UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXVII CICLO DEL DOTTORATO DI RICERCA IN FISICA

Neural Network approaches to frustrated quantum spin models

Settore scientifico-disciplinare: **PHYS/04**

DOTTORANDO
Luciano Loris Viteritti Luciano Loris Viteritti

COORDINATORE
Prof. Francesco Longo Francesco Longo

SUPERVISORE DI TESI
Prof. Federico Becca Federico Becca

ANNO ACCADEMICO 2023/2024

A Riccardo e Roberta,
i migliori compagni di viaggio

Contents

Introduction

Quantum many-body systems have captured the attention of researchers for more than five decades. Nevertheless, even today, a significant portion of the proposed models remains unsolved. On one hand, analytical treatments often require approximations that are not fully justified in the strongly interacting regime. On the other hand, exact numerical methods are limited to small clusters due to the exponential growth of the Hilbert space with the number of particles. As a result, these small system sizes are typically insufficient for capturing the physical properties in the thermodynamic limit.

Numerical techniques based on the variational principle, however, offer a viable alternative tool to assess the low-energy properties of these systems, even beyond the perturbative regimes. While these methods overcome the limitations imposed by the exponentially large Hilbert space, the key challenge lies in finding a compact representation of the ground state encoding the correct physical properties of strongly-interacting systems. Despite this, variational approaches for studying quantum many-body systems have proved fundamental for understanding the properties of extremely complicated physical systems, famous examples being the Bardeen-Cooper-Schrieffer state [1] and Laughlin [2] wave functions to explain superconductivity and fractional quantum Hall effect, respectively.

A particularly interesting class of quantum systems is represented by frustrated spin models, characterized by the competition among various types of interactions. These models describe the behavior of strongly interacting fermions on a lattice, and understanding their properties could potentially unlock a theoretical explanation for high-temperature superconductivity [3]. Furthermore, due to the presence of competing interactions, these systems can give rise to exotic non-magnetic phases at extremely low temperatures in two or three spatial dimensions. One of the most iconic examples is the emergence of quantum spin liquids, which represent a unique state of matter distinguished by distinctive properties such as the absence of broken symmetry, a high degree of entanglement and fractional excitations [4]. From a numerical perspective, one difficulty in approaching frustrated spin models is related to the sign structure of the ground state, which is, in general, highly nontrivial, implying the necessity of a full optimization of the variational state that involves both moduli and signs. Consequently Quantum Monte Carlo methods cannot be applied to obtain exact properties. For this reason, in the last thirty years, alternative approaches have been developed. Density-matrix renormalization group (DMRG) [5] is free from sign problems, but, while it gives excellent results for a variety of one-dimensional models, its performance considerably worsens when dealing with two-dimensional systems. In this regard, the extensions based on Tensor Networks (e.g., Projected-Entangled Pair States) [6] represent a promising avenue to reach accurate results in more than one dimension, even in the thermodynamic limit.

Recently, a new class of wave functions based on neural networks, known as Neural-Network Quantum States (NQS), has been introduced and developed [7–9]. Starting from Restricted Boltzmann Machines (RBMs) [7], which are the simplest neural-network *Ansatz* (namely only one fully-connected hidden layer), numerous studies have been carried out testing different types of architectures; examples include Convolutional-Neural Networks (CNNs) [10–13], Recurrent-Neural Networks (RNNs) [14, 15], but also combinations of neural networks with standard variational wave functions (e.g., Gutzwiller-projected fermionic ones) [16, 17]. In the last few years, Transformers [18] have emerged as one of the most powerful deep learning tools [19–21]. The success of Transformers lies in their remarkable flexibility: with minimal modifications, they excel in addressing diverse problem domains, often outperforming specialized approaches [22–24]. The Transformer architecture has been also employed in the approximation of ground state, achieving highly accurate results across different systems [25–33]. The original work by Carleo and Troyer [7] was limited to Heisenberg models in one and two dimensions, where the sign structure of the ground state was known by the Marshall-sign rule [34]. This fact largely facilitates the numerical treatment, giving rise to an impressive accuracy of the neural-network states. More complicated models, such as the frustrated Heisenberg model, are more difficult to deal with. At present, frustrated quantum spin models remain extraordinarily challenging problems to be addressed by numerical techniques. From one side, DMRG calculations have reached remarkable accuracies on a cylindrical geometry (with large circumferences), thus approaching the two-dimensional limit [35]; from the other side, NQS, which can represent quantum states in arbitrary dimensions, have progressively demonstrated highly accurate descriptions of two-dimensional frustrated spin models. These results are competitive with, and often surpass, those obtained from standard methods such as Gutzwiller-projected states and Tensor Networks [27, 28, 36, 37]. However, further improvements should be pursued, in order to be able to perform calculations on large clusters and assess the real nature of the exact ground state within the highly-frustrated regime, where gapped or gapless spin liquids may exist.

Outlook

The focus of my PhD research was the investigation of exotic phases of matter that arise in frustrated spin models, primarily through numerical simulations. A key aspect of this work was the development of innovative approaches to obtain compact variational representations of the ground states of these systems, using parametrizations based on artificial neural networks. Specifically, the thesis presents numerical studies of frustrated quantum spin models on both one- and two-dimensional lattices, employing variational methods grounded on Neural-Network Quantum States.

While, from a theoretical standpoint, the accuracy of NQS can be systematically enhanced by increasing the number of parameters to efficiently represent highly-entangled quantum states, achieving this in practice requires the careful design of suitable architectures and optimization methods. The present research focused on developing and refining these techniques to tackle models of increasing complexity that cannot be effectively addressed by other numerical methods, improving and adapting them to address the specific challenges posed by each problem.

The thesis is organized as follows:

- In Chapter 1, we introduce the quantum many-body problem, providing an overview of its significance and challenges. We then outline the stochastic methods employed in this thesis to address this task. Specifically we focus on the Variational Monte Carlo, a general framework for the approximation of the ground state of a generic quantum many-body system.

- In Chapter 2, we explore gradient-based methods for the optimization of variational wave functions. In particular, we focus on the Stochastic Reconfiguration technique and we detail how to modify it for optimizing quantum states with a large number of parameters. This latter discussion includes original contributions derived from our work in Ref. [27].

- In Chapter 3, we present a class of variational wave function known as Neural-Network Quantum States. We compare the performance of both basic and advanced neural-network architectures on the one-dimensional J_1-J_2 Heisenberg model. The numerical results and the variational states discussed in this Chapter are adapted from our works in Ref. [38] and Ref. [31].

- In Chapter 4, we introduce a general framework based on Representation Learning, adapted from Ref. [28], to define accurate neural-network wave functions. We benchmark this approach on the J_1-J_2 Heisenberg model on the square lattice, with the numerical results taken from Ref. [27]. Additionally, we discuss the fine-tuning properties of neural networks elaborating the calculations from our work in Ref. [39].

- In Chapter 5, we present numerical calculations on the two-dimensional Shastry-Sutherland model. Our results, adapted from Ref. [28], reveal the existence of a small, but finite, region in the phase diagram which is consistent with a gapless spin liquid state.

At the end, in the final chapter Conclusions and Future Directions, we summarize the results of the thesis, drawing our conclusions and discussing some possible directions for future research.

List of Publications

Most of the results presented in this thesis are adapted from the following publications and preprints:

- L. L. Viteritti, F. Ferrari, F. Becca, *Accuracy of Restricted Boltzmann Machines for the one-dimensional J_1-J_2 Heisenberg model*, SciPost Phys. 12, 166 (2022)
- L. L. Viteritti, R. Rende, F. Becca, *Transformer variational wave functions for frustrated quantum spin systems*, Phys. Rev. Lett. 130, 236401 (2023)
- R. Rende, L. L. Viteritti, L. Bardone, F. Becca, S. Goldt, *A simple linear algebra identity to optimize Large-Scale Neural Network Quantum States*, Communications Physics 7, 260 (2024)
- L. L. Viteritti, R. Rende, A. Parola, S. Goldt, F. Becca, *Transformer Wave Function for the Shastry-Sutherland Model: emergence of a Spin-Liquid Phase*, arXiv:2311.16889 (2023)
- R. Rende, S. Goldt, F. Becca, L. L. Viteritti, *Fine-tuning Neural Network Quantum States*, arXiv:2403.07795 (2024)
- R. Rende, L. L. Viteritti, *Are queries and keys always relevant? A case study on Transformer wave functions*, arXiv:2405.18874 (2024)

My PhD work led also to the following preprint, whose results have not been reported in this thesis:

- G. Crognaletti, G. Di Bartolomeo, M. Vischi, L. L. Viteritti, *Equivariant Variational Quantum Eigensolver to detect Phase Transitions through Energy Level Crossings*, arXiv:2403.07100 (2024)

Chapter 1

The Quantum Many-Body problem

The objective of this Chapter is to provide a brief introduction to the many-body problem, highlighting the significant computational challenges, particularly related to the exponential growth of the Hilbert space, and the strategies devised to address them. Special attention will be given to the formulation and implementation of methods to approximate the ground state of generic quantum many-body spin systems. Additionally, the Chapter explores the *Variational Monte Carlo framework*, with a discussion on the choice of appropriate variational wave functions and the importance of efficient sampling techniques.

1.1 The origins

Technological advancements in the 20th century revealed fascinating physical phenomena such as superconductivity [1], superfluidity [40], and the fractional quantum Hall effect [2, 41]. These phenomena could not be adequately explained by mean-field theories and necessitated fully incorporating the electron-electron interactions, motivating significant interest in the study of quantum many-body systems.

Specifically, the discovery of high-temperature superconductivity [42, 43] marked a paradigm shift in the investigation of strongly correlated systems. Following Anderson's seminal contribution [3], the focus moved from studying increasingly complicated systems via *ab-initio* approaches, such as those based on Density Functional Theory [44, 45], to adopting a reductionist perspective. The latter approach aimed the construction of simplified lattice models designed to capture the essential physics of strongly correlated materials, disregarding from specific microscopic details. Notable examples include the Hubbard [46] and the Heisenberg [47] Hamiltonians, which respectively represent minimal descriptions of interacting electrons and spins on a lattice. These models have proven essential in exploring exotic phases of matter, such as quantum spin liquids [37, 48–50] and high-temperature superconductivity [51, 52].

However, despite their conceptual simplicity, determining the phase diagrams of these models remains an open and formidable challenge [53]. The explicit treatment of electron-electron interactions within these models leads to an exponential scaling of computational complexity, rendering the exact solution of the many-body problem intractable. Consequently, the development of approximate techniques continue to be a crucial area of research to understand the rich physics of strongly correlated systems.

1.2 Exponential complexity?

Given an Hamiltonian $\hat{H}$ describing a generic quantum model defined on a lattice, a complete characterization of its physical properties is obtained by solving the time-independent Schrödinger equation:

$$\hat{H}\left|\Psi_n\right\rangle = E_n\left|\Psi_n\right\rangle \ . \tag{1.1}$$

From a numerical perspective, the eigenvalues E_n and the corresponding eigenstates $\left|\Psi_n\right\rangle$ can be computed by diagonalizing the Hamiltonian $\hat{H}$ in a certain basis of the Hilbert space. To be concrete we focus on a system of N spin $S = 1/2$ on a generic lattice; in this case the most simple choice is the *computational basis*, namely $\left|\sigma\right\rangle = \left|\sigma_1^z, \sigma_2^z, \dots, \sigma_N^z\right\rangle$, with $\sigma_i^z = \pm 1$. Inserting a completeness in the form $\hat{\mathbb{1}} = \sum_{\sigma'}\left|\sigma'\right\rangle\left\langle\sigma'\right|$ and projecting on $\left|\sigma\right\rangle$, Eq. (1.1) becomes:

$$\sum_{\sigma'} H_{\sigma,\sigma'}\Psi_n(\sigma') = E_n\Psi_n(\sigma) \ , \tag{1.2}$$

where $H_{\sigma,\sigma'} = \langle\sigma|\hat{H}|\sigma'\rangle$ are the matrix elements of the Hamiltonian in the chosen basis and $\Psi_n(\sigma)$ are the coefficients of the quantum state when expanded in the computational basis, namely $\left|\Psi_n\right\rangle = \sum_\sigma \Psi_n(\sigma)\left|\sigma\right\rangle$. The complete solution of Eq. (1.2) requires in general the diagonalization of a $2^N \times 2^N$ hermitian matrix. However physical models are usually described in terms of *local Hamiltonians*, where a generic state $\left|\sigma\right\rangle$ has a number of *connected elements* $\left|\sigma'\right\rangle$, such that $\langle\sigma|\hat{H}|\sigma'\rangle \neq 0$, at most $\mathrm{Poly}(N)$. Therefore the Hamiltonian matrix $H_{\sigma,\sigma'}$ is sparse with at most $\mathrm{Poly}(N)\times 2^N$ non zero elements. Approaches in which the entire Hamiltonian matrix is stored in the computer are known as *Exact Diagonalization* (ED) methods [54]. An additional simplification is given by the fact that, typically, the physical properties of a quantum model can be extracted by knowing only the ground state and the low-energy excitations. In this view, iterative Lanczos-type approaches [54] allow us to obtain to the exact low-energy properties of the system without the need to allocate the entire Hamiltonian matrix. Although more efficient in terms of memory with respect to ED approaches, such methods still require the storage in memory of a certain number of vectors of Hilbert space dimension. Consequently, their effectiveness is typically limited to a number of sites $N < 40$ [54, 55].

The strategy to avoid the storage of the exponentially large full quantum state is grounded on the observation that physical states, for example low-energy states of local Hamiltonians or states produced by time evolution under local Hamiltonians, typically occupy a small portion of the full Hilbert space contrary to a generic random state (see left panel of Fig. 1.1) [56]. To clarify this concept, let us use an analogy with images. Suppose fixing a certain number of pixels in an image. By varying the intensity of each pixel, we can generate an exponentially vast number of possible images. However, if the intensity of each pixel is random, the result will be a chaotic pattern with no discernible meaning. In contrast, to produce an image with recognizable content, such as a person's face or an animal, there must be a structured pattern in the pixel intensities. Similarly, in classification and recognition tasks, we focus on a small, meaningful subset of all possible images that could be generated by modifying pixel intensities.

This parallels the idea of physical states in the Hilbert space: just as meaningful images represent a small subset of all possible pixel combinations, the physical states in a Hilbert space form a highly specific subset within the vast space of all possible states.

Starting from this observation, the alternative approach is to introduce a variational parametrization of the quantum state depending on a set of P variational parameters θ:

$$|\Psi_\theta\rangle = \sum_{\{\sigma\}} \Psi_\theta(\sigma) \, |\sigma\rangle \;\; , \tag{1.3}$$

where we have defined $\Psi_\theta(\sigma) = \langle\sigma|\Psi_\theta\rangle$. In this way instead of storing the amplitude of the wave function for each configuration of the basis (as required for ED or Lanczos methods) we store the vector θ and, in order to have an efficient parametrization, we require that $P \sim \mathrm{Poly}(N)$, so $P \ll 2^N$. In practice, we are performing a compression of the full quantum state, if this compression is efficient we should be able to span with this

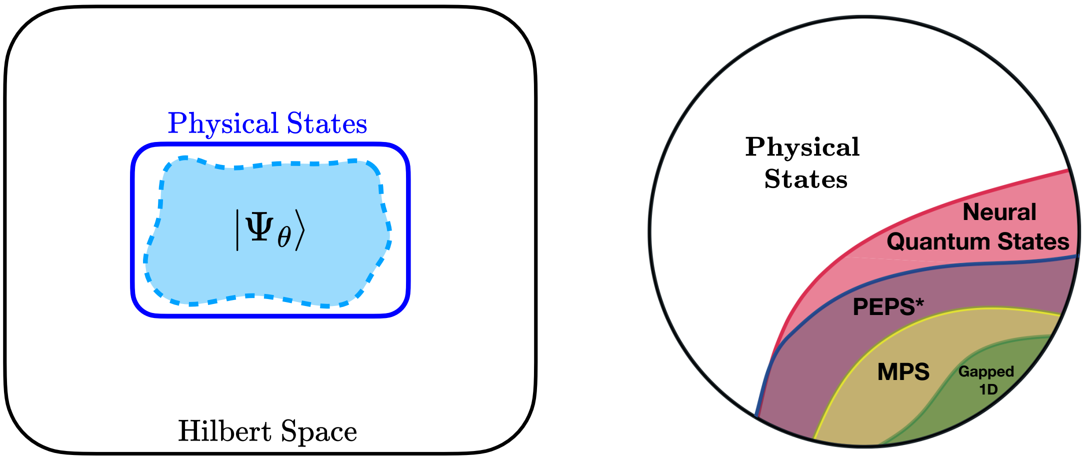

Figure 1.1: **Left Panel**: Illustration of the Hilbert space associated with a quantum many-body system. Physical states occupy a small, specific region within the whole space. An effective variational Ansatz $|\Psi_\theta\rangle$ should accurately capture the relevant portion of the Hilbert space where these physical states reside. **Right Panel**: Comparative representation of the expressive power of different classically tractable variational states to capture the physical states within the Hilbert space. Neural-Network Quantum States offer a more flexible parametrization of quantum states compared to tensor network-based methods like Matrix Product States (MPS) and Projected Entangled Pair States (PEPS). This image is adapted from Ref. [57].

parametrization not the full Hilbert space but the corner of the physical states (see left panel of Fig. 1.1).

In the last fifty years several variational parametrizations have been proposed to capture the low-energy properties of quantum Hamiltonians. The simplest examples are the *Mean-Field/Product States*

$$\Psi_\theta(\sigma) = \Phi_1(\sigma_1^z)\Phi_2(\sigma_2^z)\cdots\Phi_N(\sigma_N^z)\ . \tag{1.4}$$

where $\Phi_j(\sigma_j^z)$ is a scalar function which acts on a single spin variable σ_j^z. Typically this kind of parametrization gives the correct picture of the phase diagram but they are not able to capture exotic phases of matter, since they do not encode quantum correlations.

A powerful generalization of this kind of variational states are the *Matrix Product States* (MPS). In this case, the wave function is defined by the contraction of several tensors:

$$\Psi_\theta(\sigma) = \sum_{i_1,i_2,\dots,i_N} A^{(1)}_{i_1}(\sigma_1^z)A^{(2)}_{i_i,i_2}(\sigma_2^z)\cdots A^{(N)}_{i_N}(\sigma_N^z)\ , \tag{1.5}$$

where $A^{(j)}_{i,j}(\sigma_j^z)$ is a $\chi \times \chi \times d_{\text{local}}$ tensor. Here, d_{local} is the local dimension of the Hilbert space, so for spin 1/2 we have $d_{\text{local}} = 2$, and χ is called *bond dimension*, leading to a total number of $d_{\text{local}} N \chi^2$ variational parameters. Despite its simple structure, this kind of Ansatz is able to accurately describe the ground state of gapped one-dimensional Hamiltonians, where χ is the control parameter that regulates the accuracy of the variational approximations. In principle when χ is exponentially large in the system size it is possible to obtain an exact representation of the ground state, however in practice polynomial resources are sufficient to obtain numerically exact result for one dimensional gapped Hamiltonian [58, 59]. Although, this approach can accurately describe one-dimensional systems, where a large bond dimension can be easily used, in two dimensions, serious limitations appear, either imposing to work with a high-rank tensor structure, for instance *Projected Entangled Pair States* (PEPS) [60], or with quasi-one-dimensional cluster with low-rank tensors arranged in a snaked path [61]. Following S. White's pioneering work in 1992, where he introduced the *Density Matrix Renormalization Group* (DMRG) algorithm [5], numerous strategies have been developed for optimizing tensor-network-based states. However, this thesis will not delve into the specifics of these variational states or their optimization techniques. For a detailed discussion on these topics, see for example Refs. [58, 59]. Instead, our focus will be on using these approaches as a basis for comparison with other methods.

Few years ago, in a seminal work Carleo and Troyer [7] propose an innovative parametrization of quantum states based on artificial neural networks, introducing the so-called *Neural-Network Quantum States* (NQS)

$$\Psi_\theta(\sigma) = \exp\left[\mathcal{G}_\theta(\sigma_1^z, \dots, \sigma_N^z)\right] \ , \tag{1.6}$$

where $\mathcal{G}_\theta(\sigma_1^z, \dots, \sigma_N^z)$ is a neural network which has as input the physical spin configuration. A comprehensive description of this type of wave function is provided in Chapter 3. Here, we highlight that, in principle, this parametrization is highly flexible and does not face the limitations that arise with tensor network-based approaches as the dimensionality of the system increases. The primary advantage of these Ansätze lies in their ability to systematically improve accuracy by increasing the number of parameters, making them highly effective at representing complicated, highly-entangled quantum states. In the right panel of Fig. 1.1 we represent in a diagram the expressive power of classically tractable variational states. It shows that that there exist quantum states that are not efficiently expressible in terms of MPS or PEPS but that are instead efficiently expressible with NQS [57, 62].

1.3 Variational Principle

Up to this point, we have introduced the concept of using a variational state $|\Psi_\theta\rangle$ that is sufficiently expressive to capture the physically relevant states within Hilbert space. Let us suppose we are interested in approximating the ground state of a given Hamiltonian $\hat{H}$. In this case, we introduce the *variational energy* E_θ, defined as the expectation value of the Hamiltonian with respect to the variational state $|\Psi_\theta\rangle$, expressed as

$$E_\theta = \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} \ . \tag{1.7}$$

A crucial observation is that this quantity is bounded from below by the exact ground state energy E_0. To see this, we expand the variational state in the basis of the Hamiltonian's eigenstates, $|\Psi_\theta\rangle = \sum_n \langle\Psi_n|\Psi_\theta\rangle\,|\Psi_n\rangle$. Substituting this expansion into Eq. (1.7), we can rewrite the variational energy as follows:

$$E_\theta = \sum_n E_n \frac{|\langle\Psi_n|\Psi_\theta\rangle|^2}{\langle\Psi_\theta|\Psi_\theta\rangle} = E_0 + \sum_{n\neq 0}(E_n - E_0)\frac{|\langle\Psi_n|\Psi_\theta\rangle|^2}{\langle\Psi_\theta|\Psi_\theta\rangle} \geq E_0 \ . \tag{1.8}$$

The previous equation embodies the *Variational Principle*, which provides a controlled method for approximating the ground state wave function of a given Hamiltonian. Within this framework, a specific trial wave function $|\Psi_\theta\rangle$ is proposed, and its parameters are optimized to minimize the corresponding variational energy E_θ. This approach allows for the comparison of different trial states: the trial state that yields the lowest energy is then selected as the best approximation to the true ground state $|\Psi_0\rangle$.

1.4 Variational Monte Carlo Framework

The *variational principle* defines the loss function that must be minimized to find the best approximation of the ground state of a given Hamiltonian within the manifold of states defined by the chosen variational Ansatz. However, the principle itself does not specify how to obtain this optimal state in practice. To address this, we introduce a general framework known as *Variational Monte Carlo* (VMC), which not only enables us to determine the optimal state but also allows for the evaluation of its physical properties. Achieving this requires two key tasks:

i) Calculating the expectation values of operators with respect to the variational state;

ii) Updating the variational parameters in a manner that minimizes the variational energy.

In this Chapter, we will focus on the first task, while Chapter 2 will be dedicated to the optimization of the variational parameters.

1.4.1 Expectation values of quantum operators

In general, given a variational state $|\Psi_\theta\rangle$ and a quantum operator $\hat{A}$, the exact evaluation of its expectation value, $\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle/\langle\Psi_\theta|\Psi_\theta\rangle$, requires a summation over all elements of the Hilbert space basis $\{|\sigma\rangle\}$ which grows exponentially with system size:

$$\frac{\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \sum_\sigma \frac{\langle\Psi_\theta|\sigma\rangle\,\langle\sigma|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} \ . \tag{1.9}$$

The previous expression can be rewritten in a more convenient form by multiplying and dividing by $\langle\sigma|\Psi_\theta\rangle$:

$$\frac{\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \sum_\sigma \frac{|\,\langle\sigma|\Psi_\theta\rangle\,|^2}{\langle\Psi_\theta|\Psi_\theta\rangle}\frac{\langle\sigma|\hat{A}|\Psi_\theta\rangle}{\langle\sigma|\Psi_\theta\rangle} \ . \tag{1.10}$$

Notably, this procedure remains valid even when $\langle\sigma|\Psi_\theta\rangle = 0$, since we can decompose the sum in Eq. (1.9) as follows:

$$\frac{\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \sum_{\sigma\ :\ \langle\sigma|\Psi_\theta\rangle\neq 0} \frac{\langle\Psi_\theta|\sigma\rangle\,\langle\sigma|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} + \sum_{\sigma\ :\ \langle\sigma|\Psi_\theta\rangle = 0} \cancelto{0}{\frac{\langle\Psi_\theta|\sigma\rangle\,\langle\sigma|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}} \ . \tag{1.11}$$

Thus, without loss of generality, we can assume that Eq. (1.10) is restricted to configurations such that $\langle\sigma|\Psi_\theta\rangle \neq 0$.

At this stage, we define the Born probability distribution as follows:

$$P_\theta(\sigma) = \frac{|\,\langle\sigma|\Psi_\theta\rangle\,|^2}{\langle\Psi_\theta|\Psi_\theta\rangle} \ , \tag{1.12}$$

along with the local observable $A^L_\theta(\sigma)$ corresponding to the operator $\hat{A}$, expressed as:

$$A^L_\theta(\sigma) = \frac{\langle\sigma|\hat{A}|\Psi_\theta\rangle}{\langle\sigma|\Psi_\theta\rangle} = \sum_{\sigma'} \langle\sigma|\hat{A}|\sigma\rangle\,\frac{\Psi_\theta(\sigma')}{\Psi_\theta(\sigma)} \ . \tag{1.13}$$

Notably, solely the ratio between two amplitudes $\Psi_\theta(\sigma')/\Psi_\theta(\sigma)$ is relevant for the computation of the local observables, thereby permitting the use of unnormalized states (see also Sec. 1.4.3).

Using the previous definitions, we can rewrite the expectation value of a quantum operator as the expectation value of the local observable $A^L_\theta(\sigma)$ over the probability distribution $P_\theta(\sigma)$. Thus, Eq. (1.10) can be reformulated as:

$$\frac{\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \sum_\sigma P_\theta(\sigma) A^L_\theta(\sigma) \ . \tag{1.14}$$

This expression is formally exact, but more importantly, it allows to introduce a controlled approximation method for computing expectation values. Specifically, we can perform a *stochastic estimation* of the expectation values:

$$\frac{\langle\Psi_\theta|\hat{A}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} \approx \frac{1}{M}\sum_{i=1}^{M} A^L_\theta(\sigma_i) \ , \tag{1.15}$$

where $\{\sigma_1, \sigma_2, \ldots, \sigma_M\} \sim P_\theta(\sigma)$. To perform the stochastic estimation in Eq. (1.15) we need to generate configurations σ_i that are distributed according to the desired probability $P_\theta(\sigma)$ and computing the function $A^L_\theta(\sigma_i)$ for all these configurations. Regarding the latter point, the estimation of $A^L_\theta(\sigma_i)$ generally involves summations over an exponential number of terms in the system size [see Eq. (1.13)]. To perform the computation in polynomial time with to the system size, we restrict our focus to a specific class of quantum operators. Specifically, physically relevant observables are typically local, meaning they can be expressed as $\hat{A} = \sum_j \hat{A}_j$, where $\hat{A}_j$ acts on a small number of degrees of freedom. For instance, in the Heisenberg model, the Hamiltonian $\hat{H} = \sum_{\langle i,j\rangle} \hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_j$ consists of terms that act only on pairs of spins. Consequently, the sum in Eq. (1.13) is restricted to configurations σ' such that $\langle\sigma'|\hat{A}|\sigma\rangle \neq 0$. For local operators, the number of *connected configurations* σ' to a given configuration σ scales polynomially with the system size. Therefore, $A^L_\theta(\sigma)$ can be computed in polynomial time. Furthermore, it can be shown that for local operators, the variance of the estimator in Eq. (1.15) is *finite* [56, 63]. Consequently, the error in the expectation value estimate decreases as $O(1/\sqrt{M})$, allowing for arbitrary accuracy in the estimation of expectation values by increasing the number of samples.

1.4.2 Zero Variance property

In the special case where the operator $\hat{A}$ is the Hamiltonian $\hat{H}$ and the variational state $|\Psi_\theta\rangle$ coincides with an exact eigenstate of the Hamiltonian, such as $|\Psi_n\rangle$, the *local energy*

$$E_L(\sigma) = \frac{\langle\sigma|\hat{H}|\Psi_\theta\rangle}{\langle\sigma|\Psi_\theta\rangle} \ , \tag{1.16}$$

becomes independent of the configuration σ and equals the eigenvalue E_n. Consequently, in this scenario, the local energy $E_L(\sigma)$ in Eq. (1.16) exhibits *zero variance*. This property is significant within this framework, as a small variance in the local energy generally indicates that the trial state is close to an exact eigenstate of the Hamiltonian. Indeed, the variance of the Hamiltonian,

$$\mathrm{Var}(H) = \frac{\langle\Psi_\theta|\hat{H}^2|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} - \left(\frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\right)^2 \ , \tag{1.17}$$

is a non-negative quantity, $\mathrm{Var}(H) \geq 0$, providing a measure of the quality of the variational approximation [63].

1.4.3 Markov Chain Monte Carlo

In Section 1.4.1, we discussed how to compute expectation values of quantum operators in an approximate manner. This stochastic estimation is based on evaluating the local observables on a set of configurations $\{\sigma_1, \sigma_2, \ldots, \sigma_M\}$ that are distributed according to $P_\theta(\sigma)$. When it is possible to generate configurations with a specific probability distribution, this process is referred to as *direct sampling*. In this scenario, all configurations are independent of each other. Unfortunately, this is feasible only in a limited number of cases. In general, to perform direct sampling, one needs to know the normalization constant of the probability distribution, which is given by $\langle\Psi_\theta|\Psi_\theta\rangle = \sum_\sigma |\Psi(\sigma)|^2$. This normalization constant involves summing an exponential number of terms with respect to the system size, making it computationally infeasible for generic variational states as the system size increases. Generally, we are unable to directly sample from the desired probability distribution, necessitating the use of indirect methods to obtain such configurations. This is where the concept of *Markov Chain Monte Carlo* (MCMC) becomes relevant. Notably, the latter approach is highly versatile and can be applied to a wide variety of cases without needing to compute the normalization constant explicitly [63].

A Markov Chain is a sequence of configurations generated stochastically, where each configuration in the sequence is obtained from the previous one by making random changes. Specifically, the transition probability $T(\sigma'|\sigma)$ at each step depends solely on the current configuration. It can be shown that a sufficient condition for the sequence to have a unique stationary target distribution $P_\theta(\sigma)$ is that it satisfies the *Detailed Balance*:

$$P_\theta(\sigma)T(\sigma'|\sigma) = P_\theta(\sigma')T(\sigma|\sigma') \ . \tag{1.18}$$

This condition ensures that, regardless of the initial configuration of the chain, the sequence will eventually converge to the correct distribution $P_\theta(\sigma)$ in the long-time limit.

Choosing an appropriate transition rule $T(\sigma|\sigma')$ is a non-trivial task. One of the most well-known parameterizations is provided by the *Metropolis-Hastings algorithm*, where the *transition kernel* $T(\sigma'|\sigma)$ is decomposed into two local subprocesses that can be computed efficiently [56, 64, 65]:

$$T(\sigma'|\sigma) = k(\sigma'|\sigma)A(\sigma',\sigma) + \delta_{\sigma,\sigma'}\sum_{\sigma''} k(\sigma''|\sigma)[1 - A(\sigma'',\sigma)] \ , \tag{1.19}$$

Here, $k(\sigma'|\sigma)$ is the *proposal kernel*, which proposes the new configuration σ' given σ, while $A(\sigma',\sigma)$ is the *acceptance probability* for accepting the proposed state σ'. The second term[1], proportional to $\delta_{\sigma,\sigma'}$, in Eq. (1.19) accounts for the possibility that the proposed configuration σ' is identical to the current configuration σ [65–67]. Typically, the proposal kernel $k(\sigma'|\sigma)$ involves modifying only a few degrees of freedom, such as flipping a single spin in a given configuration. Special care must be taken when dealing with systems that exhibit symmetries. For example, in systems where the total magnetization along the direction of the computational basis is conserved, it may be necessary to impose a transition rule that preserves it. For instance, a valid proposed move in such a system could involve flipping two spins oriented in opposite directions to conserve the total magnetization. This ensures that the Markov chain remains within a specific subspace.

The simplest choice of acceptance probability that satisfies the Detailed Balance condition in Eq. (1.18) with the previously defined transition kernel [see Eq. (1.19)] is given by

$$A(\sigma',\sigma) = \min\left(1, \frac{P_\theta(\sigma')}{P_\theta(\sigma)}\frac{k(\sigma|\sigma')}{k(\sigma'|\sigma)}\right) \ . \tag{1.20}$$

[1] This contribution is essential for ensuring that the transition kernel is normalized, such that $\sum_{\sigma'} T(\sigma'|\sigma) = 1$ [65].

