflects how neural networks can capture the essential quantum fluctuations in the vicinity of a phase transition. We stress that this approach differs from the standard transfer-learning paradigm in which a neural network is initially trained to solve a specific task, and then all of the parameters are trained to solve a different task. To the best of our knowledge, fine-tuning experiments on NQS have not been explored previously.

The methodology outlined here is generally applicable to any deep neural network, but to be concrete, in the following we parametrize the function $V(\boldsymbol{\sigma};\phi)$ using a Vision Transformer (ViT) with real-valued parameters [19,20]. This choice is suggested by the flexibility of the Transformer architecture [3,6], already used to achieve highly accurate results in various types of systems, both one- and two-dimensional [19,20,22–26]. The hyperparameters of this architecture are the number of heads h of the Multi-Head Factored Attention Mechanism [27], the embedding dimension d, and the number of layers n_l (a detailed description of the architecture is reported in Ref. [19]). In addition, the function $f(\boldsymbol{z},W)$ is defined as

$$f(\boldsymbol{z};W) = \sum_{\alpha=1}^{K} g(b_\alpha + \boldsymbol{w}_\alpha \cdot \boldsymbol{z}), \tag{2}$$

where the number of neurons K is chosen to be equal to d and $2 \times d$ in the pretraining and in the fine-tuning steps, respectively. In order to describe nonpositive ground states, the parameters $W = \{b_\alpha, \boldsymbol{w}_\alpha\}_{\alpha=1}^{K}$ of the linear transformation in Eq. (2) are taken to be complex valued. Here we set $g(\cdot) = \log\cosh(\cdot)$, thus $f(\boldsymbol{z},W)$ represents the well-known Restricted Boltzmann Machine (RBM) introduced by Carleo and Troyer [8]. The crucial difference is that in this case it is not applied directly on physical configurations $\boldsymbol{\sigma}$, but instead on hidden representations $\boldsymbol{z}$ generated by the Transformer [19]. We remark that this framework offers a huge computational advantage, since it requires the costly optimization of the full architecture, including the feature extractor $V(\boldsymbol{\sigma};\phi)$, only once in the pretraining step. Then, with the addition of a minimal cost, the targeted optimization of the RBM can be used to obtain an accurate description of the physical properties of the system in a wide region across the transition point. In what follows we focus on spin $S = 1/2$ models on a lattice considering system sizes where numerically exact solutions are available for comparison. In this case $\Psi(\boldsymbol{\sigma};\theta)$ refers to the amplitude of the variational state $|\Psi_\theta\rangle$ in the computational basis having a definite spin value in the z direction, i.e., $\{|\boldsymbol{\sigma}\rangle = |\sigma_1^z, \ldots, \sigma_N^z\rangle\}$ with $\sigma_i^z = \pm 1$.

Finally, we mention a few details on the Monte Carlo procedure adopted here. During the pretraining stage, we perform $N_{\text{opt}} = 10\,000$ optimization steps. Then, during the fine-tuning stage, the number of steps is reduced to $N_{\text{opt}} =$

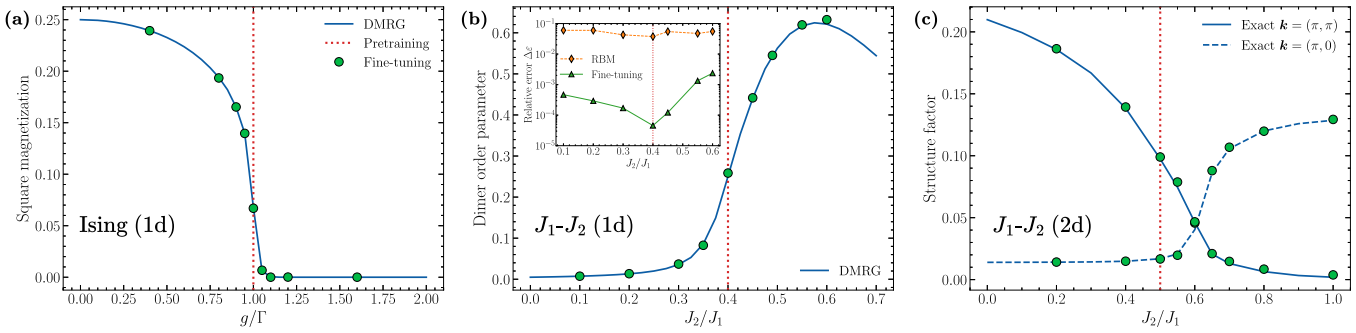

FIG. 2. (a) Ising chain. A ViT with hyperparameters $h = 12$, $d = 72$, $n_l = 4$ is pretrained at $g/\Gamma = 1$, on a chain with $N = 100$ sites. After the fine-tuning, the square magnetization order parameter is computed and compared to DMRG results (bond dimension $\chi = 10^3$). (b) Heisenberg J_1-J_2 chain. A ViT with hyperparameters $h = 12$, $d = 192$, $n_l = 4$ is pretrained at $J_2/J_1 = 0.4$, on a chain with $N = 100$ sites. After the fine-tuning, the dimer order parameter is computed and compared to DMRG results ($\chi = 10^3$). Inset: Relative error $\Delta\varepsilon$ (with respect to DMRG) of the same fully connected network (RBM) trained on the hidden representations generated by the pretrained ViT and directly on configurations. (c) Two-dimensional Heisenberg J_1-J_2. A ViT with hyperparameters $h = 18$, $d = 216$, $n_l = 8$ is pretrained at $J_2/J_1 = 0.5$, on a 6×6 square lattice. After the fine-tuning, the structure factors at $\boldsymbol{k} = (\pi, \pi)$ and $\boldsymbol{k} = (0, \pi)$ are computed and compared to exact diagonalization results.

3000. For all the simulations, we estimate stochastically the observables choosing a number of samples of $M = 3000$. The optimization of the variational parameters is performed with the Stochastic Reconfiguration (SR) method [28]. In particular, working with variational states featuring approximately $P = 10^6$ parameters, we employ the alternative formulation of SR [20,29] efficient in the regime $P \gg M$ (available in NetKet [30] under the name of `VMC_SRt`). We use a cosine decay scheduler for the learning rate, setting the initial value to $\tau = 0.03$.

III. RESULTS

A. The one-dimensional quantum Ising model

We start by considering the one-dimensional Ising model in transverse magnetic field, described by the following Hamiltonian:

$$\hat{H} = -\Gamma \sum_{i=1}^{N} \hat{S}_i^z \hat{S}_{i+1}^z - g \sum_{i=1}^{N} \hat{S}_i^x, \tag{3}$$

where $\hat{S}_i^x$ and $\hat{S}_i^z$ are spin-1/2 operators on site i. The ground-state wave function, for $g \geqslant 0$, is positive definite in the computational basis.

In the thermodynamic limit, the ground state exhibits a second-order phase transition at $g/\Gamma = 1$, from a ferromagnetic ($g/\Gamma < 1$) to a paramagnetic ($g/\Gamma > 1$) phase. On finite systems with N sites, the estimation of the critical point can be obtained from the long-range behavior of the spin-spin correlations, i.e., $m^2(r) = 1/N \sum_{i=1}^{N} \langle \hat{S}_i^z \hat{S}_{i+r}^z \rangle$ (specifically, we can consider the largest distance $r = N/2$, which gives the square magnetization).

First, we pretrain the full architecture at the critical point $g/\Gamma = 1$. Then we fine-tune only the output layer at different values of the external field, from $g/\Gamma = 0.4$ to $g/\Gamma = 1.6$, i.e., in both ferromagnetic and paramagnetic phases. The results for $N = 100$ with periodic-boundary conditions are shown in Fig. 2(a) compared with density-matrix renormalization group (DMRG) [31] calculations (on the same system). The high level of accuracy demonstrates that the fine-tuned network is effective in the prediction of the order parameter. Remarkably, the fine-tuning procedure involves optimizing merely 6.6% of the total parameters, which is ten times faster than optimizing the entire network and demands significantly less GPU memory (see Appendix B for a detailed description of the GPU memory requirements). The remarkable fact is that, by exclusively adjusting the parameters of the output (fully connected) layer and keeping the clusters of the hidden representation fixed, it is possible to effectively describe both ordered and disordered phases.

In the following, we want to gain insights into the learning process of the fine-tuning stage. For that, we sample a set of M configurations $\{\boldsymbol{\sigma}_i\} \sim |\Psi(\boldsymbol{\sigma}; \theta_p)|^2$ from the pretrained network and show the corresponding amplitudes after the fine-tuning procedure [visualizing them on top of UMAP [32] projections of the hidden representations $\boldsymbol{z}_p(\boldsymbol{\sigma}_i)$, for $i = 1, \ldots, M$]; see Fig. 3. To highlight the differences, both color and size of each point are proportional to their amplitudes. At the transition point ($g/\Gamma = 1$), the configurations with all parallel spins (either up or down along z) have the largest amplitude; other configurations, with a few spin flips have also considerable weights (see middle panel). In the ordered phase ($g/\Gamma = 0.4$), only one of these fully polarized configurations is "selected," i.e., frequently visited along the Monte Carlo sampling, and the amplitudes for all other configurations are practically negligible (left panel). This effect is related to the difficulty of simple sampling approaches (that performs local spin flips) to overcome the (large) barrier that separates the two ground states, which are almost degenerate on finite systems. By contrast, in the disordered phase ($g/\Gamma = 1.6$), many configurations have similar amplitudes: the two fully polarized configurations showing a reduced weight compared to all the others (right panel). A brief discussion about the connection between the features extracted by the NQS and the order parameter is given in Appendix C.

B. The one-dimensional J_1-J_2 Heisenberg model

In order to assess the accuracy of our method on more complicated systems, specifically with nonpositive ground states