The crucial point here is that the normalization constant cancels out, meaning that only the ratio $P_\theta(\sigma')/P_\theta(\sigma) = |\Psi_\theta(\sigma')/\Psi_\theta(\sigma)|^2$ is relevant [see Eq. (1.12)]. This allows us to consider unnormalized variational states. In most cases, it is useful to consider symmetric conditional probabilities, where $k(\sigma|\sigma') = k(\sigma'|\sigma)$. In this scenario, the acceptance probability simplifies to

$$A(\sigma', \sigma) = \min\left(1, \frac{P_\theta(\sigma')}{P_\theta(\sigma)}\right) \ , \tag{1.21}$$

This case is simply known as the *Metropolis algorithm.* In our treatment of quantum many-body systems on lattice, we will focus on this case.

The advantage of working with non-normalized wave functions is that it provides considerable freedom in the choice of the parameterization. However, from a practical standpoint, we may encounter numerical issues such as underflow and overflow when evaluating the amplitude of the wave function. For this reason, instead of parameterizing the wave function directly, we typically parameterize its logarithm. Assuming in general that $\Psi_\theta(\sigma) \in \mathbb{C}$, we have:

$$\mathrm{Log}[\Psi_\theta(\sigma)] = \log[|\Psi_\theta(\sigma)|] + i\ \arg(\Psi_\theta(\sigma)) \ . \tag{1.22}$$

where $\mathrm{Log}(\cdot)$ denotes the complex logarithm and $\log(\cdot)$ denotes the real logarithm. In the following, we will show that the Metropolis algorithm can be implemented equivalently without ever exponentiating the wave function. The key is to rewrite the algorithm in logarithmic scale, leveraging the fact that the logarithm is a monotonic function of its argument.

Let us assume that σ is the current configuration of the Markov chain. To obtain the new configuration of the Markov Chain according to the Metropolis algorithm, we iterate through the following steps:

1. Generate a configuration σ' according to the proposal kernel $k(\sigma'|\sigma)$.

2. Evaluate the log-acceptance ratio of the proposed move using Eq. (1.21):

$$\log[A(\sigma', \sigma)] = \min\left(0, \log\left[\frac{P_\theta(\sigma')}{P_\theta(\sigma)}\right]\right) \ , \tag{1.23}$$

with

$$\log\left[\frac{P_\theta(\sigma')}{P_\theta(\sigma)}\right] = 2\Re\{\mathrm{Log}[\Psi_\theta(\sigma')]\} - 2\Re\{\mathrm{Log}[\Psi_\theta(\sigma)]\} \ , \tag{1.24}$$

where $P_\theta(\sigma)$ is the Born amplitude defined in Eq. (1.12).

3. Accept the new configuration σ' with probability $A(\sigma', \sigma)$. In practice, this is done by drawing a random number $u \in (0, 1]$ and proceeding as follows:
    - **Accept** the move if $\log(u) \leq \log[A(\sigma', \sigma)]$;
    - **Reject** the move if $\log(u) > \log[A(\sigma', \sigma)]$, in this case the new configuration in the Markov Chain remains σ.

This procedure allows us to generate a sample of configurations distributed according to $P_\theta(\sigma)$, relying solely on the logarithm of the wave function $\mathrm{Log}[\Psi_\theta(\sigma)]$.

It is important to note that the efficiency of the Metropolis algorithm in generating a sample of configurations depends on the ability to compute the amplitude of the wave function $\Psi_\theta(\sigma)$ (or equivalently its logarithm $\mathrm{Log}[\Psi_\theta(\sigma)]$) efficiently. Specifically, this computation must require a number of operations that is polynomial in the size of the system.

Chapter 2

Optimization of Large-Scale Variational Wave Functions

In Chapter 1, we introduced the VMC framework, with a particular focus on how stochastically estimate the expectation values of quantum operators with respect to a generic variational state using a sample of configurations generated via MCMC. It is important to note that this procedure is performed with fixed variational state parameters; we have not yet addressed how these parameters are adjusted. This addition will complete the VMC framework by defining an iterative procedure that, starting from a variational state with random parameters, optimizes the variational parameters and converges to an optimal state at the end of the training process. Specifically, in this Chapter we will discuss the *Stochastic Reconfiguration* (SR) method, a gradient-based approach for optimizing variational wave functions. Developed by Sandro Sorella in the late 1990s [68, 69], SR has become an important tool in quantum many-body physics. We will start by deriving the equations for updating the variational parameters. Then, we will show how these equations can be adapted to efficiently optimize large-scale variational wave functions [27, 36], allowing us to treat Ansatze with thousands to millions of variational parameters.

2.1 Gradient of the variational energy

Finding the ground state of a quantum system with the variational principle involves minimizing the variational energy $E_\theta = \langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle \, / \, \langle \Psi_\theta | \Psi_\theta \rangle$, where $|\Psi_\theta\rangle$ is a variational state parametrized through a set of P parameters θ (see Sec. 1.3).

In a gradient-based optimization approach, the fundamental ingredient is the evaluation of the gradient of the loss, which in this case is the variational energy E_θ, with respect to the parameters, namely $F_\alpha = -\partial E_\theta/\partial\theta_\alpha$, with $\alpha = 1, \dots, P$. The expectation value of the Hamiltonian $\hat{H}$ with respect to the variational state $|\Psi_\theta\rangle$ in the computational basis can be written as:

$$E_\theta = \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \sum_{\sigma,\sigma'} H_{\sigma\sigma'} \frac{\Psi_\theta^*(\sigma)\Psi_\theta(\sigma')}{\langle\Psi_\theta|\Psi_\theta\rangle} \ , \tag{2.1}$$

where we inserted two completeness of the form $\hat{\mathbb{1}} = \sum_\sigma |\sigma\rangle\langle\sigma|$, and we have defined the matrix elements $H_{\sigma\sigma'} = \langle\sigma|\hat{H}|\sigma'\rangle$. Now we can perform the derivative of the previous equation with respect to the parameter θ_α with $\alpha = 1, \dots, P$. Specifically, assuming real-valued parameters[2] we obtain:

$$\begin{aligned}\frac{\partial E_\theta}{\partial\theta_\alpha} = \sum_{\sigma,\sigma'} H_{\sigma\sigma'} \Bigg[& \frac{1}{\langle\Psi_\theta|\Psi_\theta\rangle} \left(\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma) + \Psi_\theta^*(\sigma)\partial_\alpha\Psi_\theta(\sigma')\right) - \\ & \frac{\Psi_\theta^*(\sigma)\Psi_\theta(\sigma')}{\langle\Psi_\theta|\Psi_\theta\rangle} \frac{1}{\langle\Psi_\theta|\Psi_\theta\rangle} \sum_{\sigma''} \left(\Psi_\theta(\sigma'')\partial_\alpha\Psi_\theta^*(\sigma'') + \Psi_\theta^*(\sigma'')\partial_\alpha\Psi_\theta(\sigma'')\right)\Bigg] \ ,\end{aligned} \tag{2.2}$$

where for brevity we have identified $\partial_\alpha = \partial/\partial\theta_\alpha$. The first term in the previous sum can be rewritten as

$$\begin{aligned}&\sum_{\sigma,\sigma'} H_{\sigma\sigma'}\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma) + \sum_{\sigma,\sigma'} H_{\sigma\sigma'}\Psi_\theta^*(\sigma)\partial_\alpha\Psi_\theta(\sigma') \\ =&\sum_{\sigma,\sigma'} \left[H_{\sigma\sigma'}\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma) + H_{\sigma\sigma'}^*\Psi_\theta^*(\sigma')\partial_\alpha\Psi_\theta(\sigma)\right]\end{aligned} \tag{2.3}$$

where we have exploit the fact that the Hamiltonian matrix $H_{\sigma\sigma'}$ is hermitian, namely $H_{\sigma\sigma'} = H_{\sigma'\sigma}^*$. Then, with the hypothesis of real-valued parameters:

$$\begin{aligned}H_{\sigma\sigma'}\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma) + H_{\sigma'\sigma}^*\Psi_\theta^*(\sigma')\partial_\alpha\Psi_\theta(\sigma) &= 2\Re\{H_{\sigma\sigma'}\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma)\} \ , \\ \Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma) + \Psi_\theta^*(\sigma)\partial_\alpha\Psi_\theta(\sigma') &= 2\Re\{\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma)\} \ .\end{aligned} \tag{2.4}$$

Combining the previous equations, the gradient in Eq. (2.2) becomes:

$$\frac{\partial E_\theta}{\partial\theta_\alpha} = \sum_{\sigma,\sigma'} \frac{2\Re\{H_{\sigma\sigma'}\Psi_\theta(\sigma')\partial_\alpha\Psi_\theta^*(\sigma)\}}{\langle\Psi_\theta|\Psi_\theta\rangle} - \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} \sum_{\sigma''} \frac{2\Re\{\Psi_\theta(\sigma'')\partial_\alpha\Psi_\theta^*(\sigma'')\}}{\langle\Psi_\theta|\Psi_\theta\rangle} \ . \tag{2.5}$$

[2] If the variational wave function is defined with complex-valued parameters, the latter can be treated as couples of independent real-valued parameters.

Noting that $\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle \in \mathbb{R}$ and $\langle\Psi_\theta|\Psi_\theta\rangle \in \mathbb{R}$, we obtain:

$$\frac{\partial E_\theta}{\partial\theta_\alpha} = 2\Re\left\{\frac{\langle\partial_\alpha\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} - \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\frac{\langle\partial_\alpha\Psi_\theta|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\right\} . \tag{2.6}$$

The expression in Eq. (2.6) requires further manipulations to derive a formula that involves only expectation values with respect to the variational state $|\Psi_\theta\rangle$. This reformulation is crucial as it allows an efficient estimation using Monte Carlo approaches (see Sec. 1.4.1).

Starting from Eq (2.6), we insert two completeness and then we multiple and divide for $\Psi^*_\theta(\sigma)$ and $\Psi^*_\theta(\sigma'')$ in the first and in the second term, respectively, getting the following expression:

$$\frac{\partial E_\theta}{\partial\theta_\alpha} = 2\Re\left\{\sum_{\sigma,\sigma'} H_{\sigma\sigma'}\frac{\partial_\alpha\Psi^*_\theta(\sigma)}{\Psi^*_\theta(\sigma)}\frac{\Psi_\theta(\sigma')\Psi^*_\theta(\sigma)}{\langle\Psi_\theta|\Psi_\theta\rangle} - \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\sum_{\sigma''}\frac{\partial_\alpha\Psi^*_\theta(\sigma'')}{\Psi^*_\theta(\sigma'')}\frac{|\Psi_\theta(\sigma'')|^2}{\langle\Psi_\theta|\Psi_\theta\rangle}\right\} . \tag{2.7}$$

At this point we introduce the operator $\hat{O}_\alpha$ for $\alpha = 1, \dots, P$ such that:

$$\begin{aligned} \langle\sigma|\hat{O}_\alpha|\sigma'\rangle &= O_\alpha(\sigma)\delta_{\sigma\sigma'} \ , \\ O_\alpha(\sigma) &= \frac{1}{\Psi_\theta(\sigma)}\frac{\partial\Psi_\theta(\sigma)}{\partial\theta_\alpha} = \frac{\partial \mathrm{Log}[\Psi_\theta(\sigma)]}{\partial\theta_\alpha} . \end{aligned} \tag{2.8}$$

In general, they depend upon the variational parameters θ, however, to keep the notation simple, we prefer not to put the label in these local operators. From their definition it is easy to show that $\langle\sigma|\hat{O}_\alpha|\Psi_\theta\rangle = \partial_\alpha\Psi_\theta(\sigma) \ \forall \ |\sigma\rangle$ and consequently $\hat{O}_\alpha|\Psi_\theta\rangle = |\partial_\alpha\Psi_\theta\rangle$.
At the end we obtain the *Gradient of the Variational Energy* [63]:

$$\boxed{F_\alpha = -\frac{\partial E_\theta}{\partial\theta_\alpha} = -2\Re\left\{\frac{\langle\Psi_\theta|\hat{O}^\dagger_\alpha\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} - \frac{\langle\Psi_\theta|\hat{H}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\frac{\langle\Psi_\theta|\hat{O}^\dagger_\alpha|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}\right\}} . \tag{2.9}$$

This final expression highlights the fact that the gradient of the energy can be recasted as a correlation function between the Hamiltonian $\hat{H}$ and a local operator $\hat{O}_\alpha$. Moreover, it is easy to show that it can be alternatively written as follows:

$$F_\alpha = -\frac{\partial E_\theta}{\partial\theta_\alpha} = -2\Re\left\{\langle\left(\hat{O}_\alpha - \langle\hat{O}_\alpha\rangle\right)^\dagger\left(\hat{H} - \langle\hat{H}\rangle\right)\rangle\right\} , \tag{2.10}$$

where $\langle\cdot\rangle = \langle\Psi_\theta|\cdot|\Psi_\theta\rangle \,/\, \langle\Psi_\theta|\Psi_\theta\rangle$. In order to evaluate efficiently Eq. (2.10) by employing Monte Carlo techniques, we insert a completeness relation:

$$F_\alpha = -2\Re\left\{\sum_\sigma\left[E_L(\sigma) - \langle\hat{H}\rangle\right]^*\left[O_\alpha(\sigma) - \langle\hat{O}_\alpha\rangle\right]\frac{|\Psi(\sigma)|^2}{\langle\Psi_\theta|\Psi_\theta\rangle}\right\} , \tag{2.11}$$

where $E_L(\sigma) = \langle\sigma|\hat{H}|\Psi_\theta\rangle \,/\, \langle\sigma|\Psi_\theta\rangle$ is the local energy [see Eq. (1.16)]. For a given sample of M configurations $\{\sigma_1, \sigma_2, \dots, \sigma_M\}$, sampled according to $P_\theta(\sigma)$ [see Eq. (1.12)], the stochastic estimate of F_α can be obtained as:

$$\bar{F}_\alpha = -2\Re\left\{\frac{1}{M}\sum_{i=1}^{M}\left[E_L(\sigma_i) - \bar{E}_L\right]^*\left[O_\alpha(\sigma_i) - \bar{O}_\alpha\right]\right\} \ , \tag{2.12}$$

where $\bar{E}_L = (1/M)\sum_{i=1}^{M} E_L(\sigma_i)$ and $\bar{O}_\alpha = (1/M)\sum_{i=1}^{M} O_\alpha(\sigma_i)$ denote sample means.

Notice that if the variational state coincides with an eigenstate of the Hamiltonian $\hat{H}$, the local energy coincides with the corresponding exact eigenvalue, regardless the configuration σ_i. Then $\bar{F}_\alpha$ identically vanishes without statistical fluctuations and thus we recover the zero-variance property for energy derivatives [63].

2.2 Stochastic Reconfiguration

The most used methods for the optimization of loss functions with a large number of parameters rely on *stochastic gradient descent* (SGD), where the gradient of the loss function is estimated from a randomly selected subset of the data [see Eq. (2.12)]. Over the years, variations of traditional SGD, such as Adam [70] or AdamW [71], have proven highly effective, leading to more accurate results. In the late 1990s, Amari and collaborators [72, 73] suggested to use the knowledge of the geometric structure of the parameter space to adjust the gradient direction for non-convex landscapes, defining the concept of *Natural Gradients*. In the same years, Sorella [68, 69] proposed a similar method, now known as *Stochastic Reconfiguration* (SR), to enhance the optimization of variational functions in quantum many-body systems.

In general, starting from the variational state $|\Psi_\theta\rangle$, the exact ground state $|\Psi_0\rangle$ of the Hamiltonian $\hat{H}$ can be obtained performing the imaginary time evolution, namely:

$$|\Psi_0\rangle \propto \lim_{\beta\to+\infty} \mathrm{e}^{-\beta\hat{H}}\,|\Psi_\theta\rangle \ . \tag{2.13}$$

assuming that $\langle\Psi_\theta|\Psi_0\rangle \neq 0$. However, the exact application of the operator $\mathrm{e}^{-\beta\hat{H}}$ is infeasible for large system size, being it in general a non-sparse exponentially large matrix $d_{\text{local}}^N \times d_{\text{local}}^N$, where d_{local} is the dimension of the local Hilbert space (e.g., $d_{\text{local}} = 2$ for 1/2 spins and $d_{\text{local}} = 4$ for fermions) and N the number of particles in the system.

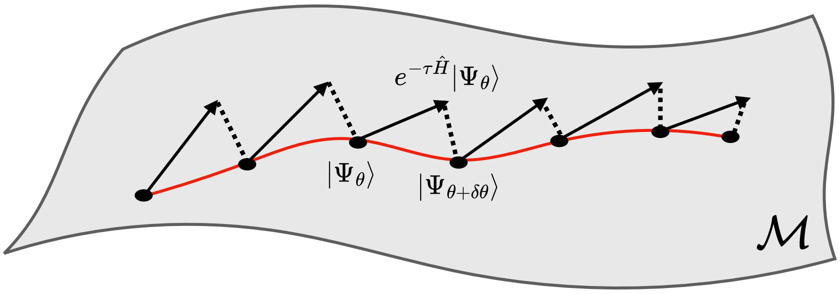
$e^{-\tau\hat{H}}|\Psi_\theta\rangle$
$|\Psi_\theta\rangle$
$|\Psi_{\theta+\delta\theta}\rangle$
$\mathcal{M}$

Figure 2.1: Graphical representation of the manifold $\mathcal{M}$, which contains the variational states $|\Psi_\theta\rangle$ as the parameters θ vary. Generally, the state $e^{-\tau\hat{H}}|\Psi_\theta\rangle$ lies outside $\mathcal{M}$, but a variational state $|\Psi_{\theta+\delta\theta}\rangle$ can be constructed by projecting $e^{\tau\hat{H}}|\Psi_\theta\rangle$ onto the manifold $\mathcal{M}$, yielding an approximation of the imaginary time-evolved state.

Formally, the operator $e^{-\beta\hat{H}}$ can be decomposed into a product of N_τ operators of the form $e^{-\tau\hat{H}}$, where $\tau = \beta/N_\tau$. Consequently, the ground state can be approximated through an iterative application of these operators on the initial state:

$$|\Psi_0\rangle \approx \underbrace{e^{-\tau\hat{H}}e^{-\tau\hat{H}}\cdots e^{-\tau\hat{H}}}_{N_\tau \text{ times}}|\Psi_\theta\rangle \ . \tag{2.14}$$

It is important to note that while $|\Psi_\theta\rangle$ belongs to a manifold $\mathcal{M}$, which represents a subset of states in the full Hilbert space parameterized by θ, the state $e^{-\tau\hat{H}}\,|\Psi_\theta\rangle$ generally does not lie within $\mathcal{M}$. This implies that there does not exist a set of parameters $\theta + \delta\theta$ such that $|\Psi_{\theta+\delta\theta}\rangle$ matches $e^{-\tau\hat{H}}\,|\Psi_\theta\rangle$ exactly. However, if the variational Ansatz is sufficiently expressive, we can approximate $e^{-\tau\hat{H}}\,|\Psi_\theta\rangle$ by changing the variational parameters. The central idea of SR is to determine the optimal update $\delta\theta$ that best approximates the imaginary time evolution for a duration τ (see Fig. 2.1). This process effectively projects the evolved state back onto the variational manifold $\mathcal{M}$, allowing us to maintain a tractable representation of the state which approximate the ground state. Consequently, the SR approach can be interpreted as an effective imaginary time evolution in the variational manifold.

From a mathematical perspective, our objective is to determine the optimal parameter update $\delta\theta$ that minimizes the distance between the imaginary time-evolved state and the updated variational state. This can be formulated as[3] [74]:

$$\delta\theta^* = \underset{\delta\theta}{\operatorname{argmin}}\ \mathcal{F}\left[\mathrm{e}^{-2\tau\hat{H}}\,|\Psi_\theta\rangle\,, |\Psi_{\theta+\delta\theta}\rangle\right] \ , \tag{2.15}$$

[3] The factor of 2 in the exponent of Eq. (2.15) is introduced to simplify the resulting expressions, eliminating numerical factors in the final results.

where $\mathcal{F}[\cdot,\cdot]$ denotes the *Fubini-Study distance*, a metric that quantifies the distance between two arbitrary states in the Hilbert space. The latter is defined as:

$$\mathcal{F}\left[|\Psi\rangle, |\Phi\rangle\right] = \arccos\sqrt{\frac{\langle\Psi|\Phi\rangle\langle\Phi|\Psi\rangle}{\langle\Psi|\Psi\rangle\langle\Phi|\Phi\rangle}} \ . \tag{2.16}$$

Finding the argmin in the Eq. (2.15) is formally equivalent to [74]:

$$\delta\theta^* = \underset{\delta\theta}{\text{argmax}} \ \mathrm{F}^2\left[\mathrm{e}^{-2\tau\hat{H}}|\Psi_\theta\rangle, |\Psi_{\theta+\delta\theta}\rangle\right] \ , \tag{2.17}$$

where $\mathrm{F}[\cdot,\cdot]$ is known as *fidelity*, and it is defined as

$$\mathrm{F}^2\left[|\Psi\rangle, |\Phi\rangle\right] = \frac{\langle\Psi|\Phi\rangle\langle\Phi|\Psi\rangle}{\langle\Psi|\Psi\rangle\langle\Phi|\Phi\rangle} \ . \tag{2.18}$$

To simplify the notation we define the following states

$$|\Psi_\tau\rangle = \mathrm{e}^{-2\lambda\tau\hat{H}}|\Psi_\theta\rangle \qquad |\Psi_\delta\rangle = |\Psi_{\theta+\lambda\delta\theta}\rangle \ , \tag{2.19}$$

where we have introduces the control parameter λ which we will set equal to one at the end of the calculation. Note that we are assuming τ and $\delta\theta_\alpha$ to be of the same order of magnitude, this assumption will be validated at the end of the calculation.

In the following we will solve Eq. (2.17) in an approximated way, specifically we will neglect contributions of order $O(\lambda^3)$. First we expand both states in Eq. (2.19) up to the same order:

$$\begin{aligned}|\Psi_\tau\rangle = \mathrm{e}^{-2\lambda\tau\hat{H}}|\Psi_\theta\rangle &= \left(\hat{\mathbb{1}} - 2\lambda\tau\hat{H} + 4\lambda^2\tau^2\hat{H}^2\right)|\Psi_\theta\rangle + O(\lambda^3) \\ &= \left(\hat{\mathbb{1}} + \lambda\hat{\varepsilon} - \lambda^2\hat{\varepsilon}^2\right)|\Psi_\theta\rangle + O(\lambda^3) \ ,\end{aligned} \tag{2.20}$$

where we have introduced the hermitian operator $\hat{\varepsilon} = -2\tau\hat{H}$. Analogously, assuming real-valued parameters θ (in the case of complex-valued parameters we can always treat real and imaginary part separately), we expand the other state in Eq. (2.19):

$$\begin{aligned}|\Psi_\delta\rangle = |\Psi_{\theta+\lambda\delta\theta}\rangle &= |\Psi_\theta\rangle + \lambda\sum_{\alpha=1}^{P}\delta\theta_\alpha\frac{\partial}{\partial\theta_\alpha}|\Psi_\theta\rangle + \lambda^2\sum_{\alpha,\beta=1}^{P}\delta\theta_\alpha\delta\theta_\beta\frac{\partial^2}{\partial\theta_\alpha\theta_\beta}|\Psi_\theta\rangle + O(\lambda^3) \\ &= \left(\hat{\mathbb{1}} + \lambda\sum_{\alpha=1}^{P}\delta\theta_\alpha\hat{O}_\alpha + \lambda^2\sum_{\alpha,\beta=1}^{P}\delta\theta_\alpha\delta\theta_\beta\hat{O}_\alpha\hat{O}_\beta\right)|\Psi_\theta\rangle + O(\lambda^3) \ ,\end{aligned} \tag{2.21}$$

here we have used the diagonal operator $\hat{O}_\alpha$ defined in Eq. (2.8). We can further simplify the notation by defining[4] $\hat{R} = \sum_{\alpha=1}^{P} \delta\theta_\alpha \hat{O}_\alpha$:

$$|\Psi_\delta\rangle = \left(\hat{\mathbb{1}} + \lambda \hat{R} + \lambda^2 \hat{R}^2\right) |\Psi_\theta\rangle + O(\lambda^3) \ . \tag{2.22}$$

We stress that the states $|\Psi_\tau\rangle$ in Eq. (2.20) and $|\Psi_\delta\rangle$ in Eq. (2.22) are in general not normalized. Now we compute the various overlaps in the definition of the fidelity in Eq. (2.17). We start estimating the overlap between $|\Psi_\tau\rangle$ and $|\Psi_\delta\rangle$:

$$\begin{aligned} \langle\Psi_\tau|\Psi_\delta\rangle &= \langle\Psi_\theta| \left(\hat{\mathbb{1}} + \lambda\hat{\varepsilon} - \lambda^2\hat{\varepsilon}^2\right) \left(\hat{\mathbb{1}} + \lambda \hat{R} + \lambda^2 \hat{R}^2\right) |\Psi_\theta\rangle + O(\lambda^3) \\ &= \langle\Psi_\theta|\Psi_\theta\rangle \left[1 + \lambda \left(\langle\hat{R}\rangle + \langle\hat{\varepsilon}\rangle\right) + \lambda^2 \left(\langle\hat{\varepsilon}\hat{R}\rangle - \langle\hat{\varepsilon}^2\rangle + \langle\hat{R}^2\rangle\right)\right] + O(\lambda^3) \ , \end{aligned} \tag{2.23}$$

where for brevity we indicate $\langle\cdot\rangle = \langle\Psi_\theta| \cdot |\Psi_\theta\rangle \, / \, \langle\Psi_\theta|\Psi_\theta\rangle$. In a similar way we obtain:

$$\langle\Psi_\delta|\Psi_\tau\rangle = \langle\Psi_\theta|\Psi_\theta\rangle \left[1 + \lambda \left(\langle\hat{R}^\dagger\rangle + \langle\hat{\varepsilon}\rangle\right) + \lambda^2 \left(\langle\hat{R}^\dagger\hat{\varepsilon}\rangle - \langle\hat{\varepsilon}^2\rangle + \langle(\hat{R}^\dagger)^2\rangle\right)\right] + O(\lambda^3) \ . \tag{2.24}$$

We can compute the numerator of the fidelity [see Eq. (2.18)] performing the product of the overlaps in Eq. (2.23) and Eq. (2.24):

$$\begin{aligned} |\langle\Psi_\tau|\Psi_\delta\rangle|^2 = |\langle\Psi_\theta|\Psi_\theta\rangle|^2 &\left[1 + \lambda \left(\langle\hat{R}\rangle + 2\langle\hat{\varepsilon}\rangle + \langle\hat{R}^\dagger\rangle\right) + \right. \\ &\lambda^2 \left(\langle\hat{\varepsilon}\hat{R}\rangle + \langle\hat{R}^2\rangle + \langle\hat{R}^\dagger\rangle\langle\hat{R}\rangle + \langle\hat{R}^\dagger\rangle\langle\hat{\varepsilon}\rangle + \langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle + \langle\hat{\varepsilon}\rangle^2 - 2\langle\hat{\varepsilon}^2\rangle + \right. \\ &\left.\left. \langle\hat{R}^\dagger\hat{\varepsilon}\rangle + \langle(\hat{R}^\dagger)^2\rangle\right)\right] + O(\lambda^3) \ . \end{aligned} \tag{2.25}$$

Then we focus on the denominator of the fidelity [see Eq. (2.18)]

$$\begin{aligned} \langle\Psi_\tau|\Psi_\tau\rangle &= \langle\Psi_\theta| \left(\hat{\mathbb{1}} + \lambda\hat{\varepsilon} - \lambda^2\hat{\varepsilon}^2\right) \left(\hat{\mathbb{1}} + \lambda\hat{\varepsilon} - \lambda^2\hat{\varepsilon}^2\right) |\Psi_\theta\rangle + O(\lambda^3) \\ &= \langle\Psi_\theta|\Psi_\theta\rangle \left(1 + 2\lambda\langle\hat{\varepsilon}\rangle - \lambda^2\langle\hat{\varepsilon}^2\rangle\right) + O(\lambda^3) \ , \end{aligned} \tag{2.26}$$

analogously

$$\begin{aligned} \langle\Psi_\delta|\Psi_\delta\rangle &= \langle\Psi_\theta| \left(\hat{\mathbb{1}} + \lambda\hat{R}^\dagger + \lambda^2(\hat{R}^\dagger)^2\right) \left(\hat{\mathbb{1}} + \lambda\hat{R} + \lambda^2\hat{R}^2\right) |\Psi_\theta\rangle + O(\lambda^3) \\ &= \langle\Psi_\theta|\Psi_\theta\rangle \left[1 + \lambda \left(\langle\hat{R}\rangle + \langle\hat{R}^\dagger\rangle\right) + \lambda^2 \left(\langle\hat{R}^2\rangle + \langle(\hat{R}^\dagger)^2\rangle + \langle\hat{R}^\dagger R\rangle\right)\right] + O(\lambda^3) \ . \end{aligned} \tag{2.27}$$

[4] It is important to point out that the operator $\hat{R}$ is not hermitian, contrary to $\hat{\varepsilon}$ that it is hermitian.

Then we perform the product between the norms in Eq. (2.26) and in Eq. (2.27):

$$\langle\Psi_\tau|\Psi_\tau\rangle\langle\Psi_\delta|\Psi_\delta\rangle = |\langle\Psi_\theta|\Psi_\theta\rangle|^2\Big[1+\lambda\Big(\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle+2\langle\hat{\varepsilon}\rangle\Big)+ \\ \lambda^2\Big(\langle\hat{R}^2\rangle+\langle(\hat{R}^\dagger)^2\rangle+\langle\hat{R}^\dagger R\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}^\dagger\rangle-\langle\hat{\varepsilon}^2\rangle\Big)+O(\lambda^3)\Big]\ . \tag{2.28}$$

Employing the series expansion $1/(1+x)=1-x+x^2+O(x^3)$ for $x\ll 1$ we obtain:

$$\frac{1}{\langle\Psi_\tau|\Psi_\tau\rangle\langle\Psi_\delta|\Psi_\delta\rangle}=\frac{1}{|\langle\Psi_\theta|\Psi_\theta\rangle|^2}\Big[1-\lambda\Big(\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle+2\langle\hat{\varepsilon}\rangle\Big)+\lambda^2\Big(\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle+2\langle\hat{\varepsilon}\rangle\Big)^2- \\ \lambda^2\Big(\langle\hat{R}^2\rangle+\langle(\hat{R}^\dagger)^2\rangle+\langle\hat{R}^\dagger R\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}^\dagger\rangle-\langle\hat{\varepsilon}^2\rangle\Big)+O(\lambda^3)\Big]\ . \tag{2.29}$$

The fidelity in Eq. (2.18) can be computed by performing the product between the expression in Eq. (2.25) and in Eq. (2.29):

$$\frac{|\langle\Psi_\tau|\Psi_\delta\rangle|^2}{\langle\Psi_\tau|\Psi_\tau\rangle\langle\Psi_\delta|\Psi_\delta\rangle}=1-\cancel{\lambda\Big(\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle+2\langle\hat{\varepsilon}\rangle\Big)}+\cancel{\lambda^2\Big(\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle+2\langle\hat{\varepsilon}\rangle\Big)^2}- \\ \lambda^2\Big(\cancel{\langle\hat{R}^2\rangle}+\cancel{\langle(\hat{R}^\dagger)^2\rangle}+\langle\hat{R}^\dagger R\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle+2\langle\hat{\varepsilon}\rangle\langle\hat{R}^\dagger\rangle-\langle\hat{\varepsilon}^2\rangle\Big)+ \\ \cancel{\lambda\Big(\langle\hat{R}\rangle+2\langle\hat{\varepsilon}\rangle+\langle\hat{R}^\dagger\rangle\Big)}-\cancel{\lambda^2\Big(\langle\hat{R}\rangle+2\langle\hat{\varepsilon}\rangle+\langle\hat{R}^\dagger\rangle\Big)^2}+ \\ \lambda^2\Big(\langle\hat{\varepsilon}\hat{R}\rangle+\cancel{\langle\hat{R}^2\rangle}+\langle\hat{R}^\dagger\rangle\langle\hat{R}\rangle+\langle\hat{R}^\dagger\rangle\langle\hat{\varepsilon}\rangle+\langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle+\langle\hat{\varepsilon}\rangle^2-2\langle\hat{\varepsilon}^2\rangle+ \\ \langle\hat{R}^\dagger\hat{\varepsilon}\rangle+\cancel{\langle(\hat{R}^\dagger)^2\rangle}\Big)+O(\lambda^3)\ . \tag{2.30}$$

After appropriate simplifications, the remaining terms can be arranged to define correlations functions:

$$\frac{|\langle\Psi_\tau|\Psi_\delta\rangle|^2}{\langle\Psi_\tau|\Psi_\tau\rangle\langle\Psi_\delta|\Psi_\delta\rangle}=1+\lambda^2\Big[\Big(\langle\hat{\varepsilon}\hat{R}\rangle-\langle\hat{\varepsilon}\rangle\langle\hat{R}\rangle\Big)+\Big(\langle\hat{R}^\dagger\hat{\varepsilon}\rangle-\langle\hat{R}^\dagger\rangle\langle\hat{\varepsilon}\rangle\Big)- \\ \big(\langle\hat{\varepsilon}^2\rangle-\langle\hat{\varepsilon}\rangle^2\big)-\Big(\langle\hat{R}^\dagger\hat{R}\rangle-\langle\hat{R}^\dagger\rangle\langle\hat{R}\rangle\Big)\Big]+O(\lambda^3). \tag{2.31}$$

Notably, first order contributions in λ do not appear in the final expression. Moreover, the above expression can be obtained also if the expansion of $|\Psi_\tau\rangle$ [see Eq. (2.20)] and $|\Psi_\delta\rangle$ [see Eq. (2.22)] are performed neglecting terms of order λ^2, since all of them cancels out when performing the ratio in Eq. (2.31) [75].