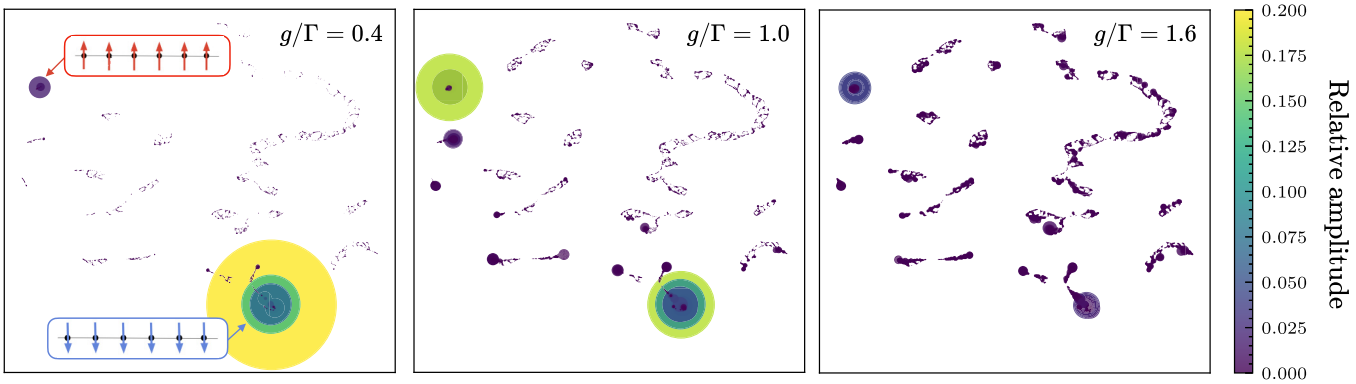

FIG. 3. Dimensional reduction of the hidden representations for a set of $M = 3000$ configurations built using a ViT pretrained at $g/\Gamma = 1$ with hyperparameters $h = 12$, $d = 72$, and $n_l = 4$ for a system of $N = 100$ sites. The data points represent UMAP projections of vectors z. Both the colors and sizes of the points are related to the amplitudes predicted after the fine-tuning procedure at three distinct points along the phase diagram: ordered phase $g/\Gamma = 0.4$ (left panel), transition point $g/\Gamma = 1$ (central panel), and disordered phase $g/\Gamma = 1.6$ (right panel).

in the computational basis, we investigate the frustrated J_1-J_2 Heisenberg model:

$$\hat{H} = J_1 \sum_{\langle i,j \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j, \tag{4}$$

where $\hat{\mathbf{S}}_i = (S_i^x, S_i^y, S_i^z)$ and $J_1 > 0$ and $J_2 \geqslant 0$ are antiferromagnetic couplings for nearest and next-nearest neighbors, respectively.

We first discuss the results in one dimension. Here the ground-state phase diagram shows two phases, separated by a Berezinskii-Kosterlitz-Thouless transition at $(J_2/J_1)_c = 0.24116(7)$ [33]: a gapless phase with no order whatsoever and a gapped one, with long-range dimer order. On finite systems, the latter one may be extracted from the long-distance behavior of the dimer-dimer correlation functions $D(r) = \langle \hat{S}_1^z \hat{S}_2^z \hat{S}_r^z \hat{S}_{r+1}^z \rangle - \langle \hat{S}_1^z \hat{S}_2^z \rangle \langle \hat{S}_r^z \hat{S}_{r+1}^z \rangle$ [34,35]. Specifically, performing a finite-size scaling, an estimation of the dimer order parameter can be obtained as $D^2 = 9|D(N/2-1) - 2D(N/2) + D(N/2+1)|$ [34,35]. However, we emphasize that the order parameter is exponentially small close to the transition, making it difficult to extract an accurate estimation of the actual value of $(J_2/J_1)_c$ (indeed, the location of the transition may be easily obtained by looking at the level crossing between the lowest-energy triplet and singlet excitations [33]). As before, we pretrain at a given point, here $J_2/J_1 = 0.4$, and optimize the output layer of the network for different values of the frustrating ratio, in both the gapless and gapped regions. The results for $N = 100$ (with periodic boundary conditions) are reported in Fig. 2(b), again compared to DMRG calculations on the same system. In addition, in the inset of Fig. 2(b), we compare the relative energy error $\Delta\varepsilon$ (with respect to the DMRG energies) of an RBM trained directly on the physical configurations [8] and of a fine-tuned ViT. This analysis underscores the importance of exploiting the features constructed by the pretrained ViT, resulting in an accuracy gain of more than two orders of magnitude with respect to the same network trained directly on configurations. For completeness, we report the ground-state energies for various frustration ratios J_2/J_1 in Table I. They are obtained through three distinct methodologies: DMRG, ViT trained from scratch, and ViT pretrained at $J_2/J_1 = 0.4$ and subsequently fine-tuned for other frustration ratios. Notably, the fine-tuned ViT exhibits remarkable accuracy when compared to DMRG results, reaching an accuracy $\Delta\varepsilon \lesssim 10^{-3}$ for all the values of the frustration ratio in the interval $J_2/J_1 \in [0.1, 0.6]$. Let us move to the discussion of how the output layer can modify the sign structure during the fine-tuning step. For the J_1-J_2 Heisenberg chain, the sign structure of the ground-state wave function is not known except for $J_2 = 0$, where the so-called Marshall Sign Rule (MSR) [36] applies. However, even for large system sizes, the MSR constitutes an accurate approximation of the sign structure up to $J_2/J_1 \leqslant 0.5$ [37]. In Fig. 4 we show the predicted phases (0 or π), on top of the UMAP projections of the vectors z_p generated by the pretrained network at $J_2/J_1 = 0.4$. At $J_2/J_1 = 0.1$ [see Fig. 4(a)], after the fine-tuning procedure, the signs exactly match the ones obtained at $J_2/J_1 = 0.4$ (not shown). This is because, at the pretraining point, where the clusters are formed, the MSR remains a highly accurate approximation of the ground-state sign structure. By contrast, for $J_2/J_1 = 0.6$, this is no longer true, and the output layer must adjust the phases accordingly [see Fig. 4(b)]; still, the fine-tuned ViT performs better than a RBM trained on spin configurations; see Appendix D for a detailed discussion.

TABLE I. Variational ground-state energies for the J_1-J_2 Heisenberg chain with system size $N = 100$. The DMRG computations are conducted employing a bond dimension up to $\chi = 10^3$ under periodic boundary conditions. For both instances involving ViT, one trained from scratch and another pretrained at $J_2/J_1 = 0.4$ followed by fine-tuning, the Monte Carlo error attributed to finite sampling effects affects the last digit of the reported results.

J_2/J_1	DMRG	ViT	Fine-tuning
0.10	−0.425417395	−0.4254174	−0.425218
0.20	−0.408572967	−0.4085728	−0.408453
0.30	−0.393126745	−0.3931204	−0.393059
0.40	−0.380387370	−0.3803726	−0.380370
0.60	−0.380804138	−0.3807913	−0.379902

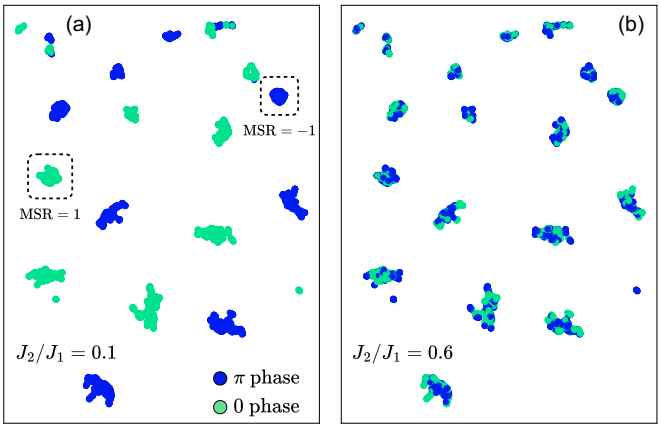

FIG. 4. Graphical representation of the hidden representations for the J_1-J_2 Heisenberg chain. The data points, corresponding to a sample of $M = 3000$ physical configurations, represent UMAP projections of vectors z_p generated by a ViT with hyperparameters $h = 12$, $d = 192$, and $n_l = 4$, pretrained at the point $J_2/J_1 = 0.4$ for a system size of $N = 100$. The depicted colors correspond to the predicted phases (0 or π) after fine-tuning at two specific points within the phase diagram: $J_2/J_1 = 0.1$ (a) and $J_2/J_1 = 0.6$ (b). Panel (a) reveals a close resemblance between the cluster structure identified during the pretraining at $J_2/J_1 = 0.4$, which matches the Marshall Sign Rule.

C. The two-dimensional J_1-J_2 Heisenberg model

Finally, we consider the two-dimensional J_1-J_2 Heisenberg model on an $L \times L$ square lattice. The ground state of this model features magnetic order in the two limits $J_1 \ll J_2$ and $J_1 \gg J_2$. Its presence can be characterized with the spin structure factor $S(\boldsymbol{k}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} \langle \hat{\boldsymbol{S}}_{\boldsymbol{0}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R}} \rangle$, where $\boldsymbol{R}$ runs over all the lattice sites of the square lattice. Specifically, when $J_2 = 0$ the model reduces to the unfrustrated Heisenberg model where long-range Néel order is present [38,39]. The latter one can be detected by measuring $m^2_{\text{Néel}} = S(\pi, \pi)/L^2$. In the opposite regime $J_2/J_1 \to \infty$, the system exhibits instead columnar magnetic order, identified by the order parameter $m^2_{\text{stripe}} = [S(0, \pi) + S(\pi, 0)]/(2L^2)$. In the intermediate region, around $J_2/J_1 \approx 0.5$ the system is highly frustrated and the nature of the ground state is still under debate [10,40–43]. Here we limit ourselves to the 6×6 system, where exact diagonalizations are possible (no DMRG calculations on the structure factor are available on larger systems with periodic boundary conditions). We first perform the pretraining at $J_1/J_2 = 0.5$, then perform the fine-tuning for $0.2 < J_2/J_1 < 1$ and evaluate the order parameters $m^2_{\text{Néel}}$ and m^2_{stripe}; see Fig. 2(c). Remarkably, even for this complicated two-dimensional model, the correct behavior of the two magnetic order parameters can be reconstructed with great accuracy starting from a single pretrained ViT.

IV. CONCLUSIONS

We showed that, for several physical systems exhibiting phase transitions, pretraining a neural-network quantum state near the transition point yields a set of features that can be fine-tuned to obtain accurate descriptions of the phase diagram. In addition, the analysis of the feature space facilitated the extraction of valuable insights into the structure of the wave function. Furthermore, the fine-tuning process proves to be computationally more efficient, in terms of both time and memory, compared to traditional approaches in which the full training is required at each point of the phase diagram. We contend that, akin to the prevalent approach in machine learning [44], the employment of pretrained networks holds great potential for advancing the exploration of physical systems through NQS. This aligns with the findings of a recent study by Ref. [45], demonstrating that, in electronic-structure problems, a single wave function can be employed to investigate multiple compounds and geometries. A similar framework has also been proposed for decoding quantum error-correcting codes with generative modeling [46]. Exploring extensions of this approach to perform fine-tuning across different physical models stands as a crucial topic for future studies, as well as the development of techniques to automatically identify the most expressive pretraining point [47–50]. Another possible application could focus on approximating the real-time dynamics [51,52] by adjusting only the parameters of the shallow output network, thereby solving the problem within the feature space rather than the configuration space. It would be intriguing to further explore the precise physical meaning of the learned features, such as establishing a connection between the clusters in the feature space and the order parameters.