By using the definitions of $\hat{R}$ and $\hat{\varepsilon}$ we can rewrite the different terms:

1. Quantum Geometric Tensor:

$$\langle \hat{R}^\dagger \hat{R}\rangle - \langle \hat{R}^\dagger\rangle \langle \hat{R}\rangle = \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta \left(\langle \hat{O}^\dagger_\alpha \hat{O}_\beta\rangle - \langle \hat{O}^\dagger_\alpha\rangle \langle \hat{O}_\beta\rangle\right) = \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta Q_{\alpha\beta} \ . \quad (2.32)$$

Here, we have introduced the *Quantum Geometric Tensor*:

$$\boxed{Q_{\alpha\beta} = \langle \hat{O}^\dagger_\alpha \hat{O}_\beta\rangle - \langle \hat{O}^\dagger_\alpha\rangle \langle \hat{O}_\beta\rangle} \ , \quad (2.33)$$

which it is an hermitian matrix $Q_{\alpha\beta} = Q^*_{\beta\alpha}$ of $P \times P$ complex-valued numbers (assuming to treat the the general case of complex-valued variational wave functions $\Psi_\theta(\sigma) \in \mathbb{C}$).

2. Gradient of the Energy:

$$\begin{aligned} \langle \hat{R}^\dagger \hat{\varepsilon}\rangle - \langle \hat{R}^\dagger\rangle \langle \hat{\varepsilon}\rangle &= -2\tau \sum_{\alpha=1}^{P} \delta\theta_\alpha \left(\langle \hat{O}^\dagger_\alpha \hat{H}\rangle - \langle \hat{O}^\dagger_\alpha\rangle \langle \hat{H}\rangle\right) \ , \\ \langle \hat{\varepsilon}\hat{R}\rangle - \langle \hat{\varepsilon}\rangle \langle \hat{R}\rangle &= -2\tau \sum_{\alpha=1}^{P} \delta\theta_\alpha \left(\langle \hat{H}\hat{O}_\alpha\rangle - \langle \hat{H}\rangle \langle \hat{O}_\alpha\rangle\right) \ , \end{aligned} \quad (2.34)$$

assuming $\delta\theta_\alpha \in \mathbb{R} \ \ \forall \alpha = 1, \dots, P$, the two equations above are one the complex conjugate of the other, and since in the fidelity the two terms are summed together [see Eq. (2.31)] we can write:

$$\begin{aligned} \left(\langle \hat{\varepsilon}\hat{R}\rangle - \langle \hat{\varepsilon}\rangle \langle \hat{R}\rangle\right) + \left(\langle \hat{R}^\dagger \hat{\varepsilon}\rangle - \langle \hat{R}^\dagger\rangle \langle \hat{\varepsilon}\rangle\right) &= -4\tau \sum_{\alpha=1}^{P} \delta\theta_\alpha \Re\left\{\langle \hat{H}\hat{O}_\alpha\rangle - \langle \hat{H}\rangle \langle \hat{O}_\alpha\rangle\right\} \\ &= 2\tau \sum_{\alpha=1}^{P} \delta\theta_\alpha F_\alpha. \end{aligned} \quad (2.35)$$

In the last step we have used the definition of the gradient of the variational energy with respect to the variational parameters $F_\alpha = -\partial E_\theta / \partial\theta_\alpha$ [see Eq. (2.9)].

3. Variance of the energy:

$$\langle \varepsilon^2\rangle - \langle \varepsilon\rangle^2 = 4\tau^2 \left(\langle H^2\rangle - \langle H\rangle^2\right) = 4\tau^2 \mathrm{Var}(H) \ , \quad (2.36)$$

where $\mathrm{Var}(H)$ is the variance of the Hamiltonian defined in Eq. (1.17). This term does not depend on the variation of the parameters $\delta\theta_\alpha$ so it will not contribute to the final equations for the parameter updates.

We can rewrite the fidelity in Eq. (2.31) as:

$$\mathrm{F}^2[\left|\Psi_\tau\right\rangle, \left|\Psi_\delta\right\rangle] = 1 + \lambda^2 \left[2\tau \sum_{\alpha=1}^{P} \delta\theta_\alpha F_\alpha - \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta Q_{\alpha\beta} - 4\tau^2 \mathrm{Var}(H) \right] + O(\lambda^3) \ . \tag{2.37}$$

We emphasize that our derivation leads to the same expression of the fidelity obtained in Ref. [74] and in the Supplemental Material of Ref. [75].

The update of the parameters $\delta\theta$ which maximize the fidelity can be obtained by deriving the above expression with respect to them and set the result to zero:

$$\sum_{\beta=1}^{P} \left(Q_{\alpha\beta} + Q_{\beta\alpha}\right) \delta\theta_\beta = 2\tau F_\alpha \ . \tag{2.38}$$

By exploiting the hermiticity of the Quantum Geometric Tensor $Q_{\alpha\beta} = Q^*_{\alpha\beta}$ we finally obtain the equations for the update of the variational parameters:

$$\boxed{\sum_{\beta=1}^{P} S_{\alpha,\beta} \delta\theta_\beta = \tau F_\alpha} \ , \tag{2.39}$$

where we have introduce the S matrix, namely the real part of the Quantum Geometric Tensor $S_{\alpha,\beta} = \Re\{Q_{\alpha\beta}\}$.

We emphasize that in the SR updates in Eq. (2.39), the variation of the parameters $\delta\theta_\alpha$ is proportional to τ, consistent with the assumption made at the beginning of the calculation.

2.2.1 Geometrical Interpretation of the Quantum Geometric Tensor

In the previous section the Quantum Geometric Tensor emerged naturally when deriving the parameter updates of the SR. Here, we provide a geometrical interpretation of the matrix Q. In order to do so, we measure the Fubini-Study distance [see Eq. (2.15)] between the variational states $\left|\Psi_\theta\right\rangle$ and $\left|\Psi_{\theta+\delta\theta}\right\rangle$. The latter can be written as

$$\mathcal{F}^2[\left|\Psi_\theta\right\rangle, \left|\Psi_{\theta+\delta\theta}\right\rangle] = \arccos^2\left(\mathrm{F}[\left|\Psi_\theta\right\rangle, \left|\Psi_{\theta+\delta\theta}\right\rangle]\right) \ , \tag{2.40}$$

where the fidelity [see Eq. (2.18)] between these two states is simply obtained by setting $\tau = 0$ in Eq. (2.37):

$$\mathrm{F}^2[|\Psi_\theta\rangle, |\Psi_{\theta+\delta\theta}\rangle] = 1 - \lambda^2 \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta Q_{\alpha\beta} + O(\lambda^3) \ . \tag{2.41}$$

With the previous expressions we can compute the Fubini-Study distance:

$$\begin{aligned} \mathcal{F}^2[|\Psi_\theta\rangle, |\Psi_{\theta+\delta\theta}\rangle] &= \arccos^2\left(\sqrt{1 - \lambda^2 \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta Q_{\alpha\beta} + O(\lambda^3)}\right) \\ &= \lambda^2 \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta Q_{\alpha\beta} + O(\lambda^4) \ , \end{aligned} \tag{2.42}$$

where in the last step we used the expansion $\arccos^2(\sqrt{1+x}) = -x + O(x^2)$, for $x \ll 1$.

In general for complex-valued wave functions, Q is a complex-valued hermitian matrix, hence we can write $Q_{\alpha\beta} = S_{\alpha\beta} + iC_{\alpha\beta}$, where $S_{\alpha\beta} = \Re\{Q_{\alpha\beta}\}$ is a symmetric matrix ($S_{\alpha\beta} = S_{\beta\alpha}$) and $C_{\alpha\beta} = \Im\{Q_{\alpha\beta}\}$ is an antisymmetric matrix ($C_{\alpha\beta} = -C_{\beta\alpha}$). Assuming real-valued parameters, $\delta\theta_\alpha\delta\theta_\beta$ is a symmetric matrix, therefore $\sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha\delta\theta_\beta C_{\alpha\beta} = 0$. At the end, the infinitesimal distance in the Hilbert space between the two quantum states $|\Psi_\theta\rangle$ and $|\Psi_{\theta+\delta\theta}\rangle$ is given by:

$$\boxed{\mathcal{F}^2[|\Psi_\theta\rangle, |\Psi_{\theta+\delta\theta}\rangle] = \sum_{\alpha,\beta=1}^{P} \delta\theta_\alpha \delta\theta_\beta S_{\alpha\beta}} \ . \tag{2.43}$$

The S matrix, which is the real part of the Quantum Geometric Tensor Q, can be shown to be a symmetric and positive-definite matrix [63]. Hence, it defines the metric in Hilbert space used to measure the distance between quantum states. The Quantum Geometric Tensor represents a generalization of the *Fisher information metric* for classical probability distributions [76].

2.2.2 Gauge Invariance of the Quantum Geometric Tensor

Each quantum state in the Hilbert space can be defined up to a phase. In this section we show that the S matrix is invariant with respect to the gauge transformation $|\Psi_\theta\rangle \to |\Psi'_\theta\rangle = e^{i\phi_\theta}|\Psi_\theta\rangle$, where $\phi_\theta \in \mathbb{R}$ is a generic function of the variational parameters,

thus defining an appropriate metric to measure distances in the Hilbert space.
The aim of this section is to evaluate the QGT on the state $|\Psi'_\theta\rangle$:

$$Q'_{\alpha,\beta} = \frac{\langle\Psi'_\theta|\hat{O}^\dagger_\alpha\hat{O}_\beta|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle} - \frac{\langle\Psi'_\theta|\hat{O}^\dagger_\alpha|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle}\frac{\langle\Psi'_\theta|\hat{O}_\beta|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle} \ . \tag{2.44}$$

First we note that $\langle\Psi'_\theta|\Psi'_\theta\rangle = \langle\Psi_\theta|\Psi_\theta\rangle$ since they are related by a gauge transformation. Then, given a physical configuration $|\sigma\rangle$ we evaluate:

$$\begin{aligned}\langle\sigma|\hat{O}_\alpha|\Psi'_\theta\rangle &= \frac{\partial}{\partial\theta_\alpha}\left[\mathrm{Log}\left(e^{i\phi_\theta}\langle\sigma|\Psi_\theta\rangle\right)\right] \\ &= i\frac{\partial\phi_\theta}{\partial\theta_\alpha} + \frac{\partial\mathrm{Log}\left(\langle\sigma|\Psi_\theta\rangle\right)}{\partial\theta_\alpha} \\ &= \langle\sigma|\left(i\frac{\partial\phi_\theta}{\partial\theta_\alpha} + \hat{O}_\alpha|\Psi_\theta\rangle\right) \ .\end{aligned} \tag{2.45}$$

Since the previous relation is valid for each configuration $|\sigma\rangle$ of the basis of the Hilbert space, then it is a relation between the operators:

$$\hat{O}_\alpha|\Psi'_\theta\rangle = i\frac{\partial\phi_\theta}{\partial\theta_\alpha} + \hat{O}_\alpha|\Psi_\theta\rangle \ . \tag{2.46}$$

At this point we can compute the first term in Eq. (2.44) using the previous expression for $\hat{O}_\alpha|\Psi'_\theta\rangle$:

$$\begin{aligned}\langle\Psi'_\theta|\hat{O}^\dagger_\alpha\hat{O}_\beta|\Psi'_\theta\rangle &= \langle\Psi_\theta|\left(-i\frac{\partial\phi_\theta}{\partial\theta_\alpha} + \hat{O}^\dagger_\alpha\right)\left(i\frac{\partial\phi_\theta}{\partial\theta_\beta} + \hat{O}_\beta\right)|\Psi_\theta\rangle \\ &= \langle\Psi_\theta|\hat{O}^\dagger_\alpha\hat{O}_\beta|\Psi_\theta\rangle - i\frac{\partial\phi_\theta}{\partial\theta_\alpha}\langle\Psi_\theta|\hat{O}_\beta|\Psi_\theta\rangle + i\frac{\partial\phi_\theta}{\partial\theta_\beta}\langle\Psi_\theta|\hat{O}^\dagger_\alpha|\Psi_\theta\rangle + \frac{\partial\phi_\theta}{\partial\theta_\alpha}\frac{\partial\phi_\theta}{\partial\theta_\beta} \ ,\end{aligned} \tag{2.47}$$

with the proper normalization we obtain:

$$\frac{\langle\Psi'_\theta|\hat{O}^\dagger_\alpha\hat{O}_\beta|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle} = \langle\hat{O}^\dagger_\alpha\hat{O}_\beta\rangle + i\frac{\partial\phi_\theta}{\partial\theta_\beta}\langle\hat{O}^\dagger_\alpha\rangle - i\frac{\partial\phi_\theta}{\partial\theta_\alpha}\langle\hat{O}_\beta\rangle + \frac{\partial\phi_\theta}{\partial\theta_\alpha}\frac{\partial\phi_\theta}{\partial\theta_\beta} \ , \tag{2.48}$$

where as in the previous section we indicate $\langle\cdot\rangle = \langle\Psi_\theta|\cdot|\Psi_\theta\rangle / \langle\Psi_\theta|\Psi_\theta\rangle$.

Analogously we compute the second term in Eq. (2.44), namely:

$$\begin{aligned}\langle\Psi'_\theta|\hat{O}_\beta|\Psi'_\theta\rangle &= e^{-i\phi_\theta}\langle\Psi_\theta|\left(i\frac{\partial\phi_\theta}{\partial\theta_\beta} + \hat{O}_\beta\right)|\Psi_\theta\rangle \\ &= e^{-i\phi_\theta}\left(i\frac{\partial\phi_\theta}{\partial\theta_\beta}\langle\Psi_\theta|\Psi_\theta\rangle + \langle\Psi_\theta|\hat{O}_\beta|\Psi_\theta\rangle\right) \ ,\end{aligned} \tag{2.49}$$

in the same way we obtain $\langle\Psi'_\theta|\hat{O}^\dagger_\alpha|\Psi'_\theta\rangle = e^{i\phi_\theta}\left(-i\frac{\partial\phi_\theta}{\partial\theta_\alpha}\langle\Psi_\theta|\Psi_\theta\rangle + \langle\Psi_\theta|\hat{O}^\dagger_\alpha|\Psi_\theta\rangle\right)$.

Normalizing the expectation values we get:

$$\frac{\langle\Psi'_\theta|\hat{O}^\dagger_\alpha|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle}\frac{\langle\Psi'_\theta|\hat{O}_\beta|\Psi'_\theta\rangle}{\langle\Psi'_\theta|\Psi'_\theta\rangle} = \langle\hat{O}^\dagger_\alpha\rangle\langle\hat{O}_\beta\rangle + i\frac{\partial\phi_\theta}{\partial\theta_\beta}\langle\hat{O}^\dagger_\alpha\rangle - i\frac{\partial\phi_\theta}{\partial\theta_\alpha}\langle\hat{O}_\beta\rangle + \frac{\partial\phi_\theta}{\partial\theta_\alpha}\frac{\partial\phi_\theta}{\partial\theta_\beta}\ . \tag{2.50}$$

At the end, replacing the results of Eq. (2.49) and Eq. (2.50) in the definition of the Quantum Geometric Tensor [see Eq. (2.44)]

$$Q'_{\alpha,\beta} = \langle\hat{O}^\dagger_\alpha\hat{O}_\beta\rangle - \langle\hat{O}^\dagger_\alpha\rangle\langle\hat{O}_\beta\rangle = Q_{\alpha,\beta}\ . \tag{2.51}$$

Note that being $S_{\alpha,\beta} = \Re\{Q_{\alpha,\beta}\}$, as a result the S matrix is invariant with respect to gauge transformations, hence it is a proper metric in the Hilbert space.

2.3 A simple linear algebra identity to optimize large-scale variational states

Over the past few years, neural networks have been extensively used as powerful variational *Ansätze* for studying interacting spin models [7], and the number of parameters have increased significantly[5]. These deep learning models have great performances when the number of parameters is large. A significant bottleneck arises when employing the original formulation of SR for optimization, as it is based on the inversion of a matrix of size $P\times P$, where P denotes the number of parameters [see Eq. (2.39)]. Consequently, this approach becomes computationally infeasible as the parameter count exceeds $O(10^4)$, primarily due to the constraints imposed by the limited memory capacity of current-generation *Graphics Processing Units* (GPUs). Recently, Chen and Heyl [36] made a step forward in the optimization procedure by introducing an alternative method, dubbed *MinSR*, to train Neural-Network Quantum States. MinSR does not require inverting the original $P \times P$ matrix but instead a much smaller $M \times M$ one, where M is the number of configurations used to estimate the SR matrix. This is convenient in the deep learning setup where $P \gg M$. Most importantly, this procedure avoids allocating the $P \times P$ matrix, reducing the memory cost. However, this formulation is obtained by minimizing the Fubini-Study distance with an ad hoc constraint. In this section, we first use a simple relation from linear algebra to show, in a transparent way, that SR can be rewritten exactly in a form

[5] For a detailed discussion on Neural-Network Quantum States refer to Chapter 3.

which involves inverting a small $M \times M$ matrix (in case of real-valued wave functions and a $2M \times 2M$ matrix for complex-valued ones) and that only a standard regularization of the SR matrix is required.

2.3.1 How to solve Stochastic Reconfiguration's equations?

The SR updates [63, 68, 74] are constructed solving the linear system in Eq. (2.39). It is important to consider that the matrix S may possess extremely small or even negligible eigenvalues. This characteristic implies that applying its inverse to the vector $\boldsymbol{F}$ [see Eq. (2.39)] can lead to numerical instability [63]. Such a situation may arise, for instance, when the matrix S becomes singular due to redundancies in the wave function parametrization. To mitigate these potential issues, we use the following modified update scheme:

$$\boldsymbol{\delta\theta} = \tau \left(S + \lambda \mathbb{1}_P\right)^{-1} \boldsymbol{F} \ , \tag{2.52}$$

where τ is the learning rate and $\lambda > 0$ is a regularization parameter to ensure the invertibility of the S matrix. The matrix S has shape $P \times P$ and it is defined in terms of the $\hat{O}_\alpha$ operators [63] (see Sec. 2.2)

$$S_{\alpha,\beta} = \Re \left[\langle (\hat{O}_\alpha - \langle \hat{O}_\alpha \rangle)^\dagger (\hat{O}_\beta - \langle \hat{O}_\beta \rangle) \rangle \right] \ . \tag{2.53}$$

The Eq. (2.52) defines the standard formulation of the SR, which involves the inversion of a $P \times P$ matrix, being the bottleneck of this approach when the number of parameters is larger than $O(10^4)$. To address this problem, we start reformulating Eq. (2.52) in a more convenient way. For a given sample of M spin configurations $\{\sigma_1, \dots, \sigma_M\}$ sampled according to $P_\theta(\sigma)$ [see Eq. (1.12)], the stochastic estimate of F_α can be obtained as reported in Eq. (2.12). Equivalently, Eq. (2.53) can be stochastically estimated as

$$\bar{S}_{\alpha,\beta} = \Re \left[\frac{1}{M} \sum_{i=1}^{M} \left[O_\alpha(\sigma_i) - \bar{O}_\alpha \right]^* \left[O_\beta(\sigma_i) - \bar{O}_\beta \right] \right] \ . \tag{2.54}$$

Notice that we adopt the convention of using latin and greek indices to run over configurations and parameters, respectively. To simplify further, we introduce $Y_{\alpha i} = (O_\alpha(\sigma_i) - \bar{O}_\alpha)/\sqrt{M}$ and $\varepsilon_i = -2[E_L(\sigma_i) - \bar{E}_L]^*/\sqrt{M}$, allowing us to express Eq. (2.12) in matrix notation as $\bar{\boldsymbol{F}} = \Re[Y \boldsymbol{\varepsilon}]$ and Eq. (2.53) as $\bar{S} = \Re[Y Y^\dagger]$. Writing $Y = Y_R + iY_I$ we obtain:

$$\bar{S} = Y_R Y_R^T + Y_I Y_I^T = X X^T \tag{2.55}$$

where $X = \text{Concat}(Y_R, Y_I) \in \mathbb{R}^{P\times 2M}$, the concatenation being along the last axis. Furthermore, using $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_R + i\boldsymbol{\varepsilon}_I$, the gradient of the energy can be recast as

$$\bar{\boldsymbol{F}} = Y_R\boldsymbol{\varepsilon}_R - Y_I\boldsymbol{\varepsilon}_I = X\boldsymbol{f} \ , \tag{2.56}$$

with $\boldsymbol{f} = \text{Concat}(\boldsymbol{\varepsilon}_R, -\boldsymbol{\varepsilon}_I) \in \mathbb{R}^{2M}$. Then, the update of the parameters in Eq. (2.52) can be written as

$$\boldsymbol{\delta\theta} = \tau(XX^T + \lambda\mathbb{1}_P)^{-1}X\boldsymbol{f} \ . \tag{2.57}$$

This reformulation of the SR updates is a crucial step, which allows the use of a simple linear algebra identity.

Push-through / Woodbury identity

Given A and B matrices respectively with dimensions $n\times m$ and $m\times n$, the following matrix identity holds for each $\lambda > 0$ [77, 78]:

$$\boxed{(AB + \lambda\mathbb{1}_n)^{-1}A = A(BA + \lambda\mathbb{1}_m)^{-1}} \ . \tag{2.58}$$

This identity can be proved starting from

$$\mathbb{1}_m = (BA + \lambda\mathbb{1}_m)(BA + \lambda\mathbb{1}_m)^{-1} \ , \tag{2.59}$$

then, multiplying from the left by A, we get

$$A = A(BA + \lambda\mathbb{1}_m)(BA + \lambda\mathbb{1}_m)^{-1} \ , \tag{2.60}$$

and exploiting the fact that $A\mathbb{1}_m = \mathbb{1}_n A$, we obtain

$$A = (AB + \lambda\mathbb{1}_n)A(BA + \lambda\mathbb{1}_m)^{-1} \ . \tag{2.61}$$

At the end, multiplying from the left by $(AB + \lambda\mathbb{1}_n)^{-1}$, we recover Eq. (2.58).

As a result, Eq. (2.57) can be rewritten as [27]

$$\boxed{\boldsymbol{\delta\theta} = \tau X(X^TX + \lambda\mathbb{1}_{2M})^{-1}\boldsymbol{f}} \ . \tag{2.62}$$

Although very simple, this derivation is an important result, indeed it shows, in a simple and transparent way, how to exactly perform the SR with the inversion of a $2M \times 2M$

matrix and, therefore, without allocating a $P \times P$ matrix. We emphasize that the last formulation is very useful in the typical deep learning setup, where $P \gg M$. Employing Eq. (2.62) instead of Eq. (2.57) proves to be more efficient in terms of both computational complexity and memory usage. The required operations for this new formulation are $O(M^2P) + O(M^3)$ instead of $O(P^3)$, and the memory usage is only $O(MP)$ instead of $O(P^2)$. For deep neural networks with n_l layers the memory usage can be further reduced roughly to $O(MP/n_l)$ (see Ref. [79]).

Other methods, based on iterative solvers, require $O(nMP)$ operations, where n is the number of steps needed to solve the linear problem in Eq. (2.52). However, this number increases significantly for ill-conditioned matrices (the matrix S has a number of zero eigenvalues equal to $P - M$), leading to many non-parallelizable iteration steps and consequently higher computational costs [80]. Our proof also highlights that the diagonal-shift regularization of the S matrix in parameter space [see Eq. (2.52)] is equivalent to the same diagonal shift in sample space [see Eq. (2.62)]. In contrast, for the MinSR update [36], a pseudo-inverse regularization is applied in order to truncate the effect of vanishing singular values during inversion.

2.3.2 Parallel and memory-efficient implementation on multiple GPUs

We developed a memory-efficient implementation of SR that is optimized for deployment on a multi-node GPU cluster, ensuring scalability and practicality for real-world applications. Indeed, the algorithm proposed in Eq. (2.62) can be efficiently distributed, both in terms of computational operations and memory, across multiple GPUs. To illustrate this, we consider for simplicity the case of a real-valued wave function, where $X = Y_R \equiv Y$. Given a number M of configurations, they can be distributed across n_G GPUs, facilitating parallel simulation of Markov chains. In this way, on the *g-th* GPU, the elements $i \in [gM/n_G, (g+1)M/n_G)$ of the vector $\boldsymbol{f}$ are obtained, along with the columns $i \in [gM/n_G, (g+1)M/n_G)$ of the matrix X, which we indicate using $X_{[:,g]}$. To efficiently apply Eq. (2.62), we employ the Message Passing Interface (MPI) *alltoall* collective operation to transpose X, yielding the sub-matrix $X_{[g,:]}$ on g-th GPU. This sub-matrix comprises the rows elements in $[gP/n_G, (g+1)P/n_G)$ of the original matrix X (see right

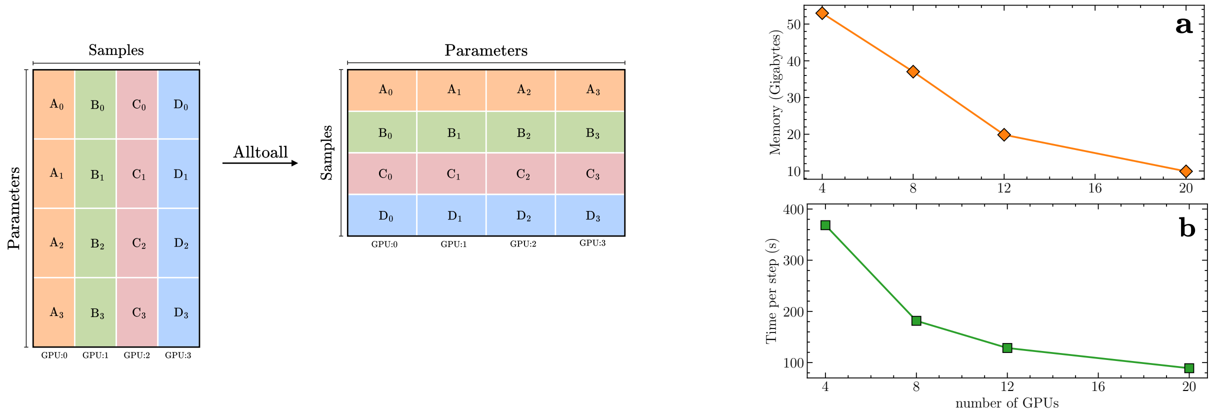

Figure 2.2: **Left panel**: Graphical representation of MPI *alltoall* operation to transpose the X matrix distributed across multiple GPUs. For example, GPU:0 initially contains sub-matrices A_0, A_1, A_2, A_3, while following the transposition, GPU:0 contains sub-matrices A_0, B_0, C_0, D_0. **Right panel**: Memory usage in Gigabytes (panel **a**) and computational time per optimization step in seconds (panel **b**) as a function of the number of GPUs. The reported values are related to a Vision Transformer architeture with $h = 12$, $d = 72$, $n_l = 8$, fully symmetrized and optimized with $M = 6000$ samples (see Sec. 4.2.3).

panel of Fig. 2.2). Consequently, we can express:

$$X^T X = \sum_{g=0}^{n_G-1} X^T_{[g,:]} X_{[g,:]} \ . \tag{2.63}$$

The inner products can be computed in parallel on each GPU, while the outer sum is performed using the MPI primitive *reduce* with the *sum* operation. The *master* GPU performs the inversion, computes the vector $\boldsymbol{t} = (X^T X + \lambda \mathbb{1}_{2M})^{-1} \boldsymbol{f}$, and then scatters it across the other GPUs. Finally, after transposing again the matrix X with the MPI *alltoall* operation, the parameter update can be computed as follows:

$$\boldsymbol{\delta\theta} = \tau \sum_{g=0}^{n_G-1} X_{[:,g]} \boldsymbol{t}_g \ . \tag{2.64}$$

This procedure significantly reduces the memory requirement per GPU to $O(MP/n_G)$, enabling the optimization of an arbitrary number of parameters using the SR approach. In the left panel of Fig. 2.2 we report the memory usage and the computational time per optimization step.

This formulation of the Stochastic Reconfiguration is implemented in NetKet [81], under the name of `VMC_SRt`.

2.4 Pseudocode of the Variational Monte Carlo algorithm

At this stage, we have established all the necessary components to define the VMC algorithm for optimizing a variational state $|\Psi_\theta\rangle$ to minimize the variational energy E_θ associated with a given Hamiltonian $\hat{H}$. For clarity, the main steps of the algorithm are summarized in the following pseudocode:

Algorithm 1 Variational Monte Carlo

```
Require: Define a variational state $\Psi_\theta(\sigma)$
Require: Initialize randomly the variational parameters $\theta$
for $t = 1, N_{\text{opt}}$ do
    samples of $M$ configurations $\{\sigma_1, \ldots, \sigma_M\} \sim P_\theta(\sigma)$ via MCMC
    Stochastic estimation of the gradient of the energy : $F_\alpha = -\partial_\alpha E_\theta$ with $\alpha = 1, \ldots, P$
    Stochastic estimation of the Quantum Geometric Tensor : $S_{\alpha,\beta}$ with $\alpha, \beta = 1, \ldots, P$
    Update of the parameters with SR: $\boldsymbol{\delta\theta} = \tau \left(S + \lambda \mathbb{1}_P\right)^{-1} \boldsymbol{F}$
    New parameters : $\theta_\alpha \leftarrow \theta_\alpha + \delta\theta_\alpha$ with $\alpha = 1, \ldots, P$
end for
```

It is important to note that the number of optimization steps, N_{opt}, is not known a priori. The required number of steps can vary significantly depending on the type of wave function being optimized. Typically, convergence is monitored by tracking the variational energy E_θ; however, other observables may require additional steps to achieve convergence.

We also point out that for the stochastic estimation of the energy gradient [see Eq. (2.12)] and the S-matrix [see Eq. (2.54)], only the derivatives of the logarithm of the wave function are required [see Eq. (2.8)]. These log-derivatives can be efficiently computed using automatic differentiation techniques [82], which is particularly advantageous when the wave function is parameterized by a large number of variational parameters, as is the case with Neural-Network Quantum States. Notably, both the MCMC sampling process for estimating observables (see Sec. 1.4) and parameter optimization (see Secs. 2.1 and 2.2) can be implemented using only the logarithm of the wave function, $\text{Log}[\Psi_\theta(\sigma)]$. This fact helps to avoid numerical instabilities, such as overflow or underflow [81], when computing the wave function's amplitude, as discussed in Sec. 1.4.3.

Chapter 3

Neural-Network Quantum States

In this Chapter, we explore a class of variational wave functions known as *Neural-Network Quantum States* (NQS), recently introduced by Carleo and Troyer [7]. We begin by introducing the foundational concepts that contribute to the success of neural-network architectures, and their adaptability to the quantum many-body problem. Then, we will progress from basic NQS architectures to more sophisticated ones, highlighting their increasing complexity and effectiveness showing their performances and limitations on a benchmark model: the J_1-J_2 Heisenberg model in one dimension.