ACKNOWLEDGMENTS

We thank A. Laio, G. Carleo, F. Vicentini, and M. Imada for useful discussions. We acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support. The DMRG calculations have been performed within the ITensor library [53].

APPENDIX A: CHOOSING DIFFERENT PRETRAINING POINTS

The accuracy of the fine-tuning across various points on the phase diagram is influenced by the choice of the pretraining point. In our study we have always pretrained near transition points, where we expect better generalization properties as discussed in the main text. Here we investigate how the accuracy of the fine-tuned results varies when choosing different pretraining points, for example, within the bulk of one phase. In Fig. 5 we show the accuracy of the energy $\Delta\varepsilon$ relative to DMRG calculations for the J_1-J_2 Heisenberg model on a chain [see Eq. (4)] of $N = 100$ sites (assuming periodic boundary conditions). The transition point of the model in the thermodynamic limit is $(J_2/J_1)_c = 0.24116(7)$; however, on a finite system with $N = 100$ sites, the point exhibiting the maximum slope in the dimer order parameter occurs around $J_2/J_1 = 0.4$ (refer to the central panel of Fig. 2). The accuracy of the fine-tuned energies, using $J_2/J_1 = 0.4$ as the pretraining point, is approximately $\Delta\varepsilon \approx 10^{-3}$ within the interval $J_2/J_1 \in [0.1, 0.6]$ (green triangles in Fig. 5). Conversely, pretraining from $J_2/J_1 = 0.1$ (blue circles in Fig. 5) yields higher accuracy before $J_2/J_1 = 0.4$, but as the distance from the pretraining point increases the accuracy deteriorates

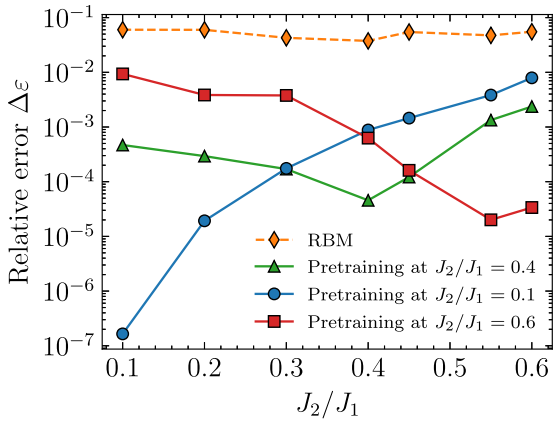

FIG. 5. Relative error $\Delta\varepsilon$ of the energy with respect to DMRG for the J_1-J_2 Heisenberg model on a chain [see Eq. (4)] of $N = 100$ sites. The curves are obtained performing the fine-tuning procedure starting from different pretraining points generated by a ViT with hyperparameters $h = 12$, $d = 192$, $n_l = 4$. Specifically, we set $J_2/J_1 = 0.4$ (green triangles), $J_2/J_1 = 0.1$ (blue circles), and $J_2/J_1 = 0.6$ (red squares). The accuracy of the same fully connected network (RBM) optimized on the physical configurations is also reported for comparison (orange diamonds).

to approximately $\Delta\varepsilon \approx 10^{-2}$. A similar behavior can be observed when choosing $J_2/J_1 = 0.6$ as the pretraining point (red squares in Fig. 5).

It is interesting to note that the accuracy of the network pretrained at $J_2/J_1 = 0.1$ (blue circles) deteriorates by four orders of magnitude when fine-tuning at $J_2/J_1 = 0.4$. In contrast, the network pretrained at $J_2/J_1 = 0.4$ (green triangles) loses less than one order of magnitude in accuracy when fine-tuning at $J_2/J_1 = 0.1$, with the error rising from $\Delta\epsilon \approx 10^{-4}$ to $\Delta\epsilon \approx 10^{-3}$. This result suggests that features learned near the phase transition are more robust for generalization compared to those learned within the bulk of a phase. Consequently, selecting a pretraining point that lies near the transition appears to strike the optimal balance, yielding an accuracy roughly consistent across all other points within the phase diagram. Nevertheless, it is worth noting that training the RBM in the hidden space consistently outperforms direct training on physical configurations, as illustrated by the orange diamonds in Fig. 5.