3.1 Basic Concepts of Neural Networks

From a mathematical perspective, an artificial neural network, or simply a neural network, is a *non-linear* function that maps inputs from a potentially high-dimensional space to a desired output space [83]. The fundamental components of a neural network are known as *neurons*. A neuron is defined as the composition of two functions that take k input variables $\boldsymbol{x} = (x_1, \dots, x_k)$ and return a scalar output $F(\boldsymbol{x})$, expressed as:

$$F(\boldsymbol{x}) = g \circ q(\boldsymbol{x}) \ , \tag{3.1}$$

where the function $q(\cdot)$ is a linear transformation given by:

$$q(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \ . \tag{3.2}$$

Here, $\boldsymbol{w} = (w_1, \dots, w_k)$ is a vector of k elements referred to as *weights*, and b is a scalar parameter called *bias*. The *non-linear* function $g(\cdot)$, typically non-polynomial, operates

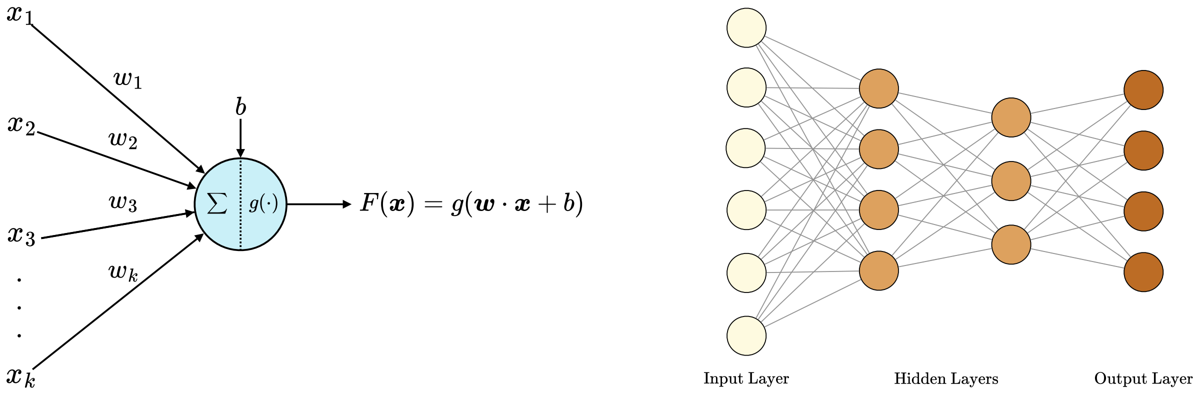

Figure 3.1: **Left panel**: Schematic representation of a single neuron. **Right panel**: Architecture of a feed-forward neural network composed by a visible layer, a output layer and two hidden layers.

on a scalar input and returns a scalar output. The latter is commonly referred to as *activation function* (see left panel of Fig. 3.1). It is important to emphasize that its choice is a crucial part of the neural network architecture and remains fixed during training. In contrast, the weights and biases are parameters that are adjusted iteratively during the training process. The structure of neural networks is inspired by the way biological organisms process information. In brains, neurons, electrically activated nerve cells, are interconnected by synapses that facilitate the transfer of information between neurons. Similarly, in artificial neural networks, once the functionality of a single neuron is defined, multiple neurons can be connected to form a network.

The connections between neurons imply that the output from one set of neurons in a given layer serves as the input for the next set of neurons in the subsequent layer. This sequential flow of information defines a specific direction from layer to layer, characterizing the architecture as a *feed-forward neural network* (FFN). As an example, in the right panel of Fig. 3.1, we show a graphical representation of a FFN.

The general structure of a neural network can be summarized in three main components:

- *Input/Visible Layer*: This is the first layer of the network, where the raw data is introduced.

- *Output Layer*: This is the final layer of the network, where the predicted results are produced.

- *Hidden Layers*: These layers lie between the input and output layers. They are termed *hidden* because they do not directly interact with the input data or produce direct output; instead, they perform intermediate computations that enable the network to learn complicated representations.

We point out that feed-forward neural networks represent one of the simplest neural network architectures. Over the years, neural networks have been progressively enhanced with additional fundamental components, such as layer normalization [84] and skip connections [85], which are crucial for facilitating the optimization process. Despite these enhancements, FFNs remain mathematically sufficient to represent any function with arbitrary precision, as we will discuss in the next section.

3.1.1 Universal approximators of arbitrary functions

The power of neural networks lies in the use of hidden layers with non-linear activation functions, which significantly enhance their representational capacity compared to simple linear regression models. From a mathematical perspective, *Universal Approximation Theorems* guarantee that any sufficiently smooth function, regardless of input and output dimensions, can be approximated with arbitrary accuracy by a neural network with a *single* hidden layer, dubbed *shallow neural network* [86–89]. The accuracy of this approximation is controlled by the number of neurons in the hidden layer. However, despite these theoretical results, it is important to emphasize that, in practical scenarios, the number of neurons in a shallow network cannot be increased indefinitely to achieve arbitrary accuracy. Approximating complicated functions with a shallow network may require an impractically large number of neurons, and scaling up the network can introduce significant optimization challenges due to the increasing complexity of the optimization landscape [90, 91].

To address these limitations, real-world applications often necessitate the use of multiple hidden layers, leading to the development of *deep neural networks* (DNNs) [90, 92–94]. The hierarchical structure of DNNs allows them to effectively capture the underlying structure of data. For example, in image recognition tasks, early layers learn *low-level* features, that are then combined into *higher-level*, more abstract features in the deeper layers [95]. This hierarchical learning approach is more natural and efficient than a flat, one-layer network attempting to learn all features simultaneously. A particularly interesting example was proposed by Eldan and Shamir [91], where they show that a simple function, which can be expressed by a small 3-layer feed-forward neural network, cannot be approximated by any 2-layer network beyond a certain constant accuracy unless the network's width is exponentially large in the input dimension. Their work formally proves that even a single additional layer can provide exponentially greater expressive power than merely increasing the width of a standard feed-forward neural network.

At the end, empirical evidences show that deep networks consistently outperform shallow networks across a wide range of tasks, including image classification [95, 96] and natural language processing [18, 97].

In the following sections, we will show practical examples related to quantum many-body systems that illustrate how performance improves as the parameters in a neural network are increased, considering both shallow and deep neural network architectures.

3.2 Neural Networks for representing Many-Body Wave Functions

In Chapter 1 we introduced the idea of using a variational parametrization $\Psi_\theta(\sigma)$ of the quantum state which allows for an efficient approximation of the exact ground state with polynomial resources. In 2017, Carleo and Troyer [7] proposed to use neural networks to parametrize $\Psi_\theta(\sigma)$, with the variational parameters θ optimized within the VMC framework (refer to Chapter 1 and Chapter 2). The mathematical foundations underlying this novel parametrization, known as *Neural-Network Quantum States* (NQS), are grounded on the representation theorems discussed in the previous section (see Sec. 3.1.1). In this context, the input of a NQS is typically the physical configuration σ, while the output is a single complex number $\Psi_\theta(\sigma) \in \mathbb{C}$. Common machine learning applications, such as image classification or natural language processing tasks, involves only the use of real valued-networks. This difference requires careful modifications in order to adapt standard machine learning architectures to handle with complex-valued outputs.

Over the years, NQS have attracted significant interest within the quantum many-body community. This interest is primarily due to two key features of NQS:

- Unlike physically inspired Ansatze, such as Gutzwiller-projected states [98], the accuracy of NQS can be systematically improved by increasing the number of parameters [7];

- NQS do not face the theoretical limitations that tensor network-based approaches often encounter when describing quantum states in more than one dimension [61, 99].

Notably, NQS have been successfully applied to some of the most challenging and traditionally unsolved problems in quantum many-body physics, achieving *state-of-the-art* results. Examples include the approximation of the ground state for two-dimensional frustrated spin [27, 36, 37], fermionic [100–103], and bosonic [104] systems; as well as the computation of excited states [37, 105–107] and spectral functions [108].

In the following sections, we will focus on two specific neural network architectures: a basic one, consisting of a single fully-connected layer, and a more advanced one based on the Transformer neural network [18]. We will test these architectures on the J_1-J_2 Heisenberg model on a chain. Specifically, we will discuss the properties of each neural network but more importantly their limitations.

3.2.1 Benchmark model : J_1-J_2 Heisenberg on a chain

In this Chapter, we will test the performances of different NQS on a non-trivial benchmark model. Specifically, we focus on to the one-dimensional J_1-J_2 Heisenberg model on finite clusters of N sites, imposing periodic boundary conditions. The Hamiltonian is defined by

$$\hat{H} = J_1 \sum_{R=1}^{N} \hat{\boldsymbol{S}}_R \cdot \hat{\boldsymbol{S}}_{R+1} + J_2 \sum_{R=1}^{N} \hat{\boldsymbol{S}}_R \cdot \hat{\boldsymbol{S}}_{R+2} \tag{3.3}$$

where $\hat{\boldsymbol{S}}_R = (S^x_R, S^y_R, S^z_R)$ is the $S = 1/2$ spin operator at site R and $J_1 > 0$ and $J_2 \geq 0$ are nearest- and next-nearest-neighbor antiferromagnetic couplings, respectively. Its phase diagram is well established by analytical and numerical studies [109]. For small values of J_2/J_1, the ground state has power-law spin-spin correlations and the excitation spectrum is gapless; for large values of J_2/J_1, the ground state is two-fold degenerate, leading to long-range dimer order (but exponentially decaying spin-spin correlations), and the spectrum is fully gapped. These two phases are separated by a critical point at $(J_2/J_1)_c = 0.24116(7)$ [54, 110]. Interestingly, for $J_2/J_1 > 0.5$, incommensurate (but short-range) spin-spin correlations have been found, whereas dimer–dimer correlations are always commensurate.

We emphasize that this model is frustrated, meaning that the amplitudes of the ground state do not have a definite sign in the computational basis. Specifically, the ground state of this model exhibits a non-trivial sign structure, particularly for $J_2/J_1 > 0.5$. Consequently, it necessitates the use of complex-valued neural networks, as we will discuss in the following sections.

3.3 Basic architectures : Shallow neural networks

In this section, we discuss how to define a wave function using the simplest neural network architecture: a fully-connected network with a single layer, namely a *shallow* neural network. Specifically, generalizing the structure of neural networks used in machine learning to take into account the necessity to learn the sign structure of the wave function. We begin by introducing a method to approximate classical probability distributions and then generalize this approach to the case of wave functions [38].

3.3.1 Parametrization of probability distributions

A class of powerful energy-based models called *Restricted Boltzmann Machines* (RBMs) has been widely employed in the context of machine learning to obtain accurate approximations of probability distributions [89]. Here, we give a brief introduction to this class of neural networks. Let us consider the case of a set of N binary variables that can take values ± 1, this case will be relevant for quantum $S = 1/2$ spin models, $\sigma = (\sigma_1, \ldots, \sigma_N)$, distributed according to a certain probability distribution $P_0(\sigma)$. In order to define the RBM probability distribution $P_{\mathrm{RBM}}(\sigma)$, we introduce an auxiliary set of K binary (*hidden*) variables $h = (h_1, \ldots, h_K)$, which are coupled to the physical variables in the energy function [89]

$$E_{\mathrm{RBM}}(\sigma, h; \theta) = -\sum_{i=1}^{N} a_i \sigma_i - \sum_{\mu=1}^{K} b_\mu h_\mu - \sum_{i=1}^{N} \sum_{\mu=1}^{K} \sigma_i w_{i,\mu} h_\mu. \tag{3.4}$$

The parameters $\boldsymbol{w}$ entering the above expression are called *weights*, while $\boldsymbol{b}$ and $\boldsymbol{a}$ are the so-called *hidden* and *input biases*, respectively; the set of all parameters is denoted in a compact form as $\theta = \{\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{a}\}$. The probability $P_{\mathrm{RBM}}(\sigma, \theta)$ is defined by tracing out the hidden variables h from the Boltzmann distribution of the RBM model, i.e., $P_{\mathrm{RBM}}(\sigma; \theta) \propto \sum_{\{h\}} \exp\{-E_{\mathrm{RBM}}(\sigma, h; \theta)\}$. Due to the absence of a direct coupling between hidden variables in E_{RBM} [see Eq. (3.4)], the trace can be performed analytically, giving:

$$P_{\text{RBM}}(\sigma;\theta) \propto \exp\left\{\sum_{i=1}^{N} a_i\sigma_i + \sum_{\mu=1}^{K} \log\left[\cosh\left(b_\mu + \sum_{i=1}^{N} w_{i,\mu}\sigma_i\right)\right]\right\}. \tag{3.5}$$

The result of this construction is a probability distribution function with non-trivial correlations between physical variables, parametrized by the set of parameters θ. For a fixed number N of physical variables, the representational power of the RBM probability distribution increases with the number of hidden variables K (or, equivalently, with the *complexity* parameter $\alpha = K/N$). The theoretical foundation of RBM models lies in the fact that they are universal approximators of probability distributions for sufficiently large values of K [111, 112]. Indeed, by a suitable definition of a *loss function*, the parameters θ of the RBM model can be tuned such that $P_{\text{RBM}}(\sigma;\theta)$ approximates the target distribution function $P_0(\sigma)$.

3.3.2 Parametrization of wave functions

Recently, RBMs have been used as variational wave functions to approximate the ground state of quantum many-body systems [7]. In contrast to probability distributions, quantum states are in general complex functions, i.e., their amplitudes in the computational basis are complex-valued. Therefore, a standard RBM parametrization making use of the $P_{\text{RBM}}(\sigma;\theta) \geq 0$ function discussed above is suitable only for those cases where the wave function is known to be real and positive definite in the computational basis (e.g., in bosonic systems). For all other cases, a generalization of the above construction is required.

For time-reversal symmetric models, the amplitudes of the ground-state wave function can be chosen to be real ($\langle\sigma|\Psi_0\rangle \in \mathbb{R}$), but their signs are not known in general. Representing the sign structure of the wave function with a real-valued parametrization is a difficult task, which requires the treatment of non-differentiable quantities or the use of gradient-free methods for the optimization [113]. For this reason, it is often convenient to adopt a complex-valued parametrization of the wave function. In this regard, two alternative formulations are presented in the following.

As a first possibility, we can employ two (independent) RBM probability functions, one for the modulus $P_{\text{RBM}}(\sigma;\theta_m)$ and one for the phase $P_{\text{RBM}}(\sigma;\theta_p)$ of the wave function [114].

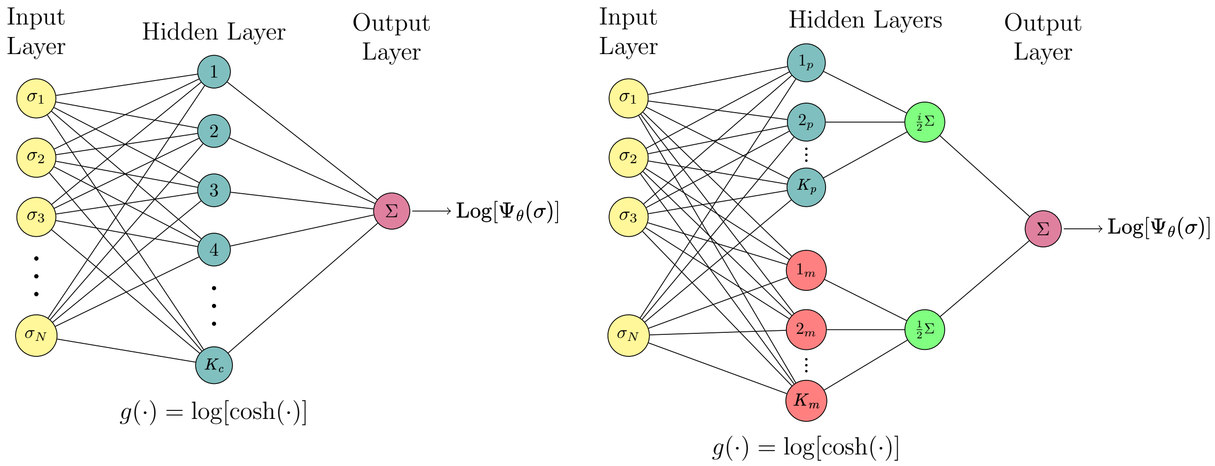

Figure 3.2: Schematic illustration of the feed-forward neural networks representation of the cRBM state of Eq. (3.8) (left panel) and pmRBM state of Eq. (3.6) (right panel). The cRBM *Ansatz* has complex-valued parameters; instead, the pmRBM state has real parameters. In both networks, the activation function of the hidden neurons is $g(\cdot) = \log\cosh(\cdot)$.

The amplitudes of the quantum state are then given by:

$$\text{Log}[\Psi_{\text{pmRBM}}(\sigma; \theta_m, \theta_p)] = \frac{1}{2}\log[P_{\text{RBM}}(\sigma; \theta_m)] + \frac{i}{2}\log[P_{\text{RBM}}(\sigma; \theta_p)]. \tag{3.6}$$

Here, the parameters of the RBMs, i.e., θ_m and θ_p, are all real. The structure of the variational state is characterized by the number of hidden variables for the modulus K_m and the phase K_p, giving the total number of hidden units being $K = K_p + K_m$. The complexity of the network is defined as the ratio between the number of hidden variables and visible ones, leading to:

$$\alpha_m = K_m/N \qquad \alpha_p = K_p/N\ . \tag{3.7}$$

We emphasize that a different number of hidden variables can be taken for the modulus and the phase. This variational *Ansatz* is dubbed *phase-modulus RBM* (pmRBM) wave function.

The second option is taking a single RBM with complex parameters, in order to provide a complete description of both amplitude and phase of the wave function with a single complex-valued network [7]:

$$\text{Log}[\Psi_{\text{cRBM}}(\sigma; \theta_c)] = \sum_{i=1}^{N} a_i\sigma_i + \sum_{\mu=1}^{K_c} \log\cosh\left(b_\mu + \sum_{i=1}^{N} w_{i,\mu}\sigma_i\right). \tag{3.8}$$

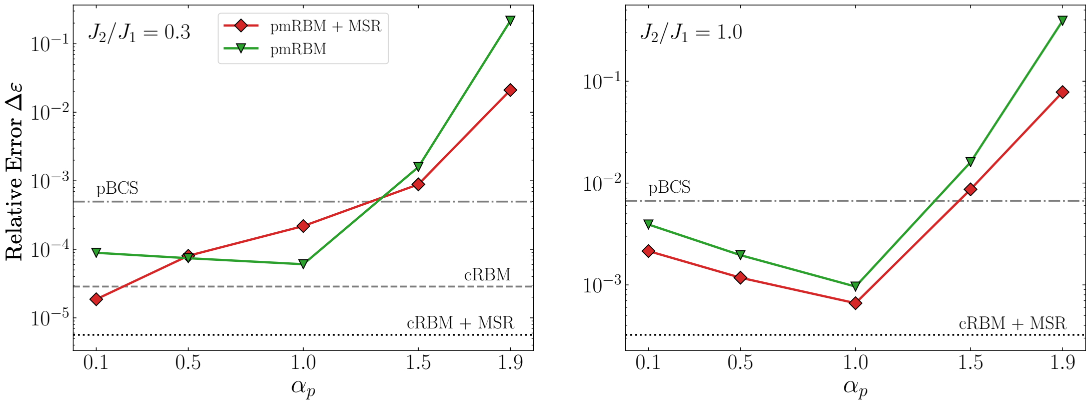

Figure 3.3: Accuracy of the variational energy for the J_1-J_2 Heisenberg model with $N = 20$ sites, for $J_2/J_1 = 0.3$ (left panel) and $J_2/J_1 = 1$ (right panel). The pmRBM *Ansatz* of Eq. (3.6) is reported as a function of α_p, with $\alpha_m + \alpha_p = 2$. The results for the cRBM wave function of Eq. (3.8) are reported for $\alpha_c = 1$, such that the total number of real parameters (840) is the same as for the pmRBM state. The results obtained by including the Marshall-sign rule are also shown. For $J_2/J_1 = 1$ (right panel), the accuracy of the cRBM with and without the Marshall-sign rule do not differ, thus we included only the former one in the plot. In both panels the results obtained by pBCS states are shown for comparison.

Here, $\theta_c \in \mathbb{C}$ and the number of hidden variables is K_c corresponding to a complexity given by:

$$\alpha_c = K_c/N \; . \tag{3.9}$$

This state is dubbed *complex RBM* (cRBM) wave function. In the following, we set input biases equal to zero ($a_i = 0$) in both phase-modulus and complex RBMs [37, 106, 115].

The variational wave functions defined in Eq. (3.6) and Eq. (3.8) can be seen as feed-forward neural networks [83] with a visible layer of N neurons that represent the physical configuration σ, one hidden layer of neurons with activation function $g(\cdot) = \log\cosh(\cdot)$, and one output neuron which performs the sum of the outputs of the hidden layer and returns the logarithm of the amplitude (see Fig. 3.2).

3.3.3 Numerical Results

In the following sections we perform a systematic study on the J_1-J_2 Heisenberg model on a chain (see Sec. 3.2.1) on small clusters in which we compare the variational results achieved by RBMs with exact quantities, computed by Lanczos diagonalization [54]. Additionally, a comparison with the variational results obtained by projected fermionic states (denoted as pBCS) is reported [116].

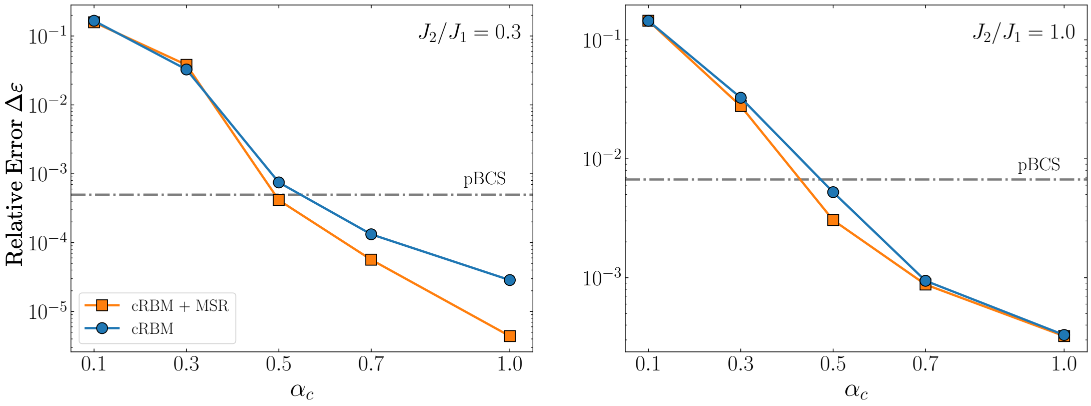

Figure 3.4: Accuracy of the variational energy for the J_1-J_2 Heisenberg model with $N = 20$ sites, for $J_2/J_1 = 0.3$ (left panel) and $J_2/J_1 = 1$ (right panel). The results for the cRBM of Eq. (3.8) are reported as a function of α_c, with and without including the Marshall signs. The accuracy of the pBCS state is also shown for comparison.

The optimization of the variational parameters can be implemented within stochastic approaches (see Chapter 1). Here, an optimization step is made by $O(10^3)$ Monte Carlo samples, each of which consists in $O(N)$ Metropolis moves (two-spin flips); variational parameters are updated at the end of every optimization step by using the Stochastic Reconfiguration algorithm [69] (see Chapter 2). In all the calculations, we make use of the the RBM wave function symmetrized with respect to translations (see Appendix B).

3.3.3.1 Accuracy of the ground-state wave function

Let us start by comparing pmRBM and cRBM *Ansätze* on a cluster with $N = 20$ sites, for which exact results can be obtained by Lanczos diagonalization. Two values of the frustrating ratio are considered, $J_2/J_1 = 0.3$ and $J_2/J_1 = 1$, corresponding to cases in which the Marshall-sign rule[6] gives good and poor approximations of the exact sign structure. In Fig. 3.3, we report the accuracy obtained by the pmRBM wave function in Eq. (3.6) for different values of α_p [see Eq. (3.7)], by plotting the relative error of the variational energy with respect to the exact one, namely $\Delta\varepsilon = |\left(E_0 - E_{\text{var}}\right)/E_0|$ where E_0 and E_{var} are the exact and variational energies, respectively. We choose to consider $\alpha_m + \alpha_p = 2$, in order to fix the total number of variational parameters. The results for the cRBM state in Eq. (3.8) with the same number of parameters, i.e., $\alpha_c = 1$ [see

[6] See Appendix A for a detailed discussion about the Marshall-sign rule.

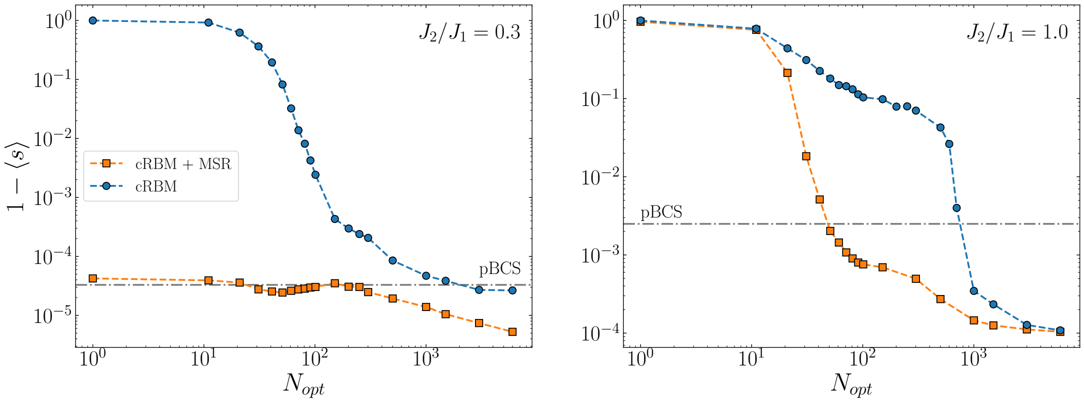

Figure 3.5: Evolution of the average sign $\langle s \rangle$ defined in Eq. (3.10) along the optimization procedure of the cRBM state, for $J_2/J_1 = 0.3$ (left panel) and $J_2/J_1 = 1$ (right panel). Here, for each optimization step N_{opt}, $\langle s \rangle$ is computed (exactly) for the corresponding variational parameters.

Eq. (3.9)], are reported. In both cases, calculations attaching the Marshall-sign rule to the wave-function amplitudes are also considered. Without including Marshall signs, the best energy of the pmRBM state is obtained for $\alpha_p \approx 1$, for both $J_2/J_1 = 0.3$ and $J_2/J_1 = 1$. This means that taking the same number of variational parameters for the modulus and the phase represents the best strategy for this kind of wave function. By contrast, when including the Marshall signs, a different behavior occurs for the two values of the frustrating ratio. For $J_2/J_1 = 0.3$, where the Marshall signs represent an excellent approximation of the exact ones, the best energy of the pmRBM *Ansatz* is obtained for $\alpha_p \ll 1$; instead, for $J_2/J_1 = 1$, the optimal energy is still obtained when $\alpha_p \approx 1$. Still, the lowest variational energies in Fig. 3.3 are those of the cRBM state. For this state, the inclusion of the Marshall-sign rule provides a substantial energy gain at $J_2/J_1 = 0.3$, while being almost ineffective for the accuracy at $J_2/J_1 = 1$. A consistent improvement with respect to pBCS wave functions [116] is achieved, even though the latter variational states require a significantly smaller number of variational parameters. In particular, for $J_2/J_1 = 0.3$ the energy accuracy of the cRBM is almost three orders of magnitude better than the pBCS *Ansatz*. Having certified the better accuracy of the cRBM wave function with respect to the pmRBM state, we choose to stick to the former architecture. In Fig. 3.4 we report the accuracy of the cRBM *Ansatz* when varying the network complexity α_c [see Eq. (3.9)]. The inclusion of the Marshall-sign rule proves to be particularly effective for $J_2/J_1 = 0.3$ and $\alpha_c \gtrsim 0.5$, while being less relevant for $J_2/J_1 = 1$.

Now, let us define a measure of the difference between the phases of the cRBM wave function and the signs of the exact ground state, namely

$$\langle s \rangle = \left| \sum_{\{\sigma\}} |\Psi_0(\sigma)|^2 \text{sign}[\Psi_0(\sigma)] e^{i\Theta_{\text{cRBM}}(\sigma)} \right|, \tag{3.10}$$

where $\Theta_{\text{cRBM}}(\sigma) = \arg\left[\langle \sigma | \Psi_{\text{cRBM}} \rangle\right]$. The absolute value is taken to overcome a possible global phase in the cRBM state. Then, $\langle s \rangle = 1$ whenever the phases (but not necessarily the moduli) of the cRBM state match the exact values. In Fig. 3.5, we track this quantity along the optimization procedure of the variational parameters, for the cases with and without the Marshall-sign rule. An evident speed-up in the convergence of the above quantity is observed when the Marshall sign structure is included, even for the case with $J_2/J_1 = 1$, for which, at the end of the simulation, no substantial energy gain is obtained by the addition of Marhsall signs.

Another instructive analysis of the learning process of the cRBM wave function is achieved by tracking the evolution of $\Theta_{\text{cRBM}}(\sigma)$ during the optimization procedure, computing it for the various spin configurations σ visited along the Monte Carlo simulation. As a benchmark, it is particularly insightful to consider the case with $J_2 = 0$, where the sign structure of the exact result is given by the Marshall-sign rule. In additon, the case with $J_2/J_1 = 1$, where the Marshall-sign rule is heavily violated, is also considered. For both cases, the values of $\Theta_0(\sigma) = \arg\left[\langle \sigma | \Psi_0 \rangle\right]$ are either 0 or π, since the exact ground state is a real-valued wave function. The evolution of $\Theta_{\text{cRBM}}(\sigma)$ during optimizations is shown in panel (a) and (b) of Fig. 3.6, where blue (red) points indicate configurations for which the exact phase is $\Theta_0(\sigma) = 0$ $[\Theta_0(\sigma) = \pi]$. After an initial transient, the values of $\Theta_{\text{cRBM}}(\sigma)$ quickly converge towards the exact values. This is particularly true for $J_2 = 0$, where $\Theta_{\text{cRBM}}(\sigma)$ approaches 0 or π with very small statistical fluctuations. A similar result is also obtained for $J_2/J_1 = 1$, even though larger fluctuations remain after convergence. It is interesting to remark that the exact signs are recovered only for the most relevant spin configurations (i.e., the ones with the largest weights), which are frequently visited in the Monte Carlo optimization, and contribute the most to the variational energy. This fact can be appreciated by looking at panel (c) and (d) of Fig. 3.6, where all the phases of the final cRBM state are shown as a function of the exact weights $|\Psi_0(\sigma)|^2$ of the corresponding spin configurations.

The results of the average sign $\langle s \rangle$ in Eq. (3.10), together with the ones for the overlap between the exact ground state and the best-energy cRBM *Ansatz* $|\langle \Psi_0 | \Psi_{\text{cRBM}} \rangle|$, are

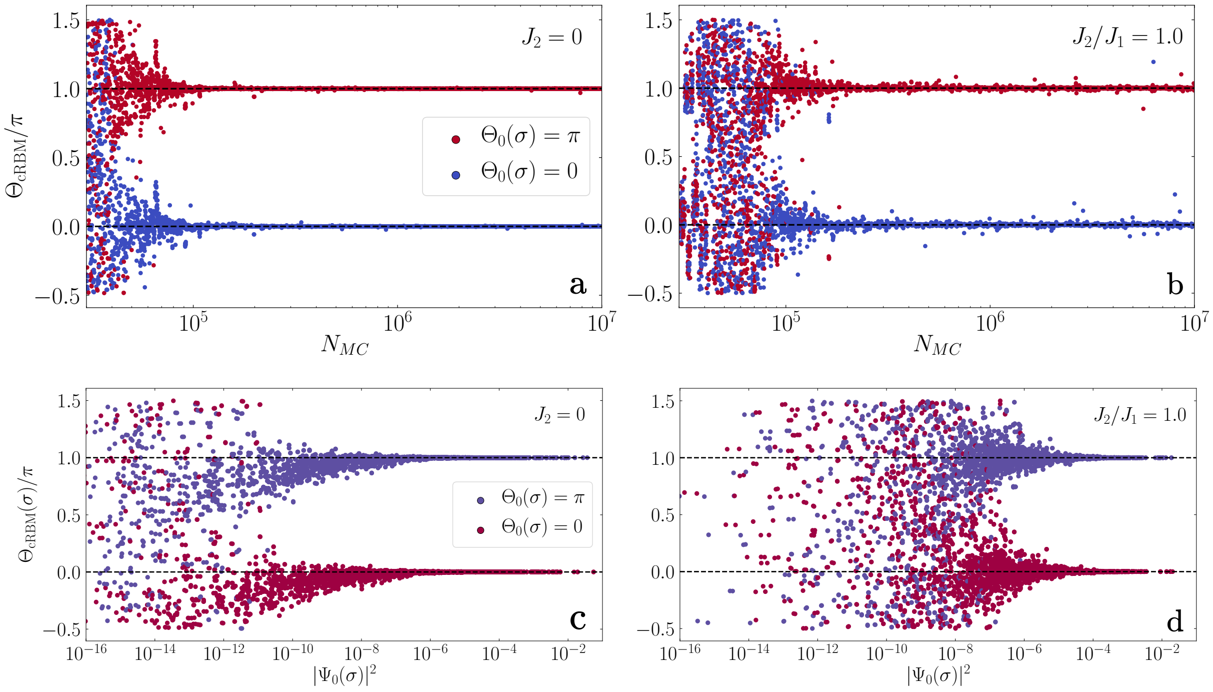

Figure 3.6: Evolution of the phases $\Theta_{\rm cRBM}(\sigma)$ for the cRBM wave function with $\alpha_c = 1$ along the Monte Carlo optimization for $J_2 = 0$ (panel **a**) and $J_2/J_1 = 1$ (panel **b**). The colors of the dots denote the phase of the exact ground state wave function. The number of sites is $N = 20$. The results are plotted as a function of the number of Monte Carlo steps $N_{\rm MC}$ and the variational parameters are updated every 10^3 steps. In panels **c** and **d** we show $\Theta_{\rm cRBM}(\sigma)$ phases of the optimal cRBM state with $\alpha_c = 1$ for all the spin configurations with $S^z_{\rm tot} = 0$ and momentum $k = 0$ (for $N = 20$ sites). The phases are plotted as a function of the exact weight $|\Psi_0(\sigma)|^2$ of the configurations. Results for $J_2 = 0$ (panel **c**) and $J_2/J_1 = 1$ (panel **d**) are shown. The colors of the dots denote the phase of the exact ground-state wave function.

reported in Fig. 3.7 for different values of J_2/J_1 ($N = 20$ sites). A comparison with the results of the pBCS wave functions is also shown. We emphasize that the complex RBM always gives a better approximation of the exact ground state than the pBCS states, especially for $J_2/J_1 > 0.5$.