APPENDIX B: MEMORY EFFICIENCY IN FINE-TUNING AND PRETRAINING PROCESSES

The primary constraint in training neural networks with a large number of parameters arises from the restricted memory capacity of contemporary graphical processing units (GPUs), rather than their computational speed. Specifically, this limitation is associated to the back-propagation algorithm [1], which is crucial for evaluating the gradients of the network efficiently, but whose memory cost scales with the depth of the computation. Consider a deep neural network that takes an input vector $\boldsymbol{x}$ and produces a scalar output $f(\boldsymbol{x}, \theta) \in \mathbb{R}$, where θ is a vector of trainable parameters. For simplicity, we arrange these parameters as $\theta = \text{Concat}(\theta_0, \ldots, \theta_{n_l})$, where θ_l is a vector containing all the P_l parameters of the lth layer, and P is the total number of parameters across all layers, i.e., $P = \sum_{l=1}^{n_l} P_l$. Additionally, assume that, when computing the output, the network generates K intermediate activations a_k, each of size A_k, and A is the overall number of activations calculated as $A = \sum_{k=1}^{K} A_k$. For a batch of M distinct input vectors, the loss function can be defined as $\mathcal{L}(\theta) = (1/M) \sum_{i=1}^{M} L[f(\boldsymbol{x}_i, \theta)]$. To efficiently backpropagate the gradients of the loss with respect to the parameters, it is necessary to store all the A activations. Thus, the total memory cost of the algorithm scales with the depth of the computations and is expressed as $M \times (A + \max_l P_l)$ (neglecting the cost of storing all P weights). On the other hand, for the forward pass the memory cost is independent of the computation depth and is equal to $M \times (\max_k A_k + \max_l P_l)$. Further details can be found in Ref. [54]. Notice that during the fine-tuning process the memory-intensive backward pass over the deep network becomes unnecessary. In the context of this paper, for the used ViT architectures, the memory needed during the fine-tuning stage is approximately ten times less than what is required during the pretraining stage. The backpropagation of gradients constitutes the primary memory bottleneck, even when employing the Stochastic Reconfiguration optimization method [20]. This method requires the allocation of a matrix containing $4M^2$ real numbers, where M denotes the number of samples used in optimization. With double precision, this memory requirement translates to $32M^2/10^9$ GB. In our optimizations, with $M = 3000$, the memory usage is approximately 0.3 GB. This is two orders of magnitude smaller than the memory required for the backward pass during pretraining.

APPENDIX C: CONNECTING THE FEATURES TO ORDER PARAMETERS

In this Appendix we examine the Ising model in a transverse field [see Eq. (3)] aiming to establish a connection between the features learned by the ViT optimized at $g/\Gamma = 1$ and the magnetization order parameter that controls the phase transition. To achieve this, we consider a fixed batch of physical spin configurations $\{\boldsymbol{\sigma}_i\} \sim |\Psi(\boldsymbol{\sigma}; \theta_p)|^2$. We then compute the corresponding hidden representation and perform principal component analysis (PCA) on it. In Fig. 6 we plot the principal component against the local magnetizations of the spin configurations, i.e., $\sum_{i=1}^{N} \sigma_i$. The two quantities exhibit a strong correlation. This organization of configurations in the feature space simplifies the description of the physics of the system, allowing for an easy transition from the ordered phase to the disordered phase. It would be interesting to extend this analysis to other systems, where the order parameter is associated to an off-diagonal operator in the computational basis.

APPENDIX D: COMPARISON WITH MARSHALL SIGN PRIOR

A relevant question is trying to understand which kind of prior information is encoded in the features generated by the pretrained network. We focus on the J_1-J_2 Heisenberg model on a chain [see Eq. (4)], with $N = 100$ sites, fixing the value of the frustration ratio to $J_2/J_1 = 0.6$. We consider

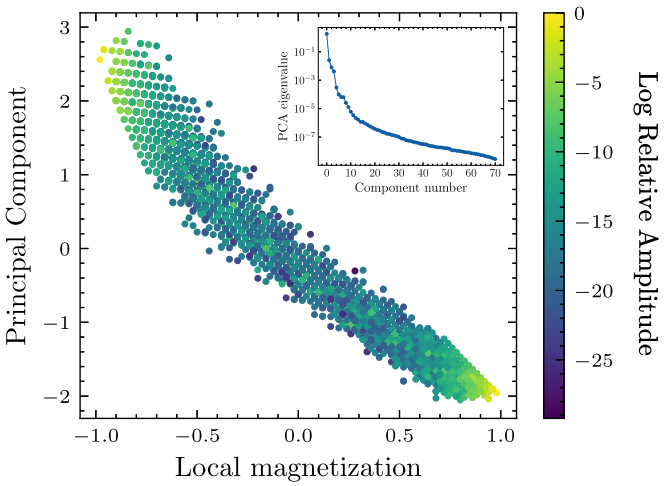

FIG. 6. Correlation between the local magnetization $\sum_{i=1}^{N} \sigma_i$ and the principal component of the hidden representations of the configurations associated to the ViT used to obtain the results for the Ising model in transverse field (see "Results" section). In the inset the PCA spectrum is shown.

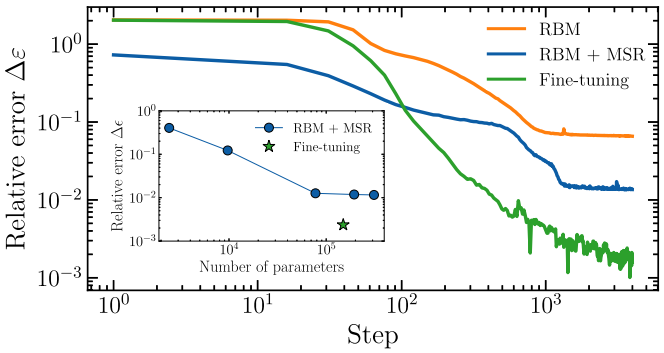

FIG. 7. Relative error in energy $\Delta\varepsilon$, compared to DMRG, plotted as a function of the optimization steps for the J_1-J_2 Heisenberg model [refer to Eq. (4)] with $J_2/J_1 = 0.6$ on a system of $N = 100$ sites. The orange curve represents the variational energy obtained using a RBM with $K = 384$ hidden neurons and 77 568 parameters. The blue curve depicts the same network with the addition of the Marshall Sign Rule as a prior for the sign structure. In contrast, the green curve is obtained by optimizing the same network on top of the hidden representation $\boldsymbol{z}$ generated by the Transformer with hyperparameters $h = 12$, $d = 192$, $n_l = 4$ at $J_2/J_1 = 0.4$. Inset: Relative error in energy $\Delta\varepsilon$ of a RBM trained with the MSR prior as a function of the number of parameters. For comparison, the accuracy of the fine-tuned network is also shown.

a Restricted Boltzmann Machine (RBM) with K hidden neurons [see Eq. (2)]. This network is employed in two distinct manners: trained directly on the physical configurations $\boldsymbol{\sigma}$, and trained on the hidden representations $\boldsymbol{z}_p$, which are generated by a pretrained ViT at $J_2/J_1 = 0.4$. As depicted in Fig. 7, using the hidden representations (green curve) achieves an accuracy of $\Delta\varepsilon \approx 10^{-3}$, which is two orders of magnitude higher compared to the same network defined directly on the physical configurations ($\Delta\varepsilon \approx 10^{-1}$, orange curve). The difference primarily arises from the physical properties of the system that are encoded in the hidden representations, such as sign structure, amplitudes, and symmetries. Furthermore, given that the sign structure at $J_2/J_1 = 0.4$ is well approximated by the MSR, we optimize an RBM, directly on the physical configurations, but implementing the Marshall sign prior (blue curve). This RBM achieves an accuracy of $\Delta\varepsilon \approx 10^{-2}$, underscoring that the information compressed in the hidden representation exceeds that provided by the Marshall sign prior. Despite increasing the number of parameters in RBMs, their performance remains inferior to the fine-tuned network due to the poor scaling behavior of the relative error in energy with the growth of network parameters and complicated structure of the landscape with a lot of local minima emerging when increasing the number hidden neurons (refer to the inset of Fig. 7).

[1] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, arXiv:1206.5538 [cs.LG].

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, OpenAI blog **1**, 9 (2019).

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse *et al.*, Language models are few-shot learners, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, 2020), pp. 1877–1901.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in *International Conference on Learning Representations* (ICLR, 2021).

[7] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, Highly accurate protein structure prediction with alphafold, Nature (London) **596**, 583 (2021).

[8] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[9] D. Luo and B. K. Clark, Backflow transformations via neural networks for quantum many-body wave functions, Phys. Rev. Lett. **122**, 226401 (2019).

[10] Y. Nomura and M. Imada, Dirac-type nodal spin liquid revealed by refined quantum many-body solver using neural-network wave function, correlation ratio, and level spectroscopy, Phys. Rev. X **11**, 031034 (2021).

[11] J. Robledo Moreno, G. Carleo, A. Georges, and J. Stokes, Fermionic wave functions from neural-network constrained hidden states, Proc. Natl. Acad. Sci. USA **119**, e2122059119 (2022).

[12] C. Roth, A. Szabó, and A. H. MacDonald, High-accuracy variational Monte Carlo for frustrated magnets with deep neural networks, Phys. Rev. B **108**, 054410 (2023).

[13] H. Lange, A. V. de Walle, A. Abedinnia, and A. Bohrdt, From architectures to applications: A review of neural quantum states, arXiv:2402.09402 [cond-mat.dis-nn].

[14] H. Lange, F. Döschl, J. Carrasquilla, and A. Bohrdt, Neural network approach to quasiparticle dispersions in doped antiferromagnets, Commun. Phys. **7**, 187 (2024).

[15] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, *Ab initio* solution of the many-electron Schrödinger equation with deep neural networks, Phys. Rev. Res. **2**, 033429 (2020).

[16] J. Kim, G. Pescia, B. Fore, J. Nys, G. Carleo, S. Gandolfi, M. Hjorth-Jensen, and A. Lovato, Neural-network quantum states for ultra-cold Fermi gases, arXiv:2305.08831 [cond-mat.quant-gas].

[17] G. Pescia, J. Han, A. Lovato, J. Lu, and G. Carleo, Neural-network quantum states for periodic systems in continuous space, Phys. Rev. Res. **4**, 023138 (2022).

[18] F. Becca and S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, 2017).

[19] L. L. Viteritti, R. Rende, A. Parola, S. Goldt, and F. Becca, Transformer wave function for the Shastry-Sutherland model: Emergence of a spin-liquid phase, arXiv:2311.16889 [cond-mat.str-el].

[20] R. Rende, L. L. Viteritti, L. Bardone, F. Becca, and S. Goldt, A simple linear algebra identity to optimize large-scale neural network quantum states, Commun. Phys. **7**, 260 (2024).

[21] Y. Tang, J. Liu, J. Zhang, and P. Zhang, Learning nonequilibrium statistical mechanics and dynamical phase transitions, Nat. Commun. **15**, 1117 (2024).

[22] R. G. Melko and J. Carrasquilla, Language models for quantum simulation, Nat. Comput. Sci. **4**, 11 (2024).

[23] K. Sprague and S. Czischek, Variational Monte Carlo with large patched transformers, Commun. Phys. **7**, 90 (2024).

[24] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, Phys. Rev. Lett. **128**, 090501 (2022).

[25] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. M. Hur, and B. K. Clark, Gauge-invariant and anyonic-symmetric autoregressive neural network for quantum lattice models, Phys. Rev. Res. **5**, 013216 (2023).