3.3.3.2 Spin-spin correlation functions

For each component $\nu = x$, y, and z of the spin operator, we consider the expectation value of the spin-spin correlations in real space:

$$C^{\nu\nu}(r) = \frac{1}{N}\sum_R \langle \hat{S}^\nu_R \hat{S}^\nu_{R+r} \rangle, \tag{3.11}$$

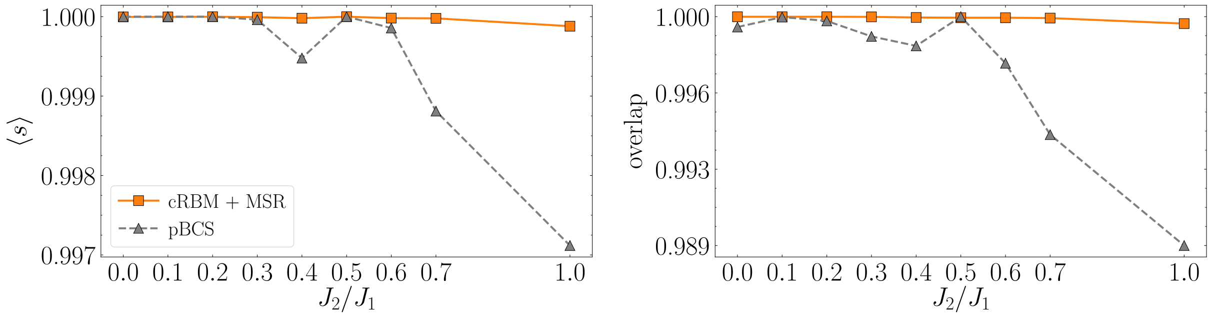

Figure 3.7: Average sign $\langle s \rangle$ defined in Eq. (3.10) (left panel) and overlap $|\langle \Psi_0 | \Psi_{\rm cRBM} \rangle|$ (right panel) for the best-energy cRBM *Ansatz* with $\alpha_c = 1$ as a function of the frustrating ratio J_2/J_1. The results of the pBCS state are also reported for comparison. The number of sites is $N = 20$.

and its Fourier transform in momentum space:

$$S^{\nu\nu}(k) = \sum_r e^{ikr} C^{\nu\nu}(r). \tag{3.12}$$

Since the RBM *Ansatz* is a function of the z-component of the spins only, it explicitly breaks the spin $SU(2)$ symmetry, leading to a difference between the z axis and the x-y plane. However, by using a large number of variational parameters, it is possible to reduce this anisotropy and obtain almost $SU(2)$ symmetric results. In Fig. 3.8, we report the relative error of the $C^{zz}(r)$ and $C^{xy}(r) = [C^{xx}(r) + C^{yy}(r)]/2$ of the cRBM state with respect to the exact spin-spin correlations, for $J_2/J_1 = 0.3$ and $J_2/J_1 = 1$ (for $N = 20$ sites). By increasing the network complexity α_c, the accuracy strongly improves and, consequently, also the anisotropy decreases. The pBCS wave function has $SU(2)$ symmetry by construction and is reported for comparison. Still, its accuracy is about one order of magnitude worse than the one obtained by the best cRBM with $\alpha_c = 1$. However, it is worth remarking that the number of variational parameters is considerably different for the two classes of wave functions, with the pBCS state requiring a maximum of 6 parameters, against the 840 parameters of the cRBM state.

Given the tiny residual anisotropy of the cRBM *Ansatz*, we report in Fig. 3.9 the results for $C^{zz}(r)$ and $S^{zz}(k)$ for three representative values of the frustrating ratio, namely $J_2 = 0$ (gapless regime), $J_2/J_1 = 0.3$ (gapped regime, with commensurate spin-spin correlations), and $J_2/J_1 = 1$ (gapped regime, with incommensurate spin-spin correlations). These calculations confirm the excellent degree of approximation obtained by cRBM in all regimes. Indeed, even though the pBCS *Ansatz* also gives remarkably accurate results,

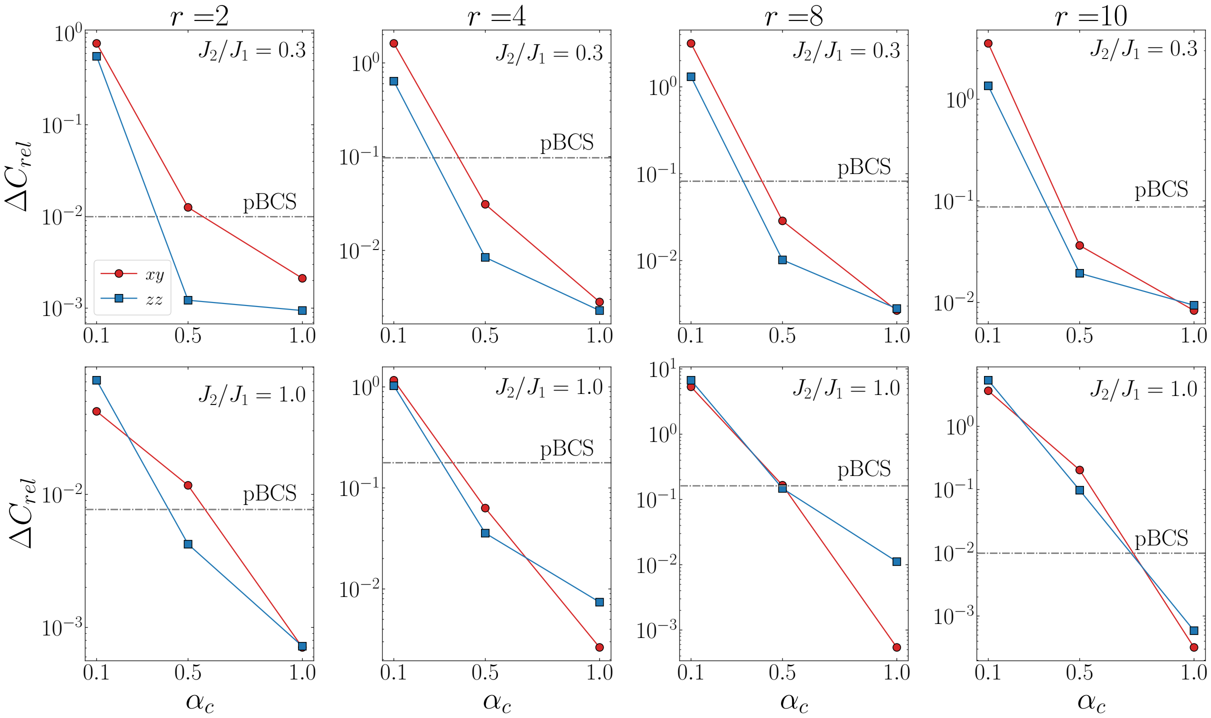

Figure 3.8: Relative error of the spin-spin correlation functions of the cRBM state with respect to the exact values, for $J_2/J_1 = 0.3$ (upper panels) and $J_2/J_1 = 1$ (lower panels). Results for $C^{zz}(r)$ (blue squares) and $C^{xy}(r)$ (red circles) are shown as a function of α_c, for several distances r on a $N = 20$ sites chain. The results of the (spin-isotropic) pBCS wave function are also reported for comparison.

the complex RBM is able to perfectly reproduce even the most challenging case with $J_2/J_1 = 1$, e.g., where the peak of $S^{zz}(k)$ is close to $k = \pi/2$.

3.3.3.3 Excited states

We finally report the calculations of excited states at finite momenta. Indeed, by using translational symmetry, it is possible to fix the momentum k of the variational *Ansatz* in the cRBM state (see Appendix B). In order to target the lowest-energy *triplet* excitation for each momentum, we restrict the wave function to the sector of the Hilbert space with $S^z_{\text{tot}} = 1$. The variational gaps for the lowest-lying triplets are shown in Fig. 3.10 for two values of the frustrating ratio in the gapped phase, $J_2/J_1 = 0.45$ and $J_2/J_1 = 1$. The results for the gapless regime $J_2 = 0$ are perfectly compatible with the ones shown in Refs. [105, 115], and are thus not reported. The comparison of the variational energies

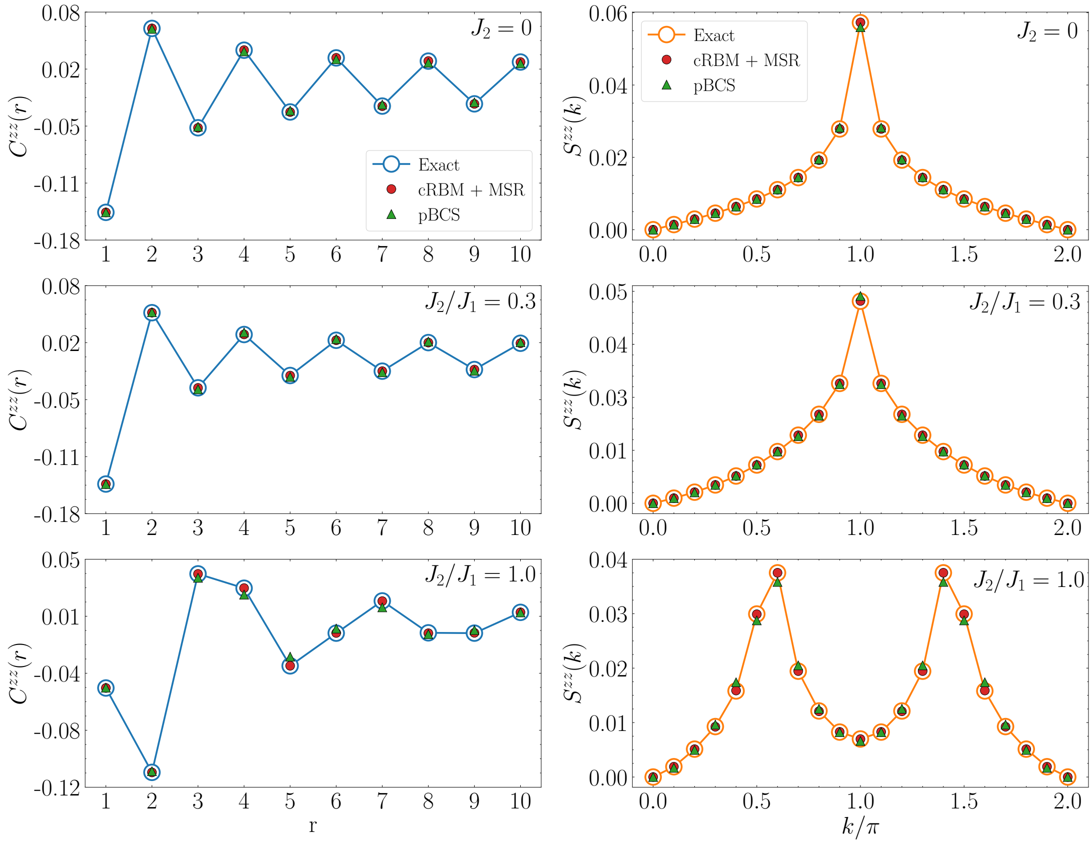

Figure 3.9: Spin-spin correlation function $C^{zz}(r)$ (left panels) and $S^{zz}(k)$ (right panels) for different values of J_2/J_1 and $N = 20$ sites. Results for the best cRBM wave function with $\alpha_c = 1$ (full circles), the pBCS state (full triangles), and the exact ground state (empty circles) are reported.

to the exact values confirm the high accuracy of the cRBM to reproduce not only the ground-state properties, but also low-energy states.

3.3.3.4 Limitations

In the previous sections, we demonstrated the ability of RBM wave functions to reproduce the ground state of a frustrated spin model in one dimension, where the sign structure can be highly non-trivial (e.g., completely different from the one given by the Marshall-sign rule). The accuracy is not limited to the ground-state energy but extends to the lowest-energy triplet excitations. However, the number of variational parameters grows as $O\left(\alpha N^2\right)$ where N is the number of sites and α the complexity of the network. Hence,

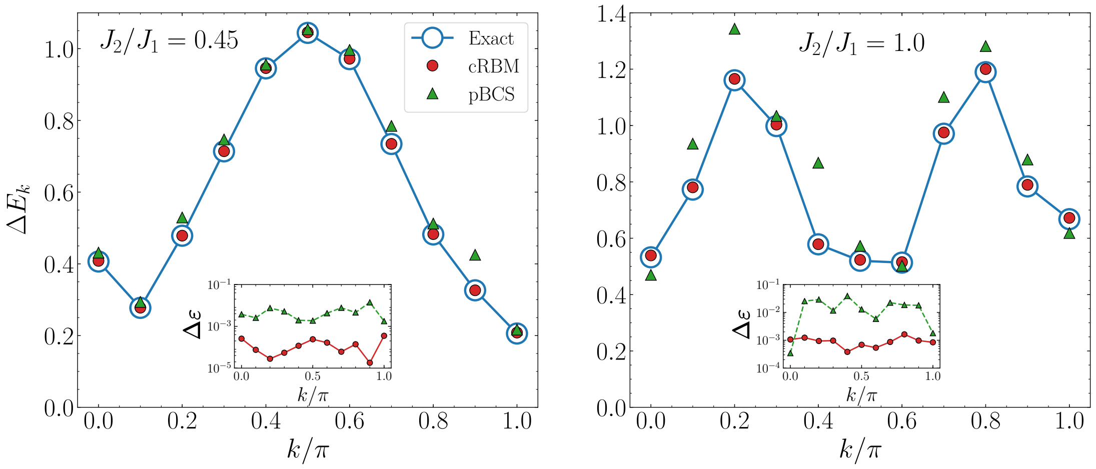

Figure 3.10: Lowest-energy triplet excitation of the J_1-J_2 Heisenberg model for $J_2/J_1 = 0.45$ (left panel) and $J_2/J_1 = 1$ (right panel), for a chain of $N = 20$ sites. ΔE_k is the difference between the lowest triplet energy at momentum k and the ground state energy. Results obtained by the best cRBM *Ansatz* with $\alpha_c = 1$ (full circles) and the pBCS state (full triangles) are shown, together with exact values (empty circles). The insets show the relative error of the variational results, i.e., $\Delta\varepsilon = |\left(E_{\text{ex,k}} - E_{\text{var,k}}\right)/\text{E}_{\text{ex,k}}|$, where $E_{\text{ex,k}}$ and $E_{\text{var,k}}$ are the exact and variational energies of the excited states, respectively.

the optimization of the variational wave function becomes very difficult for large lattices. We emphasize the fact that, due to the fully-connected structure of the network, the transferability of the parameters when increasing the size is not possible for RBMs. By contrast, pBCS wave functions have very few variational parameters (independently on the number of spins N), whose optimal values rapidly converge when increasing the system size. Thus, the results of numerical optimizations on smaller system sizes often provide an excellent starting point for optimizations on larger lattices. Calculations with $N = 10$, 20, and 30 sites exemplify the issue of size consistency. In Fig. 3.11, we show the results for the relative error of the variational energy for $J_2/J_1 = 0.3$ and $J_2/J_1 = 1$ (fixing the complexity at $\alpha_c = 1$). While the accuracy of the pBCS is lower than that of cRBMs for all sizes, pBCS states are size-consistent with very good approximation; by contrast, cRBMs with fixed complexity slightly lose accuracy when increasing the system size. As a consequence, an increase of complexity with the system size could be necessary to obtain size-consistent results. An additional remark deals with the physical interpretation of the variational states. Indeed, Gutzwiller-projected fermionic states have a transparent physical interpretation, providing a clear physical description of the phases of the system,

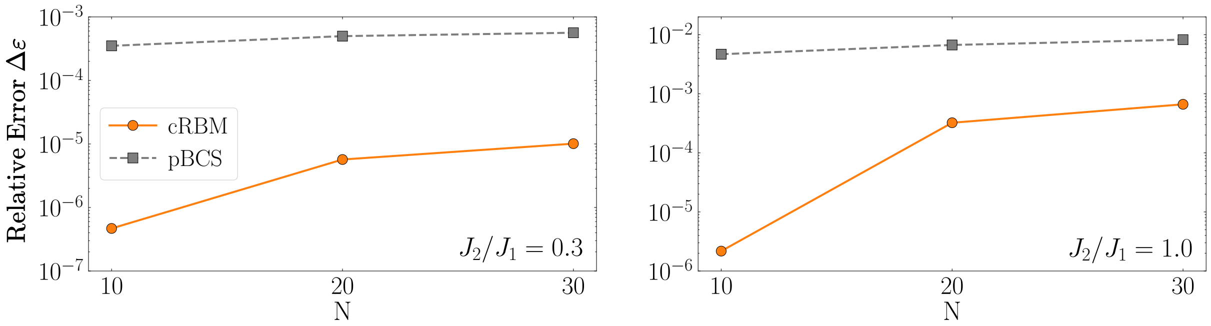

Figure 3.11: Size scaling of the relative error of the variational energy for the best-energy cRBM *Ansatz* with $\alpha_c = 1$. The results of the pBCS state are also reported for comparison. Calculations are done for $J_2/J_1 = 0.3$ (left panel) and $J_2/J_1 = 1$ (right panel) and $N = 10$, 20, and 30 sites.

even without computing correlation functions and observables. By contrast, RBM states lack of a physical interpretability of their variational parameters. One possible strategy to simplify the optimization and favor a size consistent behavior could be combining it with Gutzwiller-projected wave functions, e.g., using the RBM as correlator (a generalization of the standard Jastrow factor). A few works have taken this direction [16, 17, 37], showing that with this hybrid approach it is possible to obtain very accurate results also increasing the size of the system. Our aim, however, is to improve neural network architectures without combining them with other types of variational states. In the following sections, we will explore more complicated architectures that have been developed in the field of machine learning and adapt them specifically for this type of problems.

3.4 Advanced architectures : Transformer neural networks

In the last few years, the Transformer architecture [18] has emerged as the state-of-art choice in natural-language processing tasks. Its key feature is the ability to model relationships among all elements of an input sequence (regardless of their positions), by efficiently *transforming* input sequences into abstract representations. Specifically, Transformers map a set of input vectors within a given representation space into a new set of vectors, maintaining the same dimensionality but situated in a new representation space. This transformation is based on the idea that the new representation space is better suited for solving the target task. Although the Transformer architecture may appear complicated, with multiple components working together, in this section we will provide an intuitive explanations to clarify and motivate the design of its various elements.

3.4.1 Multi-Head Attention Mechanism

The fundamental ingredient which characterizes the Transformer architecture is the *attention mechanism*. To illustrate the idea at the basis of this mechanism, we report a nice example from Ref. [117]. Consider the following two sentences:

1. I swam across the river to get to the other bank.

2. I walked across the road to get cash from the bank.

Clearly, the meaning of the word "bank" differs between the two sentences. However, correctly interpreting it requires considering the other words of the sentence. In sentence (1), the words "swam" and "river" strongly indicate that "bank" refers to the side of a river. Conversely, in sentence (2), the word "cash" suggests that "bank" refers to a financial institution. Generally, the correct interpretation of a word depends on the context provided by the other words of the sentence, specifically some words are more relevant than others to determine the right meaning. In standard neural networks, the output of a given input is influenced by the weights that multiply those inputs. However, once the network is trained these weights are fixed. The core idea of the attention mechanism is to use weights that dynamically depend on the input data to efficiently distinguish the different meaning of a given word. For instance, through the attention mechanism the Transformer should be able to map the word "bank" to different points in the new representation space depending on the context provided by the two different sentences.

At this point we have to formulate in mathematical terms a mechanism that exhibits the features we have just discussed. Consider a set of $\mathcal{N}$ input vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\mathcal{N}}\}$, each of dimensionality d. In practice, each vector corresponds to an element of the input sequence, such as a word within a sentence, a patch within an image (see Sec. 3.4.3), or an amino acid within a protein. These vectors are typically constructed using an *embedding* procedure, which maps the original elements of the input sequence into vectors of real numbers, making them suitable for manipulation through mathematical operations [117]. The attention mechanism transforms the original set of vectors in another set of vectors $\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{\mathcal{N}}\}$, dubbed *attention vectors*, of the same dimensionality but in a new representation space. In this space, each new vector encodes information from all the input vectors. Specifically, the value of each attention vector $\boldsymbol{A}_i$ depends not only on the corresponding input vector $\boldsymbol{x}_i$, but on the entire set of input vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\mathcal{N}}\}$. The simplest way to achieve this property is to define:

$$\boldsymbol{A}_i = \sum_{j=1}^{\mathcal{N}} \alpha_{ij} \boldsymbol{x}_j \ . \tag{3.13}$$

By adjusting the values of the *attention weights* α_{ij}, we can encode the fact that certain inputs are more influential in determining the transformed representation of $\boldsymbol{A}_i$. For example, in sentence (1) from the previous example, the vectors corresponding to the words “swam” and “river” should have large attention weights in the construction of the new vector associated with the word “bank”.

While the attention vectors defined in Eq. (3.13) possess the correct properties, we can improve their definition by carefully defining the functional form of the coefficient α_{ij} and introducing other variational parameters in order to enhance flexibility. To give an intuition for the specific choices we consider another example from Ref. [117]. Consider constructing a catalog of movies for an online movie streaming service. Each movie can be associate to a vector encoding all its attributes (genre, the names of the leading actors, the length of the movie, etc.), called *key* vector. The movie file itself is dubbed *value*. The user provides a personal vector of desired attributes, defined as the *query* vector. The streaming service compares the query vector with all the key vectors to find the best match and suggest the corresponding movie to the user in the form of the value file. Mathematically, this setup can be translated into the introduction of *value* $\boldsymbol{v}_i$, *query* $\boldsymbol{q}_i$, and *key* $\boldsymbol{k}_i$ vectors, each defined as a linear transformation of the input vectors $\boldsymbol{x}_i$:

$$\boldsymbol{v}_i = V\boldsymbol{x}_i \qquad \boldsymbol{q}_i = Q\boldsymbol{x}_i \qquad \boldsymbol{k}_i = K\boldsymbol{x}_i \ . \tag{3.14}$$

Here, the matrices V, Q, and K represent parameters that are optimized during the training of the full Transformer architecture. These matrices have dimensionality $d \times d$, ensuring that the output representation maintains the same dimensionality as the input.

To determine how relevant the vector $\boldsymbol{x}_j$ is for the new representation of the vector $\boldsymbol{x}_i$, we measure the similarity between these vectors. A simple measure of similarity is to take their dot product between their corresponding key and query vectors, namely $\boldsymbol{q}_i^T \boldsymbol{k}_j$. However, in practice it is useful to constrain the attention weights to be non-negative

($\alpha_{ij} \geq 0$) and to sum to unity ($\sum_{j=1}^{\mathcal{N}} \alpha_{ij} = 1$). These constraints can be imposed by defining the coefficients α_{ij} by using the softmax function to transform the dot products[7]:

$$\alpha(\boldsymbol{q}_i, \boldsymbol{k}_j) = \frac{\exp\left(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d}}\right)}{\sum_{k=1}^{\mathcal{N}} \exp\left(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_k}{\sqrt{d}}\right)} = \text{softmax}\left(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d}}\right) \ . \tag{3.15}$$

With the functional form of the attention weights now defined, we can generalize the earlier definition of the attention vectors in Eq. (3.13) as follows:

$$\boldsymbol{A}_i = \sum_{j=1}^{\mathcal{N}} \alpha(\boldsymbol{q}_i, \boldsymbol{k}_j) \boldsymbol{v}_j \ , \tag{3.16}$$

where $\alpha(\boldsymbol{q}_i, \boldsymbol{k}_j)$ are the ones defined in Eq. (3.15) and we have replaced the input vectors $\boldsymbol{x}_i$ with the value vectors $\boldsymbol{v}_i$ defined in Eq. (3.14). This process, introduced by Vaswani et al. [18], is known as *scaled dot-product self-attention.* The term *self-attention* arises because the same sequence is used to determine the queries, keys, and values, while the similarity between query and key vectors is evaluated through a scaled dot product.

Notice that the matrices V, Q ank K are shared accross all the input vectors. As a result, the attention mechanism defined in Eq. (3.16) is *permutationally equivariant.* This means that permuting the order of the input vectors leads to the same permutation of the output vectors, consequently the representation learned by a transformer will be independent of the input vectors ordering. For example, the two sentences [117]:

1. The food was bad, not good at all.

2. The food was good, not bad at all.

contains the same words and, consequently, they correspond to the same set of input vectors. However, their meaning is different due to the word order. Therefore, it becomes necessary to inject positional information into the attention mechanism. In the original work of Vaswani et al. [18] they propose an approach known as *absolute positional encoding*, where a set of *position vectors* $\boldsymbol{r}_i$ associated with each input position, are combined with the corresponding input vector as $\boldsymbol{x}_i + \boldsymbol{r}_i$. Alternatively, positional information can

[7] The re-scaling of the product of the query and key vectors by a factor $\sqrt{d}$ is used to prevent exponentially small gradients due to the softmax function [18].

be encoded directly into the attention mechanism. Shaw et al. [118] introduce the *relative positional encoding* which allows to effectively capture the relative word orders:

$$\alpha_{ij}(\boldsymbol{q}_i, \boldsymbol{k}_j) = \frac{\exp\left(\frac{\boldsymbol{q}_i^T(\boldsymbol{k}_j + p_{i-j})}{\sqrt{d}}\right)}{\sum_{k=1}^{\mathcal{N}} \exp\left(\frac{\boldsymbol{q}_i^T(\boldsymbol{k}_j + p_{i-j})}{\sqrt{d}}\right)} \ , \tag{3.17}$$

where p_{i-j} is a vector of d learnable parameters that encode the relative position of the vectors. Following this work several methods have been proposed to improve the relative positional encoding. One well-known example is the T5 attention mechanism, discussed in Ref. [119].

Notice that the two main ingredients that characterize the attention mechanism are the semantic structure, captured by the dynamically dependence of the attention weights on the input vectors, and the positional structure, introduced through the dependence on the position of the input vectors. In certain applications, one of these features can be more relevant than the other. For instance, Cui et al. [120] develop a toy model where a phase transition between semantic and positional regimes can be observed, highlighting how these two aspects of the attention mechanism can dominate under different conditions.

The attention mechanism described above is referred to as a single *attention head*. However, the set of input vectors can contain different patters, using a single attention head to detect them can result in averaging these different features, thereby losing important information. To address this, several heads are employed simultaneously, each of them with independent learnable parameters. This approach is similar to the use of multiple channels in a convolutional layer and leads to the formulation of the *Multi-Head Attention Mechanism*:

$$\boldsymbol{A}_i^\mu = \sum_{j=1}^{\mathcal{N}} \alpha_{ij}^\mu(\boldsymbol{q}_i^\mu, \boldsymbol{k}_j^\mu)\boldsymbol{v}_j^\mu \ , \tag{3.18}$$

where $\mu = 1, \ldots, h$, with h the total number of *heads*, an hyperparameter of the architecture. Here, the queries, keys and values vectors are defined as:

$$\boldsymbol{v}_i^\mu = V^\mu \boldsymbol{x}_i \qquad \boldsymbol{q}_i^\mu = Q^\mu \boldsymbol{x}_i \qquad \boldsymbol{k}_i^\mu = K^\mu \boldsymbol{x}_i \ , \tag{3.19}$$

where the matrices Q^μ, K^μ, V^μ have dimension $d/h \times d$ for each head $\mu = 1, \ldots, h$. This results in a set of h intermediate vectors $\boldsymbol{A}_i^\mu$ of dimensionality d/h for each i. Then, this vectors are concatenated $\text{Concat}[\boldsymbol{A}_i^1, \ldots, \boldsymbol{A}_i^h]$ to form a set of $\mathcal{N}$ d-dimensional vectors. Finally, an additional linear transformation with a $d \times d$ matrix W is applied to each

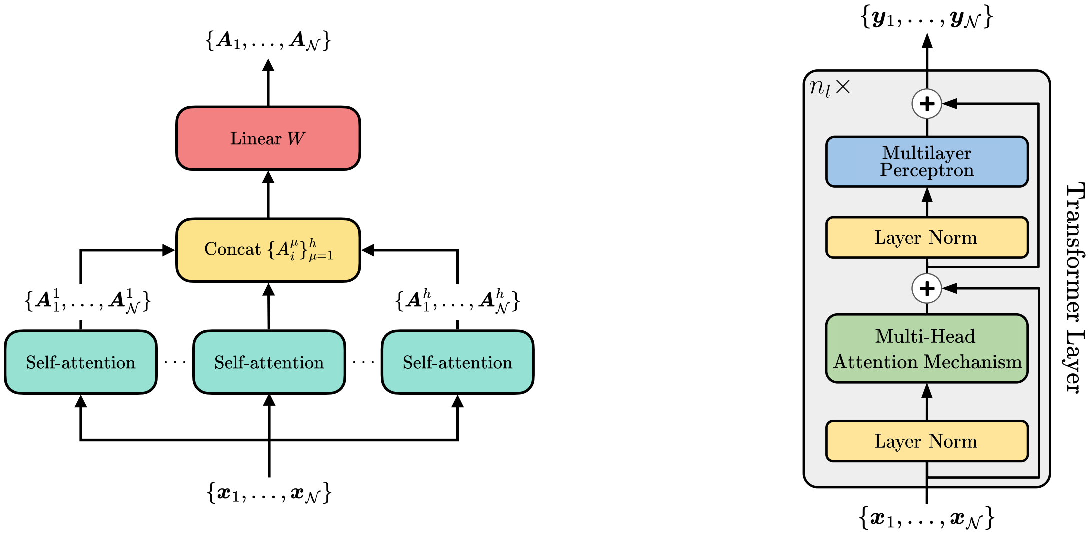
$\{\boldsymbol{A}_1, \dots, \boldsymbol{A}_{\mathcal{N}}\}$
Linear W
Concat $\{A_i^\mu\}_{\mu=1}^h$
$\{\boldsymbol{A}_1^1, \dots, \boldsymbol{A}_{\mathcal{N}}^1\}$
$\{\boldsymbol{A}_1^h, \dots, \boldsymbol{A}_{\mathcal{N}}^h\}$
Self-attention
Self-attention
Self-attention
$\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\mathcal{N}}\}$
$\{\boldsymbol{y}_1, \dots, \boldsymbol{y}_{\mathcal{N}}\}$
$n_l \times$
Multilayer Perceptron
Layer Norm
Multi-Head Attention Mechanism
Layer Norm
Transformer Layer
$\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\mathcal{N}}\}$

Figure 3.12: **Left panel** Graphical representation of the Multi-Head Attention Mechanism. The different heads process the input vectors in parallel. The intermediate results from all attention heads are concatenated and then linearly projected to mix the information across heads. **Right panel** Graphical representation of a Transformer Layer. The input vectors, are first processed by a Multi-Head Attention Mechanism, which mixes the vectors, followed by a two-layers Feed-Forward neural network, which is used to introduce a non-linearity. Skip connections and Layer Normalization are also employed to stabilize the training and improve the convergence.

attention vector to mix the representations generated by the different heads. The final result is the set of attention vectors $\{\boldsymbol{A}_1, \dots, \boldsymbol{A}_{\mathcal{N}}\}$, which are the output of the Multi-Head Attention Mechanism (see left panel of Fig. 3.12 for a graphical representation).

3.4.2 Structure of a Transformer Layer

The set of output vectors generated by the attention mechanism is a linear combination of the input vectors [see Eq. (3.18)]. Despite using non-linear functions to compute the attention weights, for a fixed set of inputs, the resulting output vectors are restricted to the subspace spanned by the input vectors. This constraint can reduce the expressive power of the attention layer. To overcome this limitation and enhance the representational power of the Transformer layer, a non-linear transformation is applied to the output vectors of the attention mechanism. This is typically done through a *multilayer perceptron* (MLP), which processes each output vector identically and independently. In practice, this MLP is implemented as a two-layer fully-connected network with input and output dimensions both equal to d, and a hidden layer that typically has $2d$ neurons with ReLU activation functions. Moreover, to improve training efficiency, *residual connections* [85, 121] are introduced within the Transformer architecture. These skip connections are employed to bypass the Multi-Head mechanism and the MLP. Crucially, the fact that each component of the Transformer layer preserves the dimensionality of the vectors is essential for the effective use of residual connections. In addition *pre-layer normalization* [84, 121] is applied before both the attention mechanism and the MLP, to further enhance the optimization process (see the right panel Fig. 3.12 for a schematic representation of a Transformer layer).

In summary, a single Transformer layer consists of two main components. The first is the attention mechanism, which combines features from different input vectors, producing a new set of vectors where each one contains information from all the input vectors. The second component transforms non linearly the features within each vector independently. Thus, a single Transformer layer can be viewed as a non-linear function that takes an input set of vectors, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\mathcal{N}}\}$, and produces a transformed set of vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\mathcal{N}}\}$, of the same dimensionality, as output. This property allows to stack multiple layers on top of each other to construct deep networks capable of learning highly complicated representations. It is important to stress that while the architecture of each layer is identical, every layer has its own set of learnable parameters.

3.4.3 Vision Transformer

Inspired by successes of Transformers in natural-language processing, very small modifications led to the so-called Vision Transformer (ViT) [122], which has been applied to image classification tasks, achieving competitive results with respect to state-of-art deep CNNs, while being much more efficient than them.

In a Transformer architecture, the input consists of a set of vectors. When dealing with images, the key aspect is to decide how to convert the input image into a set of vectors. A straightforward approach would be to treat each pixel as a vector and apply a linear projection. However, the computational cost of the Transformer scales quadratically with the number of input vectors. Given that images contain millions of pixels, this scaling renders this approach impractical for real-world applications. The most common strategy introduced by Dosovitskiy et al. [122] consists in splitting the image into a set of patches, each of the same size. Specifically, the image is divided into non-overlapping patches of size $b \times b$, which are then *flattened* into one-dimensional vectors. Each vector, of dimension b^2, is subsequently embedded into a d-dimensional space via a linear projection, defining the input set of vectors for the Transformer.

In the following sections we propose an adaptation of the ViT architecture for the parametrization of a variational wave function for quantum spin models [31].

3.4.4 Transformer Variational Wave Functions

From the numerical perspective, density-matrix renormalization group (DMRG) [5] or its modern variations based upon tensor networks *Ansätze* [58] represent one of the few approaches that can accurately assess the ground-state properties of frustrated systems in one dimension, as the J_1-J_2 Heisenberg model of Eq. (3.3). In fact, the main limitation to the use of quantum Monte Carlo techniques [64] relies on the unknown sign structure of the ground-state wave function, which prevents one to perform unbiased projection techniques. From the NQS perspective, RBM states are able to reach an excellent accuracy; however, they suffer from poor scaling behavior, due to their *fully-connected* structure in which a single hidden layer is connected to all physical degrees of freedom [38] (see Sec. 3.3.3). This fact limits the applicability their relatively small clusters.

In order to overcome these problems, we propose a simplified version of the standard ViT architecture. The main advantage of this *Ansatz* lies in the possibility to mix both local and global structures, thus limiting the number of variational parameters and simplifying the learning process. We emphasize that a complex-valued parametrization is adopted without an *a priori* encoding of the sign structure (i.e., no information about the exact signs). In the following sections, we show that the ViT wave function can reach very high accurate results compared to DMRG calculations, even on large clusters, with less then one thousand parameters and few computational resources compared to other neural-network wave functions. Most importantly, the ViT accuracy can be systematically improved by changing the hyper-parameters of the architecture.

Our goal is to use the Transformer to parameterize the many-body wave function, in order to map spin configurations of the Hilbert space $\sigma = (\sigma_1^z, \dots, \sigma_N^z)$, with $\sigma_R^z = 2S_R^z = \pm 1$, to complex numbers $\Psi_\theta(\sigma) \in \mathbb{C}$. We take inspiration from the ViT [122] introduced for computer vision tasks, where the images are split into patches and these are taken as the input sequence to a Transformer (see Sec. 3.4.3). In the same way, starting

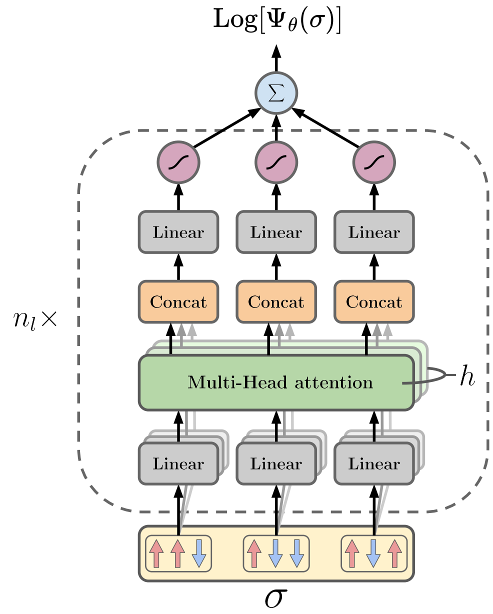

Figure 3.13: Scheme of the ViT wave function. The input spin configuration σ is split into patches of size b (which define a set of $\mathcal{N}$ vectors of dimension b). Each of them is linearly projected h times with different linear projections to produce $\mathcal{N}$ vectors of dimensionsionaly d/h. Then the attention function is applied in parallel and the h different r dimensional output vectors $\boldsymbol{A}_i^\mu$ are obtained. Then, they are concatenated to a d dimensional vector $\mathrm{Concat}(\boldsymbol{A}_i^1, \dots, \boldsymbol{A}_i^h)$ and, after another linear projection, the non-linear function logcosh$(\cdot)$ is applied. This architecture can be replicated and stacked n_l times. The last layer simply sums all the outputs and returns the logarithm of the ViT wave function.

from a spin configuration $\sigma = (\sigma_1^z, \dots, \sigma_N^z)$, we split it into $\mathcal{N}$ patches of b elements: $(\sigma^z_{(i-1)b+1}, \dots, \sigma^z_{(i-1)b+b})$, for $i = 1, \dots, \mathcal{N}$ (the total number of sites N must be a multiple of b, as a results $\mathcal{N} = N/b$ for the one-dimensional case). The sequence of these patches define the input sequence of vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_\mathcal{N}\}$, which are used to compute the attention vectors[8]. Then, a simplification of the original attention mechanism is considered, taking the attention weights only depending on positions i and j, but not on the actual values of the spins in these patches, thus leading to:

$$\boldsymbol{A}_i^\mu = \sum_{j=1}^{\mathcal{N}} \alpha_{ij}^\mu \mathrm{V}^\mu \boldsymbol{x}_j, \tag{3.20}$$

where $\mu = 1, \dots, h$, V^μ is a $d/h \times b$ matrix, with d the *embedding dimension* that must be

[8] We remark that in this simplified version of the ViT architecture we do *not* embed the sequence of input vectors in a d-dimensional space, but the vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_\mathcal{N}\}$ are defined directly through the spin configuration σ.

a multiple of the number of heads h and α^{μ} a $\mathcal{N} \times \mathcal{N}$ matrix. Crucially, in order to study frustrated quantum spin models with a non-positive ground state (in the computational basis), we choose all the parameters of the architecture to be *complex numbers*.

The input-independent attention mechanism in Eq. (3.20) is dictated by the fact that the attention weights should mainly depend on the relative positions among groups of spins and not on the actual values of the spins in the patches. This is expected to be true when the patches are far apart and is extended for generic positions i and j. The standard *dot-product self-attention* [see Eq. (3.18)] was originally developed for natural language processing tasks. As such, it can be modified to adapt to other types of problems. For instance, the *positional* or *factored* attention mechanism in Eq. (3.20) has been shown to outperform the standard one in tasks such as protein sequence analysis [123] and approximating conditional probabilities in generalized Potts models [124], and it also achieves competitive performance in image recognition tasks [125]. Additionally, since the attention weights α^{μ}_{ij} are complex-valued, it is not straightforward to define a suitable generalization of the softmax function employed in the standard attention mechanism [see Eq. (3.15) and Eq. (3.17)]. For a detailed and systematic comparison of different attention mechanisms in approximating the ground state of quantum spin models refer to Sec. 4.2.3.1.

Finally, after the concatenation of the heads, a further linear projection is taken, before the non linearity, here chosen as logcosh$(\cdot)$. This block can be repeated n_l times before applying the output layer in which all the values are summed to obtain the logarithm of the ViT wave function $\mathrm{Log}[\Psi_\theta(\sigma)]$ (see Fig. 3.13). Furthermore, a translationally-invariant wave function with $k = 0$ can be easily defined by considering the following two steps. First, we adapt the *relative positional encoding* [118] to periodic systems, taking $\alpha^{\mu}_{i,j} = \alpha^{\mu}_{i-j}$; as a result, the number of variational parameters for computing the attention vectors[see Eq. (3.20)] is reduced from $O(N^2)$ to $O(N)$. This procedure induces translational invariance between patches. To include also the one within patches, we perform the linear combination (see Appendix B):

$$\tilde{\Psi}_\theta(\sigma) = \sum_{r=0}^{b-1} \Psi_\theta(\mathrm{T}_r\sigma) \ , \tag{3.21}$$

where T_r is the translation operator. We emphasize that this approach requires a small summation (of b terms), which does not grow with the system size N.

The optimization process of all the complex parameters is performed by using stan-

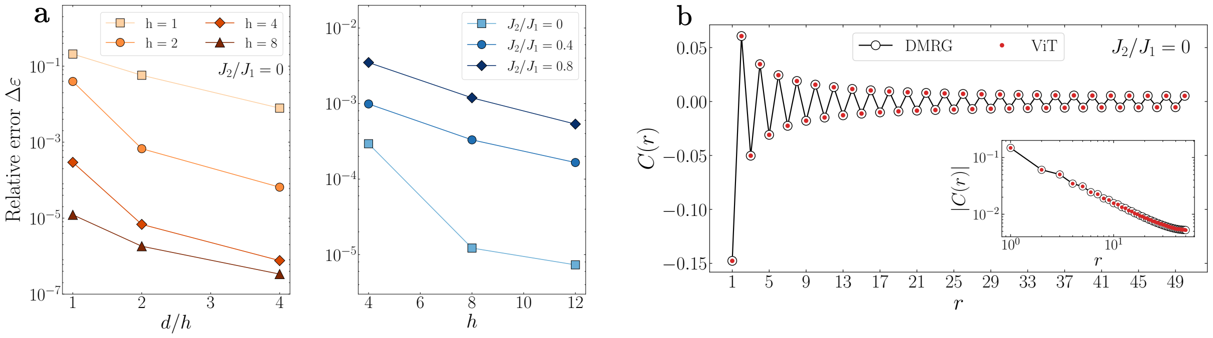

Figure 3.14: **Panel a:** Relative error $\Delta\varepsilon = |(E_{\text{ViT}} - E_{\text{DMRG}})/E_{\text{DMRG}}|$ of the ViT wave function by varying the hyper-parameters of the architecture for a cluster with $N = 100$ sites. Left panel: $\Delta\varepsilon$ as a function of d/h, with a fixed number of heads h, for the unfrustrated case. Right panel: $\Delta\varepsilon$ as a function of the number of heads h, with $d/h = 1$, for different values of frustration ratio. The reference energies are computed by DMRG [126] with a bond dimension up to $\chi = 600$ obtaining $E/J_1 = -0.4432295$ for $J_2/J_1 = 0$, $E/J_1 = -0.3803882$ for $J_2/J_1 = 0.4$, and $E/J_1 = -0.4216664$ for $J_2/J_1 = 0.8$. **Panel b:** The isotropic spin-spin correlations in real space $C(r)$ as computed by the ViT wave function (full dots) for the unfrustrated Heisenberg model ($J_2/J_1 = 0$) on a cluster with $N = 100$ sites. The DMRG results are also shown for comparison (empty circles). Inset: Log-log plot of the same correlation function.

dard variational Monte Carlo techniques (see Chapter 1), in particular the Stochastic Reconfiguration method [64, 69] (see Chapter 2). In the following, we mainly take $n_l = 1$, which represents the simplest possible adaptation of the Transformer architecture; indeed, even within this drastic assumption, we obtain excellent results in both gapless and gapped phases. At the end, we show the effect of a deeper network with $n_l > 1$. All the simulations are performed by fixing the patch size $b = 4$.

3.4.5 Numerical Results

3.4.5.1 Accuracy for the ground state energy

We start by discussing how the accuracy of the ViT wave function with one layer can be systematically improved by varying its two hyper-parameters, i.e., the number of heads h and the ratio d/h. We consider a cluster with $N = 100$ sites and three different values of the frustration ratio: $J_2/J_1 = 0$ (unfrustrated, gapless), $J_2/J_1 = 0.4$ (weakly-frustrated, gapped), and $J_2/J_1 = 0.8$ (strongly-frustrated, gapped); the reference energy is computed by using the standard DMRG approach (imposing periodic-boundary conditions on the Hamiltonian [126]). In the panel (a) of Fig. 3.14, we show the accuracy of the ground-state energy for the unfrustrated case as a function of d/h fixing the number of heads h, and for the three values of J_2/J_1 when increasing the number of heads h, at fixed ratio d/h. Even though there is a general difficulty in reconstructing the exact sign structure in highly-frustrated regimes [38, 127–130], we obtain an excellent approximation of the correct energy for all the values of J_2/J_1 that have been considered, e.g., an accuracy $\Delta\varepsilon \lesssim 0.1\%$ for $J_2/J_1 = 0.8$ and $\Delta\varepsilon \approx 0.01\%$ for $J_2/J_1 = 0.4$.

3.4.5.2 Correlation functions

Let us now move to the analysis of the correlation functions. From the previous results, we choose $h = 8$ and $d/h = 1$ as a good compromise between accuracy and complexity, for which the network can be trained on $N = 100$ sites in a few hours on ten CPUs or in a few minutes on a GPU. The spin-spin correlations are defined as

$$C^{\nu\nu}(r) = \frac{1}{L}\sum_{R=0}^{L-1}\langle \hat{S}^{\nu}_{R}\hat{S}^{\nu}_{R+r}\rangle, \tag{3.22}$$

where $\nu = x$, y, or z and $\langle \dots \rangle$ represents the expectation value over the variational quantum state.

In particular, we focus on isotropic spin-spin correlations $C(r) = [C^{zz}(r) + C^{xx}(r) + C^{yy}(r)]/3$ and the corresponding structure factor in Fourier space $S(k) = \frac{1}{N}\sum_{r=0}^{N-1} \mathrm{e}^{ikr} C(r)$. In panel (b) of Fig. 3.14, we show the results of the real-space correlations $C(r)$ for the unfrustrated Heisenberg model ($J_2/J_1 = 0$) on a cluster with $N = 100$ sites, comparing them to the DMRG outcomes (with periodic-boundary conditions). Remarkably, the ViT *Ansatz* is able to match the DMRG calculations at all distances, demonstrating that the global structure of the multi-head attention layer is able to build the algebraic long-range tail.

The high flexibility of the ViT state is also demonstrated by considering the three different regimes, with commensurate (i.e., $S(k)$ peaked at $k = \pi$) or incommensurate (i.e., $S(k)$ peaked at $k \neq \pi$) correlations, see panel (a) of Fig. 3.14.

The gapped phase is characterized by a finite dimer order (implied by the two-fold degeneracy of the ground state, in the thermodynamic limit). On any finite system, there is an exponentially small gap between the two states, with $k = 0$ and $k = \pi$, and the insurgence can be detected from the connected dimer-dimer correlations:

$$D(r) = \frac{1}{L}\sum_{R=0}^{L-1}\langle \hat{S}^{z}_{R}\hat{S}^{z}_{R+1}\hat{S}^{z}_{R+r}\hat{S}^{z}_{R+r+1}\rangle - [C^{zz}(r = 1)]^2, \tag{3.23}$$

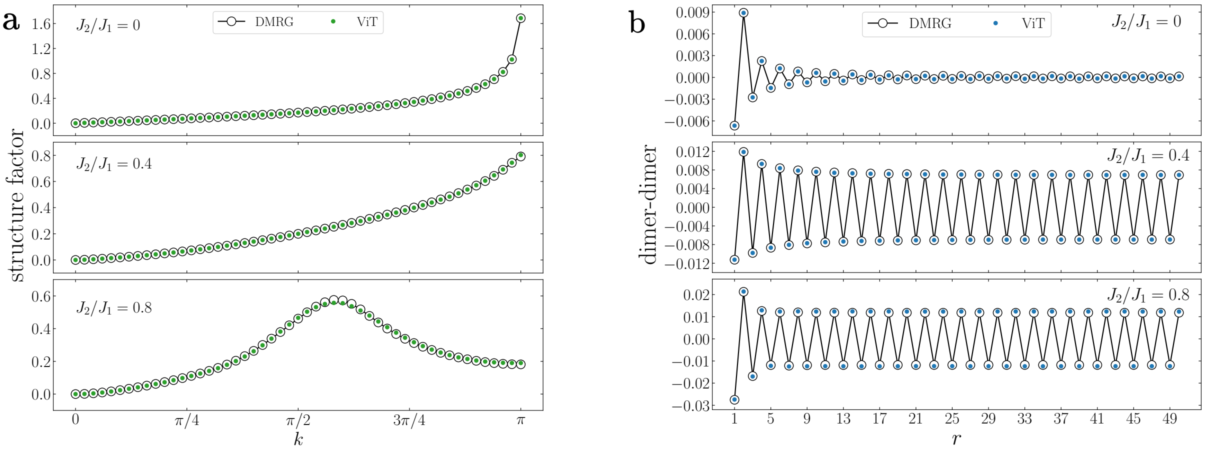

Figure 3.15: **Panel a:** The spin-spin structure factor $S(k)$ as computed by the ViT wave function (full dots) for $J_2/J_1 = 0$ (upper panel), $J_2/J_1 = 0.4$ (middle panel) and $J_2/J_1 = 0.8$ (lower panel) on a cluster with $N = 100$ sites. The DMRG results are also shown for comparison (empty circles). **Panel b:** Dimer-dimer correlations as computed by the ViT wave function (full circles) for $J_2/J_1 = 0$ (upper panel), $J_2/J_1 = 0.4$ (middle panel) and $J_2/J_1 = 0.8$ (lower panel) on a cluster with $N = 100$ sites. The DMRG results are also shown for comparison (empty circles).

where $C^{zz}(r = 1)$ is the z component of the spin-spin correlation function at distance $r = 1$ defined in Eq. (3.22). Notice that this definition considers only the z component of the spin operators [131]. In panel (b) of Fig. 3.15, we show the results for the three regimes which characterize the model. Again, the agreement with DMRG calculations is excellent in all cases, and the ViT state is able to perfectly reproduce the presence of dimer order.

3.4.5.3 Limitations

The ViT wave function can be systematically improved not only by increasing the embedding dimension d or the number of heads h (as shown in panel (a) of Fig. 3.14) but also by making deep the architecture by the stacking of multiple Transformer layers. However, optimizing deep complex-valued networks poses significant challenges [10], as standard techniques used to facilitate the training of deep architectures, such as layer normalization and appropriate activation functions, do not easily generalize to complex-valued networks. For this reason we develop a procedure based on the physical interpretation of the attention weights. We start by setting for each head and layer $\alpha_{i-j} = 0$ if $|i-j| > \text{cut}$, with cut $< N/b$, training only the remaining weights. Small cut values (e.g., cut $= 1$)

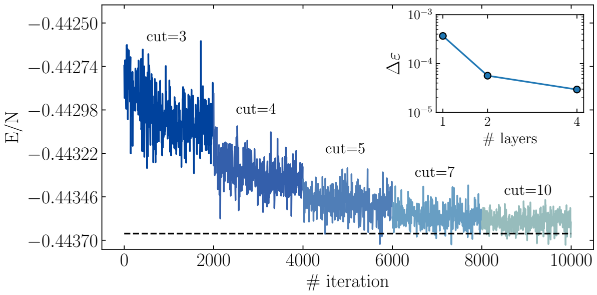

Figure 3.16: Optimization of the DeepViT with $n_l = 4$, where each layer has $h = 2$ and $d/h = 2$, for the Heisenberg model with $N = 40$ sites. Along the process, the cut in the attention is fixed and progressively increased from 1 to 10 (the first two values are not shown for better readability). At the end, once the cut has been completely relaxed, the full translational invariance is restored [see Eq. (3.21)] to compute the accuracy in the energy. Inset: Relative error $\Delta\varepsilon$ of the DeepViT wave function by varying the number of layers. The reference energy is computed by DMRG [126] with a bond dimension up to $\chi = 600$ obtaining $E/J_1 = -0.443663$.

are good starting points for stable optimizations. Then the cut is relaxed until reaching $\mathcal{N} = N/b$, where all-to-all connections among the inputs of each layer are restored. As an example, the results for the Heisenberg model with $N = 40$ are shown in Fig. 3.16. Here, we take $n_l = 4$ (each layer has $h = 2$ and $d/h = 2$) and perform the optimization stages with cut $= 1, \dots, 10$. Every time, when the cut is relaxed, the accuracy of the energy improves.

While a single layer of a complex-valued Transformer suffices to achieve highly accurate results for one-dimensional models, we expect that deeper networks will be crucial for competitive performance in two-dimensional systems compared to state-of-the-art numerical methods (see Sec. 5.2.2). In such cases, the relax-cut procedure may not be the optimal solution. In the next Chapter, we develop a general framework that enables the use of standard deep architectures from machine learning community to parametrize variational quantum states.

Chapter 4

A Representation Learning perspective on Neural-Network Quantum States

In this Chapter, we propose a general framework for constructing highly accurate NQS by leveraging the principles of *Representation Learning* [28], a cornerstone of modern deep learning [132]. We apply this framework, in combination with the optimization method described in Chapter 2, to tackle the challenging J_1-J_2 Heisenberg model on a square lattice. Specifically, we demonstrate the effectiveness of this approach by achieving the *state-of-the-art* ground state energy on the 10×10 lattice, a well-established benchmark in the study of highly-frustrated magnetism for evaluating the accuracy of new variational wave functions [27]. Moreover, we show the flexibility of this framework by demonstrating that a NQS *pretrained* close to the phase transition point yields features that can be *fine-tuned* to accurately describe a wide region of the phase diagram [39].

4.1 What is Representation Learning?

The performance of an algorithm strongly depends on the *representation* of the data [132, 133]. For instance, searching for an element in a collection of data can be exponentially faster if the data are suitably structured. Similarly, simple arithmetic operations with Arabic numerals can become more complicated when using Roman numerals. In general, many problems can be easily solved by designing the right set of features and subsequently

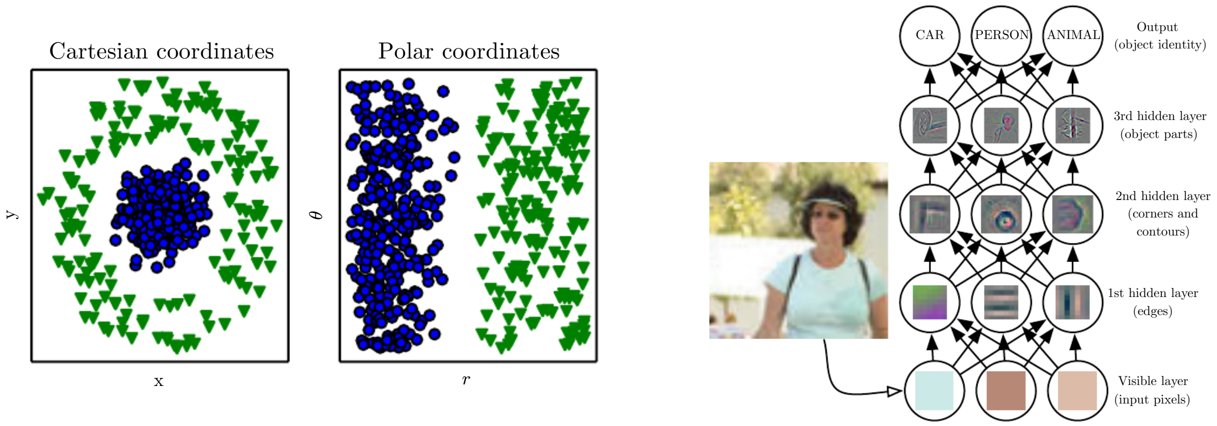

Figure 4.1: **Left panel** Example of two different representations of the same dataset. While the data are not linearly separable in Cartesian coordinates, they becomes trivially separable in polar coordinates. **Right panel:** Schematic representation of how a deep neural network hierarchically learns features from input data. It first extracts low-level features such as edges in the first hidden layer, followed by more complicated ones such corners and contours in subsequent layers, allowing the recognition of high-level objects (e.g., car, person, animal) in the output layer. Both images are adapted from the book *Deep Learning* by Goodfellow et al. [132].

providing them to a simple algorithm. Consider for example the task of separating two sets of data by drawing a line between them. In the left panel of Fig. 4.1, the data are represented in both Cartesian and polar coordinates. While the separation task cannot be solved in Cartesian coordinates, it becomes trivial when the data is transformed into polar coordinates.

For decades, during the *classical machine learning era*, researchers first designed a set of well-suited features to describe the inputs, requiring careful engineering and considerable domain expertise, and then trained a simple machine learning algorithm (e.g., linear regression) on top of those features to perform a given task. While handcrafting suitable features are feasible for simple problems (as in the one described above), the challenge arises in determining the optimal features for more complicated tasks. In the *deep learning era*, the paradigm shifted towards using machine learning to discover not only the mapping from representation to output, but also the representation itself [132, 133]. This approach, dubbed *Representation Learning*, outperforms hand-crafted representations in a variety of domains. Furthermore, methods based on neural networks that automatically discover the right set of features can rapidly adapt to new tasks with minimal modifications.

In particular, deep neural networks are able to extract high-level and abstract features from raw data by constructing hierarchical representations, where complicated features are expressed in terms of other, simpler features. In the right panel of Fig. 4.1 we show how an image of a person can be identified by combining simpler concepts such as corners and contours, which are themselves defined by even more basic concepts such as edges. Thus, the application of successive layers of a deep neural network can be seen as performing a sequence of transformation on the original data, thereby facilitating the resolution of the task.

The general structure of a modern deep neural networks [23, 96, 134] consists of two components:

1. The first component is a deep neural network which maps the original input data into a representation, dubbed *hidden representation*, which lives in a feature space.

2. The second is simple neural network (e.g., a linear classifier for image classification tasks), dubbed *output layer*, which acts on the hidden representation and generates the final output.

Interestingly, when the output layer is applied directly to the original data, the results are poor for complicated tasks. However, when it acts on the hidden representation, the results are significantly improved [134]. This demonstrates that while the problem may not be linearly separable in the original input space, the deep neural network transforms the data into a new space where linear classifier, such as those commonly used in output layers, can effectively solve the problem. The situation is analogous to the toy example in the left panel of Fig. 4.1, once the appropriate set of coordinates (in the example the polar coordinates) is identified, then the problem can be solved with a simple linear classifier.

4.2 A new parametrization for Neural-Network Quantum States

In the previous Chapter, we introduced the concept of NQS and discussed two examples of such variational states applied to the one-dimensional J_1-J_2 Heisenberg model. Among these, the Transformer wave function emerged as a promising candidate, demonstrating high accuracy. However, it still presents limitations, particularly due to the use of

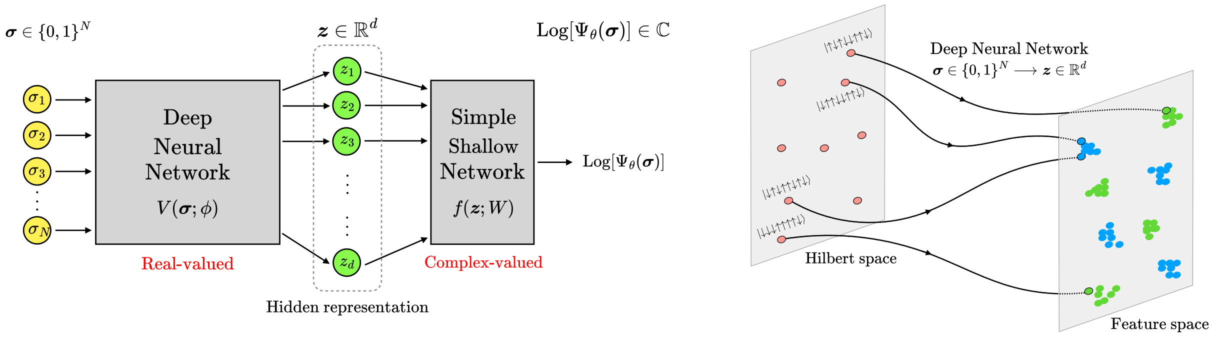

Figure 4.2: **Left panel:** The NQS is defined as the composition of two functions: first, a deep neural network $V(\sigma;\phi)$ (with real-valued parameters) maps the input configurations σ into hidden representations $\boldsymbol{z}$; then, a simple shallow network $f(\boldsymbol{z};W)$ (with complex-valued parameters) generates the logarithm of the amplitudes $\mathrm{Log}[\Psi_\theta(\sigma)]$ starting from hidden representations. **Right panel:** Pictorial illustration of the mapping process carried out by the deep neural network. During this process, the spin configurations of the Hilbert space σ are embedded into a feature space $\boldsymbol{z} \in \mathbb{R}^d$. The colours of the clusters in the feature space are related to the sign of the amplitudes $\mathrm{Log}[\Psi_\theta(\sigma)]$, corresponding to the physical configurations σ, as discussed in Section 4.3.1.

complex-valued parameters. In this Chapter, we overcome these limitations by adopting a new perspective on NQS, leveraging the principles of *Representation Learning* [133].

We start by reframing the NQS as feature extractors rather than just universal approximators of complicated functions. In this framework, the variational state is naturally perceived as the composite result of two distinct functions, each with a specific role:

$$
\begin{aligned}
\boldsymbol{z} &= V(\sigma;\phi)\ , \\
\mathrm{Log}[\Psi_\theta(\sigma)] &= f\left(\boldsymbol{z};W\right)\ ,
\end{aligned}
\tag{4.1}
$$

where the variational parameters are partitioned into two blocks $\theta = \{\phi, W\}$. The function $V(\cdot;\phi)$ is parameterized as a *deep* neural network, mapping physical configurations σ to the *hidden representations* vectors $\boldsymbol{z}$, which belong to a d-dimensional *feature space*. Conversely, $f(\cdot;W)$ is a *shallow* neural network used to generate a single scalar value $f(\boldsymbol{z};W)$ from the hidden representations $\boldsymbol{z}$. This final value is used to predict the amplitude corresponding to the input spin configuration. In order to predict both modulus and phase of the variational state (which is fundamental in cases where the exact sign is not known *a priori*), it is convenient to employ a complex-valued variational state. The structure of the *Ansatz* in Eq. (4.1) suggests the possibility of taking ϕ as real-valued parameters in the deep neural network $V(\cdot;\phi)$. Subsequently, only the parameters W of the shallow function $f(\cdot;W)$ can be taken complex-valued. We schematically represent these two steps in the left panel of Fig. 4.2; instead, a pictorial scheme of the mapping process from the physical space of the spin configurations to the feature space is depicted in the right panel of Fig. 4.2.

In the recent past, a few works showed that depth is crucial to achieve high accuracies on two-dimensional systems [36, 135–137]. However, training deep networks is a complicated task and leveraging standard building blocks simplifies the procedure. In particular, Layer Normalization [121], skip connections [96], and appropriate activation functions [138] have been developed for real-valued architectures. Unfortunately, these key elements do not have straightforward generalizations for complex-valued neural networks. For these reasons, the optimization of a deep Transformer architecture having complex-valued parameters necessitated the development of a heuristic procedure involving for example the introduction of a cut in the attention weights (see Sec. 3.4.5.3). Instead, within the current approach, such constraints are no longer required. The newly proposed *Ansatz* can be trained straightforwardly from scratch, without the need for additional restrictions and obtaining more accurate results. Moreover, working with real-valued parameters facilitates to gain physical insights into what the neural network is learning during the optimization by visualizing, for example, the hidden representations (see Section 4.3).

4.2.1 The choice of deep and shallow architectures

The composition in Eq. (4.1), inspired by Representation Learning, is general and, in principle, works for any choice of $V(\bullet;\phi)$ and $f(\bullet;W)$. However, in practice, the performance of this approach can be highly dependent on the specific properties of these functions, which must be well-suited to the specific problem. In Chapter 3, we discussed the adaptation of the ViT architecture (see Sec. 3.4.3) for parameterizing variational states to study one-dimensional frustrated systems [31], achieving results comparable with DMRG on large clusters. Given the excellent results of the ViT architecture, we aim to exploit its properties within this new framework. Specifically, we propose to parametrize the function $V(\bullet;\phi)$ with a Vision Transformer. Unlike the previous complex-valued parametrization, due to the structure in Eq. (4.1), all parameters can be taken to be real-valued. This allows us to employ the full ViT architecture (see right panel of Fig. 3.12) using the standard building blocks which allow the efficient optimization of deep architectures.

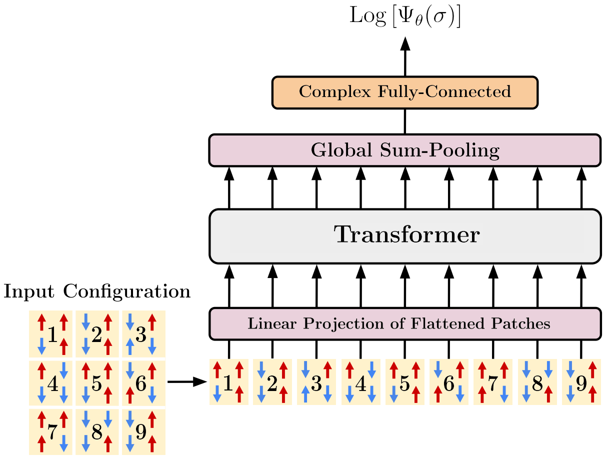

Figure 4.3: The input spin configuration σ is partitioned into patches, which are linearly projected in a d-dimensional embedding space and then processed by a Vision Transformer. The latter one builds new representations of the patches, which are then combined through summation and fed into a final single complex-valued fully-connected layer in order to obtain the logarithm of the (complex) wave function. Notice that this is a particular instantiation of the more general scheme proposed in the left panel of Fig. 4.2.