[26] L. L. Viteritti, R. Rende, and F. Becca, Transformer variational wave functions for frustrated quantum spin systems, Phys. Rev. Lett. **130**, 236401 (2023).

[27] R. Rende, F. Gerace, A. Laio, and S. Goldt, Mapping of attention mechanisms to a generalized Potts model, Phys. Rev. Res. **6**, 023057 (2024).

[28] S. Sorella, Wave function optimization in the variational Monte Carlo method, Phys. Rev. B **71**, 241103(R) (2005).

[29] A. Chen and M. Heyl, Empowering deep neural quantum states through efficient optimization, Nat. Phys. **20**, 1476 (2024).

[30] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, NetKet 3: Machine learning toolbox for many-body quantum systems, SciPost Phys. Codebases 7 (2022).

[31] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett. **69**, 2863 (1992).

[32] UMAP [55] is a general purpose dimension reduction technique for machine learning. It is constructed from a theoretical framework based in Riemannian geometry and algebraic topology; see Ref. [55] for more details.

[33] S. Eggert, Numerical evidence for multiplicative logarithmic corrections from marginal operators, Phys. Rev. B **54**, R9612 (1996).

[34] C. Lacroix, P. Mendels, and F. Mila, *Introduction to Frustrated Magnetism: Materials, Experiments, Theory* (Springer, Berlin, Heidelberg, 2011).

[35] L. Capriotti, F. Becca, A. Parola, and S. Sorella, Suppression of dimer correlations in the two-dimensional $J_1 - J_2$ Heisenberg model: An exact diagonalization study, Phys. Rev. B **67**, 212402 (2003).

[36] W. Marshall, Antiferromagnetism, Proc. R. Soc. London A **232**, 48 (1955).

[37] L. L. Viteritti, F. Ferrari, and F. Becca, Accuracy of restricted Boltzmann machines for the one-dimensional $J_1 - J_2$ Heisenberg model, SciPost Phys. **12**, 166 (2022).

[38] M. Calandra Buonaura and S. Sorella, Numerical study of the two-dimensional Heisenberg model using a green function Monte Carlo technique with a fixed number of walkers, Phys. Rev. B **57**, 11446 (1998).

[39] A. W. Sandvik, Finite-size scaling of the ground-state parameters of the two-dimensional Heisenberg model, Phys. Rev. B **56**, 11678 (1997).

[40] W.-J. Hu, F. Becca, A. Parola, and S. Sorella, Direct evidence for a gapless Z_2 spin liquid by frustrating Néel antiferromagnetism, Phys. Rev. B **88**, 060402(R) (2013).

[41] F. Ferrari and F. Becca, Gapless spin liquid and valence-bond solid in the J_1-J_2 Heisenberg model on the square lattice: Insights from singlet and triplet excitations, Phys. Rev. B **102**, 014417 (2020).

[42] L. Wang, Y. Zhang, and A. W. Sandvik, Quantum spin liquid phase in the Shastry-Sutherland model detected by an improved level spectroscopic method, Chin. Phys. Lett. **39**, 077502 (2022).

[43] S.-S. Gong, W. Zhu, D. N. Sheng, O. I. Motrunich, and M. P. A. Fisher, Plaquette ordered phase and quantum phase diagram in the spin-$\frac{1}{2}J_1-J_2$ square Heisenberg model, Phys. Rev. Lett. **113**, 027201 (2014).

[44] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame *et al.*, Huggingface's transformers: State-of-the-art natural language processing, arXiv:1910.03771 [cs.CL].

[45] M. Scherbela, L. Gerard, and P. Grohs, Towards a transferable fermionic neural wavefunction for molecules, Nat. Commun. **15**, 120 (2024).

[46] H. Cao, F. Pan, Y. Wang, and P. Zhang, qecGPT: Decoding quantum error-correcting codes with generative pre-trained transformers, arXiv:2307.09025 [quant-ph].

[47] T. Mendes-Santos, X. Turkeshi, M. Dalmonte, and A. Rodriguez, Unsupervised learning universal critical behavior

via the intrinsic dimension, Phys. Rev. X **11**, 011040 (2021).

[48] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[49] R. Zen, L. My, R. Tan, F. Hebert, M. Gattobigio, C. Miniatura, D. Poletti, and S. Bressan, Finding quantum critical points with neural-network quantum states, arXiv:2002.02618 [physics.comp-ph].

[50] L. Lewis, H.-Y. Huang, V. T. Tran, S. Lehner, R. Kueng, and J. Preskill, Improved machine learning algorithm for predicting ground state properties, Nat. Commun. **15**, 895 (2024).

[51] M. Schmitt and M. Heyl, Quantum many-body dynamics in two dimensions with artificial neural networks, Phys. Rev. Lett. **125**, 100503 (2020).

[52] T. Mendes-Santos, M. Schmitt, and M. Heyl, Highly resolved spectral functions of two-dimensional systems with neural quantum states, Phys. Rev. Lett. **131**, 046501 (2023).

[53] M. Fishman, S. White, and M. Stoudenmire, The ITensor software library for tensor network calculations, SciPost Phys. Codebases 4 (2022).

[54] R. Novak, J. Sohl-Dickstein, and S. S. Schoenholz, Fast finite width neural tangent kernel, in *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 162 (PMLR, 2022), pp. 17018–17044.

[55] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, arXiv:1802.03426 [stat.ML].