Instead, the function f which constitute the output layer is chosen to be:

$$f(\boldsymbol{z}; W) = \sum_{\alpha=1}^{K} g\left(b_\alpha + \boldsymbol{w}_\alpha \cdot \boldsymbol{z}\right) \ , \tag{4.2}$$

where the variational the parameters $W = \{b_\alpha, \boldsymbol{w}_\alpha\}_{\alpha=1}^{K}$ of the linear transformation in Eq. (4.2) are taken to be complex valued in order to describe non-positive ground states. The number of hidden neurons K is a hyperparameter of the network. Here, we set $g(\cdot) = \log\cosh(\cdot)$ and $K = d$, thus $f(\boldsymbol{z}, W)$ has the same functional form as the well-known Restricted-Boltzmann Machine introduced by Carleo and Troyer [7]. Crucially, in this case it is not applied to the physical configuration σ but instead to the hidden representation $\boldsymbol{z}$. This is the change of paradigm that we want to emphasize.

With these choices, the process of constructing the amplitude corresponding to a physical spin configuration σ involves the following steps (see Fig. 4.3):

1. The input spin configuration σ is initially divided into $\mathcal{N}$ patches.

2. The patches are linearly projected into a d-dimensional embedding space, resulting in a sequence of vectors $(\mathbf{x}_1, \cdots, \mathbf{x}_{\mathcal{N}})$, where $\mathbf{x}_i \in \mathbb{R}^d$.

3. A ViT processes these embedded patches, producing another sequence of vectors $(\mathbf{y}_1, \cdots, \mathbf{y}_{\mathcal{N}})$, where $\mathbf{y}_i \in \mathbb{R}^d$.

4. The hidden representation $\boldsymbol{z}$ of the configuration σ is defined by summing all these output vectors: $\boldsymbol{z} = \sum_{i=1}^{\mathcal{N}} \mathbf{y}_i \in \mathbb{R}^d$.

5. A fully-connected layer with complex-valued parameters, defined in Eq. (4.2), produces the amplitude $\mathrm{Log}[\Psi_\theta(\sigma)]$ corresponding to the input configuration σ.

Moreover, we want to emphasize that while the vector $\mathbf{x}_i$ depends solely on the spins contained in the i-th patch, the resulting vector $\mathbf{y}_i$, due to the attention mechanism, is a function of all the spins in the configuration. The ViT architecture is constructed as a sequence of n_l Transformer layers, in each of them, the Multi-Head Attention Mechanism (with h heads) is followed by a two layers fully-connected network (for a detailed description of the Transformer Layer refer to Section 3.4.2).

The only custom modification we introduce to the standard Vision Transformer architecture [139] is the use of the *Factored/Positional* attention mechanism [31, 123–125, 140, 141], in which the attention weights are input-independent variational parameters (see Section 4.2.3.1 for a detailed comparison of different attention mechanisms) [141]. The Multi-Head Factored Attention (MHFA) mechanism can be formally implemented as follows:

$$A_{i,p} = \sum_{q=1}^{d} W_{p,q} \sum_{j=1}^{\mathcal{N}} \alpha_{i,j}^{\mu(q)} \sum_{r=1}^{d} V_{q,r} x_{j,r} \ , \tag{4.3}$$

where $\mu(q) = \lceil q\ h/d \rceil$ select the correct attention weights of the corresponding head, being h the total number of heads. The matrix $V \in \mathbb{R}^{d\times d}$ linearly transforms each input vector identically and independently. Instead, the attention matrices $\alpha^\mu \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ combine the different input vectors and the linear transformation $W \in \mathbb{R}^{d\times d}$ mixes the representations of the different heads. To enforce translational symmetry, we define the attention weights as $\alpha_{i,j}^\mu = \alpha_{i-j}^\mu$, thereby ensuring translational symmetry among patches. This choice reduces the computational cost during the restoration of the full translational symmetry through quantum number projection [142, 143]. Under the specific assumption of translationally invariant attention weights, the Factored attention mechanism can be technically implemented as a convolutional layer with d input channels, d output channels and a very specific choice of the convolutional kernel: $K_{p,r,k} = \sum_{q=1}^{d} W_{p,q} \alpha_k^{\mu(q)} \sum_{r=1}^{d} V_{q,r} \in \mathbb{R}^{d\times d\times\mathcal{N}}$. However, it is well-established that weight sharing and low-rank factorizations in learnable tensors within neural networks can lead to significantly different learning dynamics and, consequently, different final solutions [144–146].

At the end, we provide a pseudocode (see Algorithm 2) describing the steps for the implementation of the Vision Transformer architecture described above. In particular we emphasize that skip connections and Layer Normalization are implemented as described in Ref. [121].

Algorithm 2 Vision Transformer Wave Function

```
Input configuration $\sigma \in \{-1, 1\}^N$
Patch and Embed: $\mathcal{X} \leftarrow (\boldsymbol{x}_1, \dots \boldsymbol{x}_{\mathcal{N}}) \in \mathbb{R}^{\mathcal{N} \times d}$
for $i = 1, n_l$ do
    $\mathcal{X} \leftarrow \mathcal{X} + \mathrm{MHFA}(\mathrm{LayerNorm}(\mathcal{X}))$
    $\mathcal{X} \leftarrow \mathcal{X} + \mathrm{MLP}(\mathrm{LayerNorm}(\mathcal{X}))$
end for
$(\boldsymbol{y}_1, \dots \boldsymbol{y}_{\mathcal{N}}) \leftarrow \mathrm{LayerNorm}(\mathcal{X})$
$\boldsymbol{z} \leftarrow \sum_{i=1}^{d} \boldsymbol{y}_i$
$\mathrm{Log}[\Psi_\theta(\sigma)] \leftarrow \sum_{\alpha=1}^{d} g(b_\alpha + \boldsymbol{w}_\alpha \cdot \boldsymbol{z})$
```

Notice that the structure of this variational *Ansatz* requires a large number of parameters. In order to optimize them, modern formulations of the Stochastic Reconfiguration technique [69], able to deal with a large number of variational parameters [27, 36], are used (see Sec. 2.3).

4.2.2 Benchmark model : J_1-J_2 Heisenberg on the square lattice

One of the most paradigmatic example of quantum many-body spin model is the J_1-J_2 Heisenberg model on the square lattice:

$$\hat{H} = J_1 \sum_{\langle i,j \rangle} \hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_j \tag{4.4}$$

where $\hat{\boldsymbol{S}}_i = (S_i^x, S_i^y, S_i^z)$ is the $S = 1/2$ spin operator at site i and J_1 and J_2 are nearest- and next-nearest-neighbour antiferromagnetic couplings, respectively.

The ground state of this model features magnetic order in the two limits $J_1 \ll J_2$ and $J_1 \gg J_2$. Specifically, when $J_2 = 0$ the model reduces to the unfrustrated Heisenberg model where long-range Néel order is present [147, 148]. In the opposite regime

$J_2/J_1 \to \infty$, the system exhibits instead columnar magnetic order. The presence of magnetic order can be characterized with the spin structure factor

$$S(\boldsymbol{k}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} \langle \hat{\boldsymbol{S}}_{\boldsymbol{0}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R}} \rangle \ , \tag{4.5}$$

where $\boldsymbol{R}$ runs over all the lattice sites of the square lattice. Specifically, the long-range Néel order can be detected by measuring the square magnetization as

$$m^2_{\text{Néel}} = \frac{S(\pi,\pi)}{L^2} \ , \tag{4.6}$$

instead, the columnar magnetic order is identified by the following order parameter

$$m^2_{\text{stripe}} = \frac{S(0,\pi)+S(\pi,0)}{2L^2} \ . \tag{4.7}$$

In the intermediate region, around $J_2/J_1 \approx 0.5$ the system is highly frustrated and the ground-state properties have been the subject of many studies over the years, often with conflicting results [37, 98, 149]. In particular, several works focused on the highly-frustrated regime, which turns out to be challenging for numerical methods [10, 11, 17, 36, 37, 98, 128, 143, 149–156].

4.2.3 Numerical Results

Our objective is to approximate the ground state of the J_1-J_2 Heisenberg model on the square lattice [see Eq. (4.4)]. Specifically, we use the parametrization outlined in Sec. 4.2.1, where a ViT architecture is employed for the deep neural network, and a fully-connected network for the output layer. The J_1-J_2 Heisenberg model is a translationally invariant Hamiltonian. However, as discussed in Sec. 3.4.3, a key element of the Vision Transformer is the division of the input into non-overlapping patches of shape $b \times b$ (for two-dimensional input). In general, the clustering b is an hyperparameter of the architecture and an optimal selection involves a trade-off. From one side, large patches enhance computational complexity, due to the quadratic scaling of attention mechanism with the input sequence lenght. On the other side, augmenting patch size implies an additional cost to restore translational symmetry within the system. To reduce the cost of the *a posteriori* symmetrization, we employ translational invariant attention weights (as detailed in Sec. 4.2.1), requiring only a summation involving b^2 terms for the restoration of the translational symmetry. Other symmetries of the Hamiltonian in Eq. (4.4) [rotations, reflections (C_{4v}

point group) and spin parity] can also be restored within quantum number projection (see Appendix B). As a result, the symmetrized wave function reads:

$$\tilde{\Psi}_\theta(\sigma) = \sum_{i=0}^{b^2-1} \sum_{j=0}^{7} \left[\Psi_\theta(T_i R_j \sigma) + \Psi_\theta(-T_i R_j \sigma)\right] \ . \tag{4.8}$$

In the last equation, T_i and R_j are the translation and the rotation/reflection operators, respectively. Furthermore, due to the $SU(2)$ spin symmetry of the J_1-J_2 Heisenberg model, the total magnetization is also conserved and the Monte Carlo sampling can be limited in the $S^z = 0$ sector for the ground state search.

In the following Sections, we first conduct a systematic comparison of different attention mechanisms on a small lattice to identify the most suitable one for parametrizing the Transformer wave function. Then, we use the selected architecture to determine the ground state energy on larger lattices.

4.2.3.1 Are queries and keys always relevant in the attention mechanism?

While elements like the MLP, Layer Normalization, and skip connections are task-agnostic and offer broad applicability, the functional form of the dot product attention mechanism was originally tailored for natural language processing tasks. One wonders if the dot product attention mechanism (see Sec. 3.4.1) provides an inductive bias which is the most appropriate in *any* data domain [141]. In this Section, we tackle this question by systematically investigating the performance of different attention mechanism within Transformer wave functions.

During the years, several works proposed different parametrizations of the attention weights [118, 158, 159]. Here, we consider three different mechanisms, all based on relative positional encoding [118], as appropriate for our purpose.

1. *T5 attention*, introduced in Ref. [119], is one of the most popular attention mechanisms:

$$\alpha_{ij}^{T5}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\exp\left(\frac{\boldsymbol{x}_i^T Q^T K \boldsymbol{x}_j}{\sqrt{d}} + p_{i-j}\right)}{\sum_{k=1}^n \exp\left(\frac{\boldsymbol{x}_i^T Q^T K \boldsymbol{x}_k}{\sqrt{d}} + p_{i-k}\right)} \ . \tag{4.9}$$

2. *Decoupled attention*, introduced in Ref. [157]:

$$\alpha_{ij}^{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\exp\left(\frac{\boldsymbol{x}_i^T Q^T K \boldsymbol{x}_j}{\sqrt{d}}\right)}{\sum_{k=1}^n \exp\left(\frac{\boldsymbol{x}_i^T Q^T K \boldsymbol{x}_k}{\sqrt{d}}\right)} + p_{i-j} \ . \tag{4.10}$$

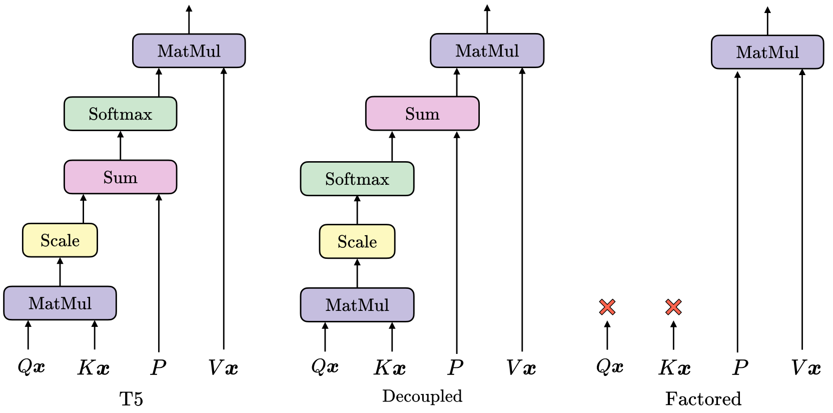

Figure 4.4: Schematic representation of the attention mechanisms employed in this work: T5 [119] (left panel), Decoupled [157] (central panel) and Factored/Positional [123–125] (right panel) attention. In each of them, relative positional encoding is used. The matrices Q, K, V and P are referred to queries, keys, values and positional encoding matrix, respectively. Refer to Eqs. (4.9),(4.10) and (4.11) in the main text for the analytical expressions.

3. *Factored/Positional attention*, introduced in Refs. [123–125]:

$$\alpha^F_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = p_{i-j} \ . \tag{4.11}$$

In Fig. 4.4, we show a schematic representation of these three different attention mechanisms. The Factored version has a reduced number of parameters, being the attention weights input independent. Regarding the computational cost for the calculation of each attention weight, we have $O(1)$ complexity in the Factored case and $O(\mathcal{N}d^2) + O(\mathcal{N}^2 d)$ in the other two cases, where $\mathcal{N}$ is the number of input vectors. Decoupled attention, as represented by Eq. (4.10), is the simplest extension of the Factored version in Eq. (4.11), where the attention weights now factor in the input dependence: setting $Q = K = 0$ allows to recover the Factored attention, albeit with a constant shift. Instead, in T5 attention [see Eq. (4.9)] all the attention weights are constrained to be positive due to the global softmax activation. In order to assess the efficacy of the three distinct attention mechanisms introduced above, we perform simulations utilizing ViT architectures to approximate the ground state of the J_1-J_2 Heisenberg model [see Eq. (4.4)] on a 6×6 lattice with periodic boundary conditions. We employ identical hyperparameters (embedding dimension d, number of heads h, and number of layers n_l), modifying only the attention mechanism, namely T5 [see Eq. (4.9)], Decoupled [see Eq. (4.10)], and Factored [see Eq. (4.11)]. In Fig. 4.5, we report the optimization curves of the relative error of the

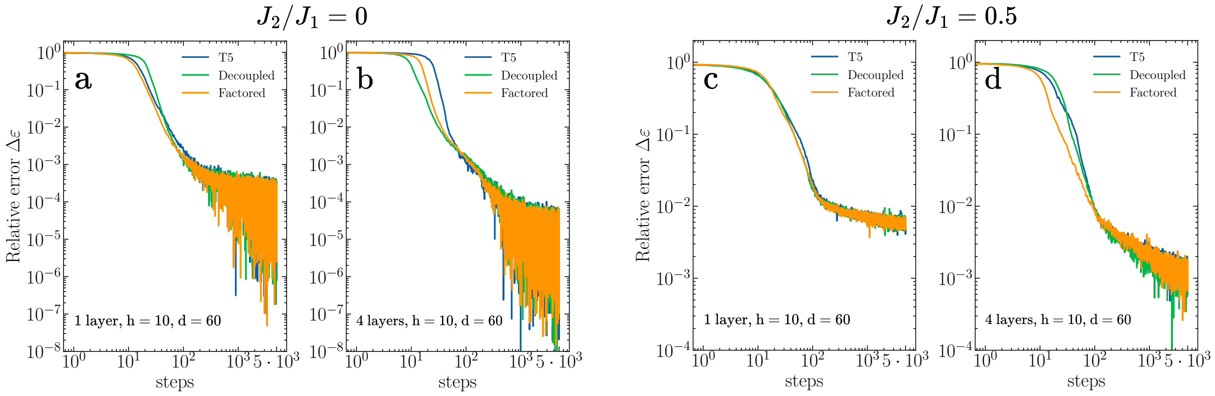

Figure 4.5: Relative error $\Delta\varepsilon = |(E_0 - E_{\text{ViT}})/E_0|$ during the optimization of the ViT wave function on the J_1-J_2 Heisenberg model at $J_2/J_1 = 0$ (panels **a** and **b**) and at $J_2/J_1 = 0.5$ (panels **c** and **d**) on a 6×6 lattice with periodic boundary conditions. The exact energies E_0 are computed with exact-diagonalization approaches. All architectures are trained with the same optimization protocol, using SR with cosine decay scheduler for the learning rate with an initial value of $\tau = 0.03$. The optimization curves are consistent across multiple runs with different random initialization of the parameters.

variational energy with respect to the exact ground-state energy as a function of the optimization steps. On the left, we present the results for the unfrustrated case ($J_2/J_1 = 0$) using ViT architectures with one [panel (a)] and four [panel (b)] layers. Instead, on the right we report the results in the frustrated regime ($J_2/J_1 = 0.5$), again using one [panel (c)] and four [panel (d)] layers architectures. We highlight that, while it is feasible to enhance the performance of the variational state by employing larger architectures (such as increasing the number of layers), utilizing T5 or Decoupled attention mechanisms with input-dependent attention weights, and subsequently increasing the computational complexity and parameter count via the matrices Q and K, does not yield improved results compared to the Factored attention with input-independent attention weights. Notably, not only the final accuracies are practically identical, but also the learning dynamics exhibit similar behavior. In Table 4.1 we report the results of the J_1-J_2 Heisenberg model on a 6×6 lattice at $J_2/J_1 = 0.5$, obtained using a four-layer architecture (as in panel (d) of Fig. 4.7). The first column shows the final mean energy achieved by the different attention mechanisms, the second column indicates the number of parameters employed in the architectures, and the last column presents the computational time measured on a single GPU A100. The exact energy for this model can be computed using exact-diagonalization techniques, yielding $E_0/J_1 = -0.503810$. Although the accuracy of the results can be further enhanced by restoring the physical symmetries of the model through quantum number projection approaches [106, 160] (see Sec. 4.2.3.2), this goes beyond the scope this Section.

The main result of the numerical simulations reported in Fig. 4.7 is that, using a ViT employing T5, Decoupled and Factored attention, the final accuracy is practically the same (see Table 4.1). This suggests that, in the case of T5 and Decoupled attention, queries and keys are effectively not used in the optimized solution. To validate this statement we study the *attention weights* in the different solutions, here dubbed *attention maps*. For the analysis, we use a single-layer architecture, where the interpretation of the results is simplified since the patches are only mixed within the attention mechanism, and the subsequent MLP cannot modify the relative weights among the various attention vectors. In panel (a) of Fig. 4.6, we consider the case of T5 attention, plotting the attention weights defined in Eq. (4.9) for three different input spin configurations. We first check that at the beginning, with random parameters, the attention maps depend on the inputs (top row), ensuring that we have an unbiased initialization. In the bottom row, we show that the architecture after optimization produces input-independent attention maps, thus automatically recovering a positional only solution. In panel (b) of Fig. 4.6, we consider the case of Decoupled attention, plotting separately the input dependent and the positional contributions of the attention weights [see Eq. (4.10)]. Again, after optimization the network swaps from an unbiased solution (top row) to a positional only solution (bottom row), where the input-dependent term converges approximately to the identity matrix shifted element-wise by a constant. In other words, Factored attention is spontaneously recovered from the Decoupled version. Additionally, in Appendix C, we provide also analytic calculations about the efficacy of input-independent attention mechanisms for approximating quantum states.

Table 4.1: Simulations on the 6×6 lattice at $J_2/J_1 = 0.5$ with four layers ViT. The exact ground state energy of the model is $E_0/J_1 = -0.503810$.

	Energy	Parameters	Time
T5	-0.50318(1)	184,260	10h
Decoupled	-0.50324(1)	184,260	10h
Factored	-0.50321(1)	154,980	6h

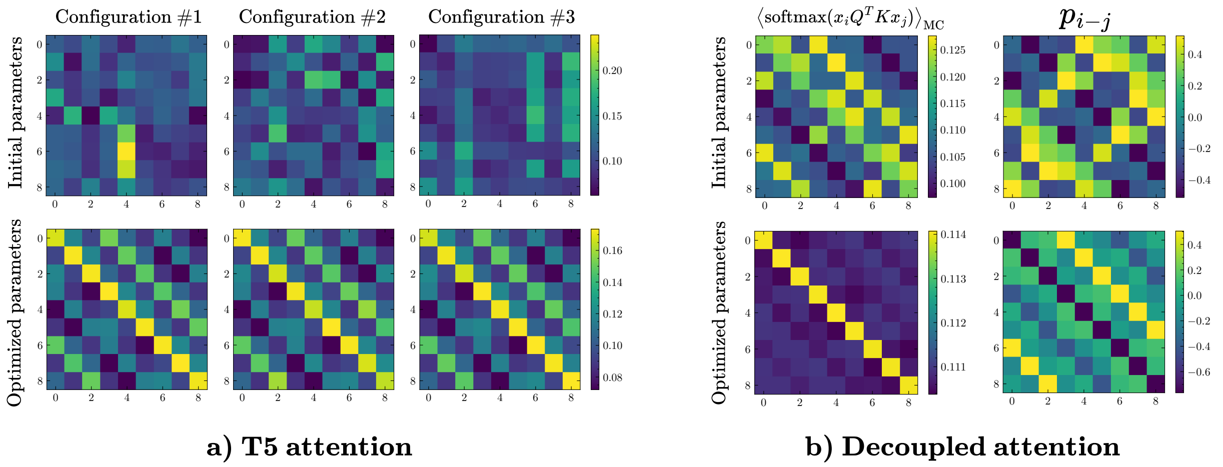

Figure 4.6: **Panel a:** Visualizations of the attention maps of a ViT with T5 attention mechanism [see Eq. (4.9)] for three different input spin configurations. When using initial random parameters there is a clear input dependence in the attention maps (top row). Instead, at the end of the optimization, the attention maps are practically input independent (bottom row). **Panel b:** Visualizations of the input-dependent term (left panels) and of the input-independent term (right panels) of a ViT with Decoupled attention mechanism [see Eq. (4.10)]. After the optimization (bottom row), the input-dependent term is approximately the identity matrix shifted element-wise by a constant, thus Factored attention is recovered [see Eq. (4.11)]. In the plots, the input-dependent term has been averaged over $M = 6000$ input configurations sampled from the optimized state. In both cases, the ViT architecture with a single layer $n_l = 1$, embedding dimension $d = 60$ and $h = 10$ different heads is optimized on a 6×6 lattice at $J_2/J_1 = 0.5$ (see panel (c) of Fig. 4.7). The plots are obtained by averaging the attention weights over all heads.

4.2.3.2 State-of-the-art ground state energy on the 10×10 lattice

After conducting a systematic study on the attention mechanism for the 6×6 lattice in the previous Section, where exact diagonalization calculations are feasible, the objective now is to approximate the ground state of the J_1–J_2 Heisenberg model at the highly frustrated point $J_2/J_1 = 0.5$ on the 10×10 square lattice, where the exact solution is not known. This challenging model has been extensively studied in recent years, and numerous calculations using various variational approaches are available (see Table 4.2).

Here, we consider as variational Ansatz the symmetrized ViT architecture in Eq. (4.8) using $b = 2$, $n_l = 8$ layers, embedding dimension $d = 72$, and $h = 12$ heads per layer. This variational state has in total 267720 real parameters (the complex-valued parameters of the output layer are treated as couples of independent real-valued parameters). Regarding the optimization protocol, we choose the learning rate $\tau = 0.03$ (with cosine decay annealing)

Table 4.2: Ground-state energy on the 10×10 square lattice at $J_2/J_1 = 0.5$ obtained with different variational states.

Energy per site	**Wave function**	**# parameters**	**Marshall prior**	**Reference**	**Year**
-0.48941(1)	MLP	893994	Not available	[156]	2023
-0.494757(12)	CNN	Not available	No	[128]	2020
-0.4947359(1)	Shallow CNN	11009	Not available	[11]	2018
-0.49516(1)	Deep CNN	7676	Yes	[10]	2019
-0.495502(1)	PEPS + Deep CNN	3531	No	[155]	2021
-0.495530	DMRG	8192 SU(2) states	No	[149]	2014
-0.495627(6)	aCNN	6538	Yes	[154]	2023
-0.49575(3)	RBM-fermionic	2000	Yes	[17]	2019
-0.49586(4)	CNN	10952	Yes	[143]	2023
-0.4968(4)	RBM ($p = 1$)	Not available	Yes	[153]	2022
-0.49717(1)	Deep CNN	106529	Yes	[152]	2022
-0.497437(7)	GCNN	67548	No	[151]	2023
-0.497468(1)	Deep CNN	421953	Yes	[150]	2022
-0.4975490(2)	VMC ($p = 2$)	5	Yes	[98]	2013
-0.497627(1)	Deep CNN	146320	Yes	[36]	2023
-0.497629(1)	RBM+PP	13132	Yes	[37]	2021
-0.497634(1)	**Deep ViT**	**267720**	**No**	**Present work**	**2023**

and the number of samples is fixed to be $M = 6000$. We emphasize that using Eq. (2.57) to optimize this number of parameters would be infeasible on available GPUs: the memory requirement would be more than $\sim 10^3$ gigabytes, one order of magnitude bigger than the highest available memory capacity. Instead, with the formulation of Eq. (2.62), the memory requirement can be easily handled by available GPUs (see Sec. 2.3.2). The simulations took four days on twenty A100 GPUs. Remarkably, as illustrated in Table 4.2, we are able to obtain the *state-of-the-art* ground-state energy using an architecture solely based on neural networks[9], without using any other regularization than the diagonal shift reported in Eq. (2.62), fixed to $\lambda = 10^{-4}$.

We stress that a completely unbiased simulation, without assuming any prior for the sign structure is performed. Furthermore, we verified with numerical simulations that the final results is not affected by the Marshall prior. Within variational methods, one of the main difficulties comes from the fact that the sign structure of the the ground state is not known for $J_2/J_1 > 0$. Indeed, the Marshall sign rule [34] gives the correct signs (for every cluster size) only when $J_2 = 0$ (see Appendix A). However, in order to stabilize the optimizations, many previous works imposed the Marshall sign rule as a first approximation for the sign structure (see *Marshall prior* in Table 4.2). By contrast, within the present approach, we do not need to use any prior knowledge of the signs, thus defining a very general and flexible variational *Ansatz*. This is an important point since

[9] During the revision process of our work in Ref. [27], we became aware of an updated version of Ref. [36] where a variational energy per site of -0.4976921(4) has been obtained.

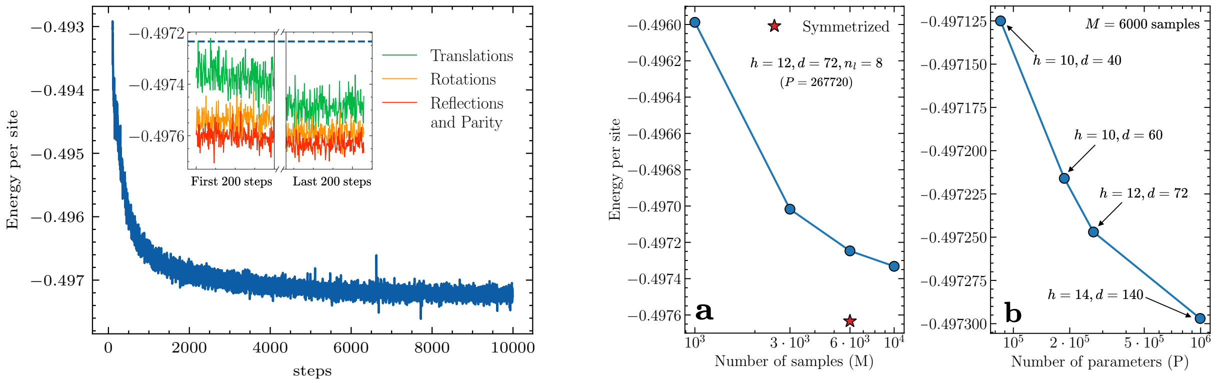

Figure 4.7: **Left panel:** Optimization of the Deep ViT with patch size $b = 2$, $n_l = 8$ layers, embedding dimension $d = 72$ and $h = 12$ heads per layer, on the J_1-J_2 Heisenberg model at $J_2/J_1 = 0.5$ on the 10×10 square lattice. The first 200 optimization steps are not shown for better readability. Inset: first and last 200 optimization steps when recovering sequentially the full translational (green curve), rotational (orange curve) and reflections and parity (red curve) symmetries. The total number of steps after restoring the symmetries is 5000 for translations, 5000 for rotations and 4000 for reflections and parity. The mean energy obtained without quantum number projection is also reported for comparison (blue dashed line). **Right panel:** Panel **a**: Energy per site as a function of the number of samples M for a ViT with $n_l = 8$ layers, embedding dimension $d = 72$ and $h = 12$ heads per layer. Panel **b**: Energy per site as a function of the number of parameters P, increased by adding heads h and taking larger embedding dimensions d, for a fixed number of layers $n_l = 8$. For both panels, the energy values (blue circles) are obtained without restoring the symmetries; for comparison we also show the energy corresponding to the fully symmetrized state in Eq. (4.8) (red star) which is the one reported in Table 4.2.

a simple sign prior is not available for the majority of the models (e.g., the Heisenberg model on the triangular or Kagome lattices). Moreover, we would like also to stress that the definition of a suitable architecture is fundamental to take advantage of having a large number of parameters. Indeed, while a stable simulation with a simple regularization scheme (only defined by a finite value of λ) is possible within the ViT wave function, other architectures require more sophisticated regularizations. For example, to optimize Deep GCNNs it is necessary to add a temperature-dependent term to the loss function [151] or, for Deep CNNs, a process of variance reduction and reweighting [36] helps in escaping local minima. We also point out that physically inspired wave functions, as the Gutzwiller-projected states [98], which give a remarkable result with only a few parameters, are not always equally accurate in other cases.

In the left panel of Fig. 4.7 we show a typical optimization on the 10×10 lattice at $J_2/J_1 = 0.5$. First, we optimize the Transformer having translational invariance among patches (blue curve). Then, starting from the previously optimized parameters, we restore sequentially the full translational invariance (green curve), rotational symmetry (orange curve) and lastly, reflections and spin parity symmetry (red curve). Whenever a new symmetry is restored, the energy reliably decreases [142]. We stress that our optimization process, which combines the SR formulation of Eq. (2.62) with a real-valued ViT followed by a complex-valued fully connected output layer [28], is highly stable and insensitive to the initial seed, ensuring consistent results across multiple optimization runs.

At the end, we discuss the impact of the number of samples M and parameters P on the variational energy wave function, showing the results in the right panel of Fig. 4.7. Specifically, in panel (b), we show the variational energy as a function of the number of parameters for a fixed number of layers $n_l = 8$, performing the optimizations with $M = 6000$ samples. The number of parameters is increased by enlarging the width of each layer. In particular, we take the following architectures: $(h = 10, d = 40)$, $(h = 10, d = 60)$, $(h = 12, d = 72)$, and $(h = 14, d = 140)$ with $P = 85400$, 187100, 267720, and 994700 parameters, respectively. Instead, in panel (a), we fix an architecture $(h = 12, d = 72)$ with $n_l = 8$ and increase the number of samples M up to 10^4. Both analyses are performed without restoring the symmetries by quantum number projection; for comparison, we report in the left panel the energy obtained after restoring the symmetries [see Eq. (4.8)]. The latter one coincides with the ViT wave function used to obtain the energy reported in Table 4.2.

In summary, we have introduced a novel approach, motivated by Representation Learning, to define variational states based on neural networks. The key feature of this method is its ability to map physical configurations into a real feature space, where it is then easy to predict amplitudes, even with a single fully-connected layer. Looking at NQS as feature extractors is an original contribution, compared to the common interpretation of them as just universal function approximators, which often involves complex-valued parameters and leads to optimization challenges in deep architectures. Crucially, we demonstrate the effectiveness of this approach by achieving state-of-the-art results on the J_1-J_2 Heisenberg model on the 10×10 square lattice, a widely recognized benchmark model for frustrated spin models on two-dimensional lattices.

4.3 Hidden representations

The composition defined in Eq. (4.1) plays a crucial role in determining the accuracy of our results. Here, we focus on the ability of the ViT state to automatically construct, during the minimization of the variational energy, physically meaningful hidden representations. Firstly, we show the hidden representation obtained at the end of the optimization process for a Heisenberg model and discuss its connection to the physical properties of its ground state [28]. Subsequently, we apply a technique referred to as *fine tuning*, demonstrating 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 [39].

4.3.1 A case study on the two-dimensional Heisenberg model

For a given set of configurations $\{\sigma_i\}_{i=1}^{M}$ (sampled along the Monte Carlo procedure), we compute the corresponding hidden vectors $\{\mathbf{z}_i\}_{i=1}^{M}$ of size $d \gg 1$, which can be visualized in two dimensions after a dimensional reduction. For this task, we apply the standard Uniform Manifold Approximation and Projection (UMAP) [161]. An exemplification of this approach is easily given in the limit $J_2 = 0$, where the system in Eq. (4.4) reduces to the (unfrustrated) Heisenberg model for which the exact sign structure of the ground state is known from the Marshall-sign rule [34]. In Fig. 4.8, we assign to each $\mathbf{z}_i$ a color representing the exact sign of the amplitude corresponding to the spin configuration σ_i. For random parameters, no discernible structure is apparent (see panel (a) of Fig. 4.8). Then, along with the minimization of the variational energy, the ViT learns automatically how to map the input configurations into different clusters of the hidden space, according to their amplitudes (see panel (b) of Fig. 4.8). In particular, the spin configurations in a given cluster have the same number of flipped spins with respect to the Néel one and, therefore, the same sign (according to the Marshall rule) and similar modulus. The crucial point is that, by using a single fully-connected layer, the prediction of the correct amplitudes is much easier when acting on these representations rather than using the original spin configurations. This result confirms that the Representation Learning approach for constructing variational wave functions performs as anticipated. Specifically, the deep neural network determines a set of features that simplify the original problem, and then the output layer behaves analogously as the linear classifier used in classification tasks. This process mirrors the illustrative toy example described at the beginning of the

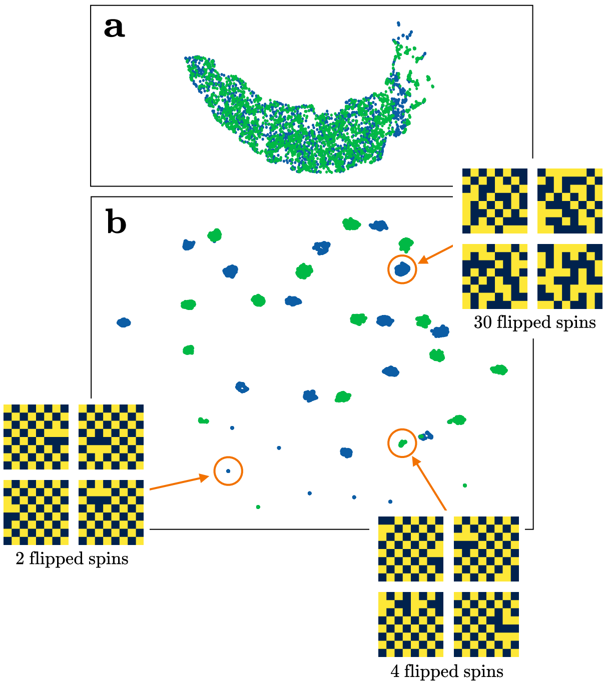

Figure 4.8: Dimensional reduction of the hidden representations for a set of configurations built using a ViT in the limit of $J_2 = 0$, leading to the Heisenberg model. Points are colored according to the exact signs given by the Marshall sign rule [34]. The calculations are performed on an 8×8 cluster. **Panel a:** Projections of the hidden representations built by a ViT with random parameters. **Panel b:** Projections built using the parameters after optimization of the variational energy.

Chapter (see right panel of Fig. 4.1).

4.3.2 Fine-Tuning Neural-Network Quantum States

In this Section, we address a key conceptual question that the Representation Learning approach raises: given a system that exhibits a phase transition, *do the representations 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, offering a concrete advantage by eliminating the need to optimize the wave function from scratch for each point in the phase diagram.

Given a system undergoing a phase transition, we want to investigate whether the representations learned near the transition point generalize to other points of the phase diagram. By referring to the composition in Eq. (4.1), we perform the following experiment in two steps, as illustrated in Fig. 4.9:

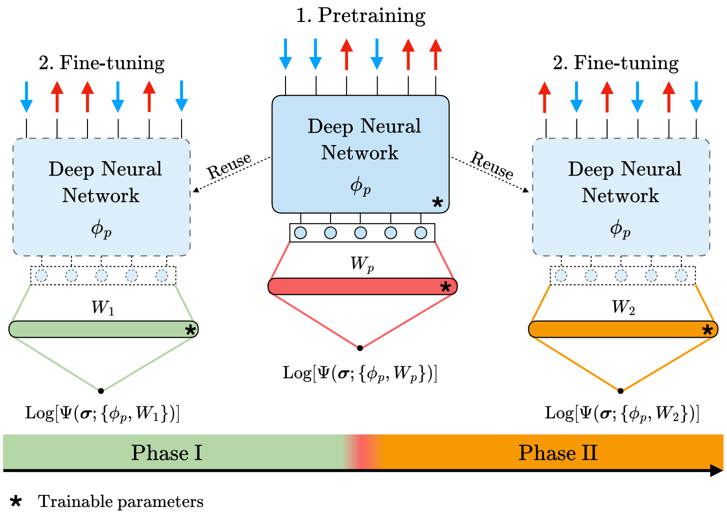

Figure 4.9: Graphical representation of the pretraining and fine-tuning procedures. Initially, during the pretraining, the entire architecture is trained in proximity to the transition point of a given system, yielding a set of parameters $\theta_p = \{\phi_p, W_p\}$. Subsequently, in the fine-tuning 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.

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, before and after the phase transition.

The pretraining of the architecture is carried out near the critical point, such that optimizing physical states with long-range correlations shape the representation learnt by the network. In the following, it is possible for the last (shallow) readout layer, which is fine-tuned in a different point in the phase diagram, to either reinforce long-range 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 contrary, in trivial phases, where only a few configurations have non-zero amplitudes, the ability of pretrained networks to generalize away from these phases is likely limited. 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 all the other points of the phase diagram [162]. 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 is generally applicable to any deep neural network, but to be concrete, in the following we employ the parametrization described in Sec. 4.2.1. Specifically, we parameterize the function $V(\sigma;\phi)$ using a Vision Transformer and the function $f(\boldsymbol{z}, W)$ is an RBM [7] defined in Eq. (4.2) 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.

We remark that this framework offers a huge computational advantage, since it requires the costly optimization of the full architecture, including the feature extractor $V(\sigma;\phi)$, only once in the pretraining step. Then, with the addition of a minimal cost, the targeted optimization of the output layer $f(\boldsymbol{z}, W)$ 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.

4.3.2.1 Numerical Results

For all the simulations in this Section we perform $N_{\text{opt}} = 10^4$ optimization steps during the pretraining stage. Then, during the fine-tuning stage, the number of steps is reduced to $N_{\text{opt}} = 3 \times 10^3$. In both stages the observables are estimate stochastically using $M = 3 \times 10^3$ configurations. The optimization of the variational parameters is performed with the Stochastic Reconfiguration (SR) method [69]. In particular, working with variational states featuring approximately $P \sim 10^6$ parameters, we employ the alternative formulation of SR [27, 36] efficient in the regime $P \gg M$ (see Sec. 2.3). We use a cosine decay scheduler for the learning rate, setting the initial value to $\tau = 0.03$.

Ising model in a transverse field. We start by considering the one-dimensional

Ising model in transverse magnetic field, described by the following Hamiltonian

$$\hat{H} = -\Gamma \sum_{i=1}^{N} \hat{S}_i^z \hat{S}_{i+1}^z - g \sum_{i=1}^{N} \hat{S}_i^x \ , \tag{4.12}$$

where $\hat{S}_i^x$ and $\hat{S}_i^z$ are spin-1/2 operators on site i. The ground-state wave function, for $g \geq 0$, 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 the panel (a) of Fig. 4.10, in comparison with density-matrix renormalization group (DMRG) [5] 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 Sec. 4.3.2.2 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 $\{\sigma_1, \dots, \sigma_M\} \sim |\Psi(\sigma; \theta_p)|^2$ from the pretrained network and show the corresponding amplitudes after the finetuning procedure (visualizing them on top of UMAP [161] projections of the hidden representations $\boldsymbol{z}_p(\sigma_i)$, for $i = 1, \dots, M$), see Fig. 4.11. 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

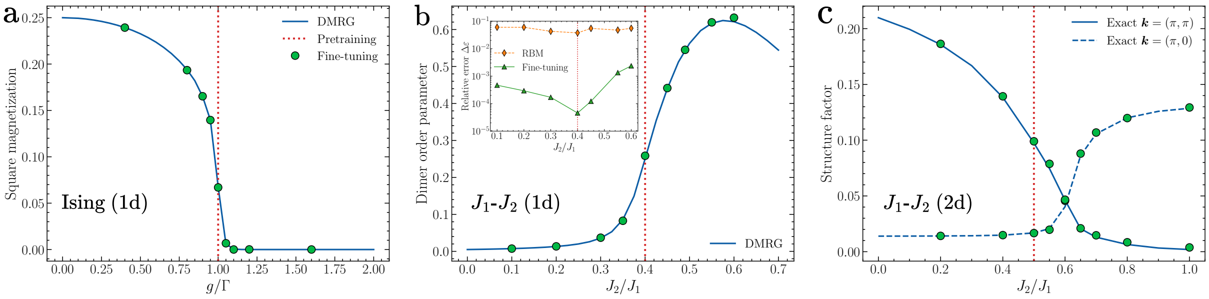

Figure 4.10: **Panel a:** Ising chain. A ViT with hyperparameters $h = 12$, $d = 72$, $n_l = 4$ is pretrained at $g/\Gamma = 1$, on a chain with $N = 100$ sites. After the fine-tuning, the square magnetization order parameter is computed and compared to DMRG results (bond dimension $\chi = 10^3$). **Panel 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. **Panel 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.

amplitudes for all other configurations are practically negligible (left panel). This effect is related to the difficulty of simple sampling (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).

Moreover, we can connect 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 then compute the hidden representations for fixed batch of physical spin configurations and then we perform *Principal Component Analysis* (PCA) on it. In the panel (a) of Fig. 4.12, 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.

J_1-J_2 **Heisenberg model on a chain.** In order to assess the accuracy of our method on more complicated systems, specifically with non-positive ground states in

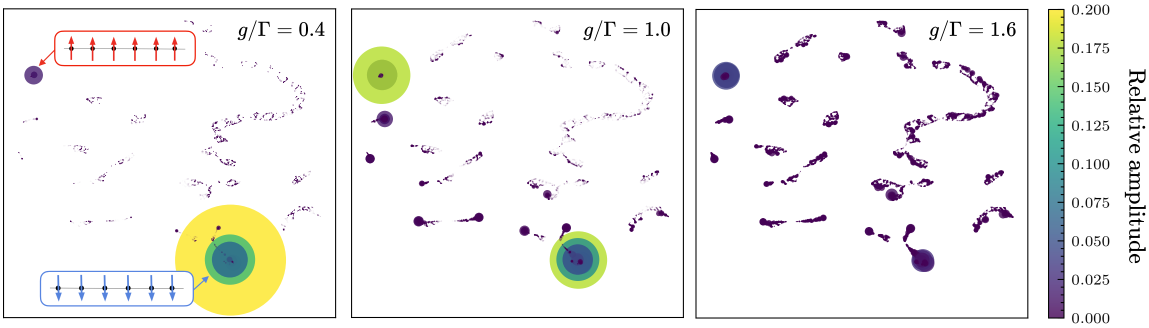

Figure 4.11: Dimensional reduction of the hidden representations for a set of $M = 3000$ configurations built using a ViT pretrained at $g/\Gamma = 1$ with hyperparameters $h = 12$, $d = 72$, and $n_l = 4$ for a system of $N = 100$ sites. The data points represent UMAP projections of vectors $\boldsymbol{z}$. 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).

the computational basis, we investigate the one dimensional frustrated J_1-J_2 Heisenberg model, whose Hamiltonians is defined in Eq. (3.3) (see Sec. 3.2.1). On finite systems, the phase transition between a gapless phase with no order whatsoever and a gapped one, with long-range dimer order may be extracted from the long-distance behavior of the dimer-dimer correlation functions defined in Eq. (3.23). Specifically, performing a finite-size scaling, an estimation of the dimer order parameter can be obtained as $\mathcal{D}^2 = 9|D(N/2-1) - 2D(N/2) + D(N/2+1)|$ [131, 163]. 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 [110]). 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, both in the gapless and gapped regions. The results for $N = 100$ (with periodic boundary conditions) are reported in the panel (b) of Fig. 4.10, again compared to DMRG calculations on the same system. In Table 4.3, we report the ground state energies obtained through three distinct methodologies: DMRG (with a bond dimension up to $\chi = 10^3$), 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 a relative error $\Delta\varepsilon \lesssim 10^{-3}$ for all the values of the frustration ratio in the interval $J_2/J_1 \in [0.1, 0.6]$. In addition, in the

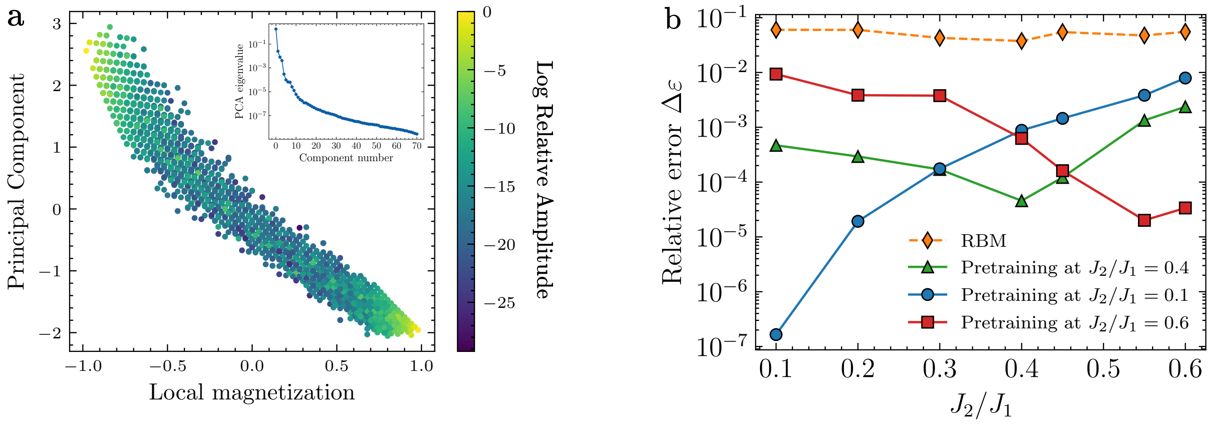

Figure 4.12: **Panel a:** Correlation between the local magnetization $\sum_{i=1}^{N} \sigma_i$ and the principal component of the hidden representations of the configurations associated to the ViT used to obtain the results for the Ising model in transverse field. In the inset, the PCA spectrum is shown. **Panel b:** Relative error $\Delta\varepsilon$ of the energy with respect to DMRG for the J_1-J_2 Heisenberg model on a chain [see Eq. (3.3)] 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), $J_2/J_1 = 0.6$ (red squares). The accuracy of the same fully-connected network (RBM) optimized on the physical configurations is also reported for comparison (orange diamonds).

inset of the panel (b) of Fig. 4.10, we compare the relative energy error $\Delta\varepsilon$ (with respect to the DMRG energies) of an RBM trained directly on the physical configurations [7] and of the fine-tuned ViT. This result 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.

Furthermore, we want to stress that the accuracy of the fine-tuning across various points on the phase diagram is influenced by the choice of the pretraining point. In our calculations, we have always pretrained near transition points, where we expect better generalization properties as discussed previously. 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 the panel (b) of Fig. 4.12, we show the accuracy of the energy $\Delta\varepsilon$ relative to DMRG calculations. 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. 4.10). 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). Conversely, pretraining from $J_2/J_1 = 0.1$ (blue circles) yields higher accuracy before $J_2/J_1 = 0.4$, but as the distance from the pretraining point increases the accuracy deteriorates 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). 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 finetuning 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.

Let us move on 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) [34] applies. However, even for large system sizes, the MSR constitutes an accurate approximation of the sign structure up to $J_2/J_1 \leq 0.5$ [38]. In the panel (a) of Fig. 4.13, we show the predicted phases (0 or π), on top of the UMAP projections of the vectors $\boldsymbol{z}_p$ generated by the pretrained network at $J_2/J_1 = 0.4$. At $J_2/J_1 = 0.1$ (see the left panel), 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

Table 4.3: Variational ground state energies for the J_1-J_2 Heisenberg chain with system size $N = 100$. The Monte Carlo error attributed to finite sampling effects in the ViT wave functions 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

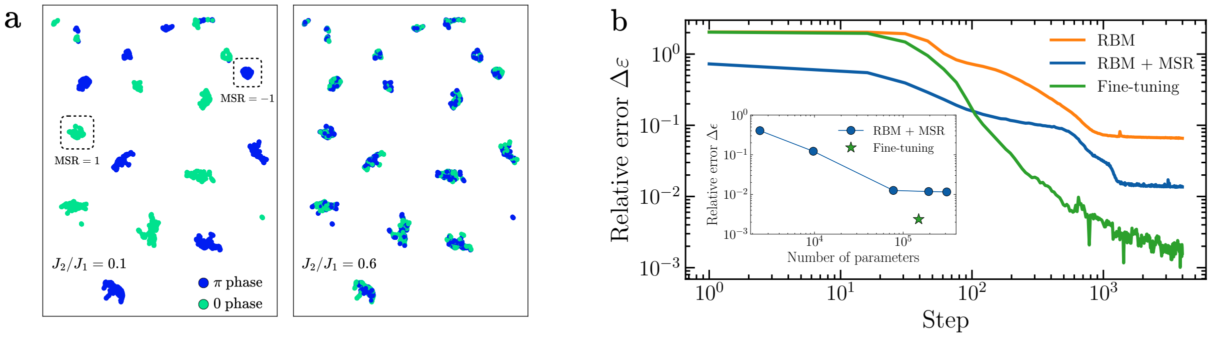

Figure 4.13: **Panel a:** Graphical representation of the hidden representations for the J_1-J_2 Heisenberg chain. The data points, corresponding to a sample of $M = 3000$ physical configurations, represent UMAP projections of vectors $\boldsymbol{z}_p$ generated by a ViT with hyperparameters $h = 12$, $d = 192$, 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: $J_2/J_1 = 0.1$ (left panel) and $J_2/J_1 = 0.6$ (right panel). The left panel reveals a close resemblance between the cluster structure identified during the pretraining at $J_2/J_1 = 0.4$ which match the Marshall sign rule. **Panel b:** 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. (3.3)] 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 77568 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 finetuned network is also shown.

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 the right panel); still, the fine-tuned ViT performs better than a RBM trained on spin configurations.

At the end, in order to understand which kind of prior information is encoded in the features generated by the pretrained network we focus on the frustration ratio $J_2/J_1 = 0.6$ and we study it with a RBM [see Eq. (4.2)]. This network is employed in two distinct manners: trained directly on the physical configurations σ, 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 the panel (b) of Fig. 4.13, 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 the panel (b) of Fig. 4.13).

J_1-J_2 Heisenberg model on the square lattice. Finally, we consider the two-dimensional J_1-J_2 Heisenberg model on an $L \times L$ square lattice (see Sec. 4.2.2). 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). In the panel (c) of Fig. 4.10 we show the results obtained by first performing the pretraining at $J_1/J_2 = 0.5$, then the fine-tuning for $0.2 < J_2/J_1 < 1$ and evaluating the order parameters $m^2_{\text{Néel}}$ and m^2_{stripe} defined in Eq. (4.6) and Eq. (4.7), respectively. 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 deep neural network.

4.3.2.2 Memory Efficiency in Fine-tuning and Pretraining Processes

In this Section, we discuss the advantages of the fine-tuning procedure in terms of memory requirements, compared to performing optimizations from scratch. 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 [133], that 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 $\theta = \text{Concat}(\theta_0, \dots, \theta_{n_l})$, where θ_l is a vector containing all the P_l parameters of the l-th 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 contrary, 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. [79]. 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 [27]. 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.

Chapter 5

Emergence of a Spin-Liquid Phase in the Shastry-Sutherland Model

In most cases the use of Neural-Network Quantum States has been limited to rather simple models, where the exact solutions were already known from other methods (e.g., the unfrustrated Heisenberg model on the square lattice or one-dimensional systems). Attempts to address challenging cases have been pursued, but without addressing important open questions on the ground-state properties. In this Chapter, we aim to push the boundaries by demonstrating that an *Ansatz* exclusively reliant on neural networks enables us to achieve unprecedented accuracy in solving the challenging Shastry-Sutherland model. This model poses a particularly demanding problem in the realm of highly-frustrated magnetism. Leveraging an architecture based on the Transformer architecture within the *Representation Learning* framework, we carry out simulations on $L \times L$ clusters, up to $L = 18$, with periodic-boundary conditions. Our results reveal the existence of a small, but finite, region in the phase diagram in which both the antiferromagnetic and plaquette order parameters vanish in the thermodynamic limit. As a result, this region is consistent with the existence of a spin-liquid state.

5.1 The Shastry-Sutherland Model

Among the variety of quantum spin models, the one introduced by Shastry and Sutherland [164] deserves particular attention since it gives an example in which the magnetic order can be melted by tuning the super-exchange interactions, eventually leading to a par-

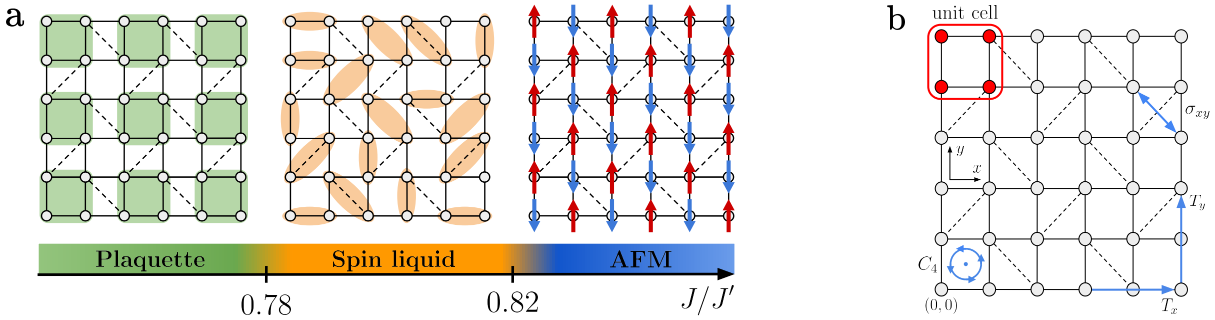

Figure 5.1: **Panel a:** The ground-state phase diagram of the Shastry-Sutherland model as obtained in this work. The super-exchange J (J') is denoted by solid (dashed) lines. **Panel b:** The (nearest-neighbor) coupling J is denoted by solid lines and (next-nearest-neighbor one) J' by dashed lines. The standard unit cell contains 4 sites, implying translations T_x and T_y (along x and y axis) by 2 lattice points. The point-group symmetries, C_4 rotations and σ_{xy} reflection, are also shown.

ticularly simple ground-state wave function, where nearby spins form singlets[10]. Most importantly, this Hamiltonian captures the low-temperature properties of $SrCu_2(BO_3)_2$ [165, 166]. The main interest in this material comes from its properties when external magnetic fields are applied. Indeed, a complicated magnetization curve is observed, with various magnetization plateaus (most notably at magnetization 1/8) that show intriguing properties [165, 167–169]. The Shastry-Sutherland model is defined by

$$\hat{H} = J \sum_{\langle \boldsymbol{R},\boldsymbol{R'} \rangle} \hat{\boldsymbol{S}}_{\boldsymbol{R}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R'}} + J' \sum_{\langle\langle \boldsymbol{R},\boldsymbol{R'} \rangle\rangle} \hat{\boldsymbol{S}}_{\boldsymbol{R}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R'}} \tag{5.1}$$

where $\hat{\boldsymbol{S}}_{\boldsymbol{R}}$ is the $S = 1/2$ operator on the site $\boldsymbol{R} = (x, y)$. Here, the first sum goes over nearest-neighbor sites on the square lattice, while the second sum is over next-nearest-neighbor sites on orthogonal dimers, according to the bond pattern of Fig. 5.1. For a detailed description of the lattice structure, including its symmetries, refer to Section 5.2.1. The ground-state properties of the Shastry-Sutherland model are well known in two limiting cases. When $J = 0$, the model reduces to a collection of decoupled dimers and its ground state is a product of singlets connected by J'; this state remains the exact ground state also for finite values of J/J', up to a certain value [164]. In the opposite limit, when $J' = 0$, the Heisenberg model on the square lattice is recovered, whose ground state is the Néel antiferromagnet; also in this case, the ground state is robust in a finite region when $J' > 0$. Despite the substantial effort that has been invested in understanding the appearance of magnetization plateaus, the ground-state properties of the Shastry-Sutherland

[10] See Appendix C for a detailed discussion on the dimer state of the Shastry-Sutherland model.

model have been investigated in much less depth. One of the first studies based on the mean-field approximation predicted an intermediate helical phase between the dimer and the Néel phases [170], while other works suggested a direct transition between these two phases [166, 171]. Later, an intermediate phase with plaquette order has been found by series expansion approaches [172] and confirmed within the generalization to $Sp(2N)$ symmetry and large-N expansion [173], by exact diagonalizations, and a combination of dimer- and quadrumer-boson methods [174]. Subsequent tensor-network approaches have corroborated the presence of the plaquette phase, for $0.675 \lesssim J/J' \lesssim 0.765$ [175]. This phase breaks the reflection symmetry across the lines containing the J' bonds (leading to a two-fold degenerate ground state) and is described by resonating singlets on half of the plaquettes with no J' bonds, see panel (a) of Fig. 5.1. The stabilization of plaquette order in $\mathrm{SrCu_2(BO_3)_2}$ has been obtained when hydrostatic pressure is applied, even though there is evidence that the broken symmetry is related to the fourfold rotations around the center of plaquettes with no J' bonds [176, 177]. In addition, high-pressure thermodynamics provided evidence of a deconfined quantum critical point between the Néel and plaquette phases [178]. The latter aspect has been supported by a numerical analysis, also suggesting the emergence of the $O(4)$ symmetry at the critical point [97, 179]. However, recent density-matrix renormalization group (DMRG) and exact diagonalization calculations [180, 181] pushed forward the idea that a spin liquid intrudes between the antiferromagnetic and plaquette phases, around $0.79 \lesssim J/J' \lesssim 0.82$. The existence of an intruding spin-liquid phase has been also suggested by renormalization group calculations [182].

5.2 Numerical Results

Numerical methods have proven crucial to obtain a description of the physical properties of the Shastry-Sutherland model. In this Section we show the results obtained employing the ViT architecture in the *Representation Learning* framework as described in Chapter 4.

5.2.1 Lattice and symmetries

The Shastry-Sutherland lattice is shown in the panel (b) of Fig. 5.1, where each site is labeled by the Cartesian coordinate $\boldsymbol{R} = (x, y)$, with $x, y \in \mathbb{Z}$. The lattice is invariant under translations $T_x : (x, y) \to (x + 2, y)$ and $T_y : (x, y) \to (x, y + 2)$. This symmetry can be easily encoded in the Transformer architecture by taking as input patches the four spins in an empty plaquette (i.e., plaquettes with no J' bonds), which constitute the unit cell and then choosing the translationally invariant attention weights, namely $\alpha_{i,j} = \alpha_{i-j}$. In addition, the lattice is invariant under the rotation with respect to the center of the empty plaquette at the origin of the lattice $R_{\pi/2} : (x, y) \to (-y + 1, x)$ and the diagonal reflection $\sigma_{xy} : (x, y) \to (y + 1, x - 1)$. For the ground state which lies in the $\boldsymbol{k} = (0, 0)$ sector, all these symmetries can be enforced by a projector operator (see Appendix B), leading to a total-symmetric state [37, 106, 183]:

$$\tilde{\Psi}_\theta(\sigma) = \sum_{r,R} \Psi_\theta(rR\sigma), \tag{5.2}$$

where $r \in \{\mathbb{I}, \sigma_{xy}\}$ and $R \in \{\mathbb{I}, R_{\pi/2}, R^2_{\pi/2}, R^3_{\pi/2}\}$. Notice that the sum in Eq. (5.2) is over a fixed number of terms and does *not* scale with the size of the system. In general this procedure gets an improvement in the accuracy of the variational state, which is difficult to obtain by just increasing the number of variational parameters. The numerical simulations shown in this Chapter are performed with the symmetrized state in Eq. (5.2). Furthermore, the Monte Carlo sampling for obtaining the ground state can be limited in the $S^z = 0$ sector due to the $SU(2)$ symmetry of the Shastry-Sutherland model.

5.2.2 Benchmarks

In order to validate our approach, we compare the results obtained by the ViT wave function with those obtained by exact diagonalizations on a small 6×6 cluster. Specifically, we focus on the challenging point $J/J' = 0.8$. We first examine the accuracy of the variational energies while varying the hyperparameters of the neural network. In Fig. 5.2a, we present the relative energy error as a function of the number of parameters, distributed in two different ways within the architecture. Initially, we maintain a single layer ($n_l = 1$) and increase the number of heads h and embedding dimension d. Subsequently, we fix a specific width ($h = 12$ and $d = 72$) and increment the number of layers from $n_l = 2$ to $n_l = 16$. We emphasize that to effectively use deep neural networks with nearly one million of parameters (see panel (a) of Fig. 5.2), we adopt the modern formulation of the Stochastic Reconfiguration [27, 36] (detailed in Chapter 2) taking $\tau = 0.03$ with a cosine decay scheduler, the regularization parameter $\lambda = 10^{-4}$ and the number of samples is fixed to be $M = 6000$. The energies for different values of n_l are reported in Table 5.1. Previous works [27, 36, 135–137] emphasized that, for two-dimensional frustrated systems, the use

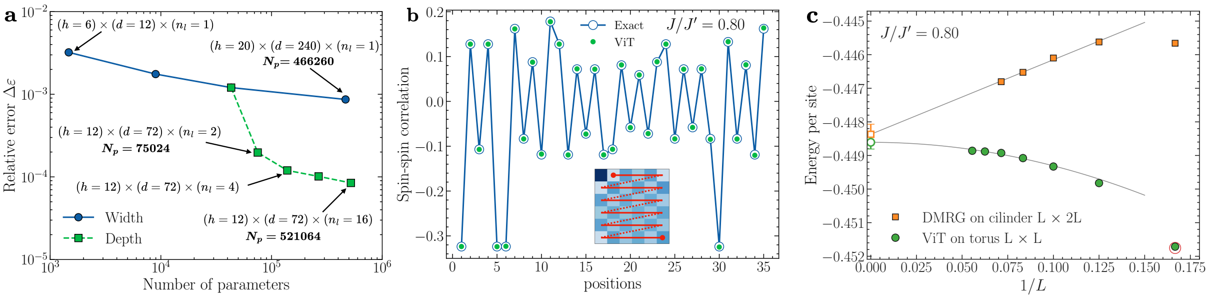

Figure 5.2: **Panel a:** Relative error $\Delta\varepsilon = |(E_{\text{exact}} - E_{\text{ViT}})/E_{\text{exact}}|$ of the ViT wave function on a 6×6 lattice at $J/J' = 0.8$. First, fixing only one layer and measuring the accuracy by increasing the width (blue dots). Then, for a fixed width, by increasing the number of layers (green squares). **Panel b:** The isotropic spin-spin correlations in real space as computed by the ViT wave function (full dots) on a 6×6 lattice at $J/J' = 0.8$. Values from exact diagonalization (empty dots) are also shown for comparison. Inset: The red line shows how the spin-spin correlations are ordered in the panel (b). **Panel c:** The comparison between the energies per site obtained by the ViT wave function (green circles) on $L \times L$ lattices with periodic-boundary conditions and the ones obtained by DMRG (orange squares) on $2L \times L$ cylinders with open-boundary conditions along the x direction [180]. The exact result on the 6×6 lattice is denoted with an empty red circle.

of deep neural networks is imperative to attain precise results. In fact, for an equivalent number of parameters, architectures distributing parameters across multiple layers exhibit superior accuracy. In addition, the comparison of isotropic spin-spin correlation functions $\langle \hat{\boldsymbol{S}}_{\boldsymbol{0}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R}} \rangle$ with exact results (see Fig. 5.2b) illustrates that our variational wave function not only yields accurate energies, but also faithfully correlation functions at all distances. For cluster sizes exceeding $L = 6$, exact results become unattainable. Consequently, in Fig. 5.2c, we compare the variational energies of the ViT *Ansatz* on $L \times L$ clusters (with periodic-boundary conditions) to the ones obtained using the DMRG method on $L_x \times L_y$ cylinders with open and periodic boundaries in the x and y direction, respectively ($L_x = 2L_y$ and $L_y = L$ are considered) [180]. Due to the global receptive field of the attention mechanism, its computational complexity scales quadratically with respect to the length of the input sequence. To improve the efficiency of the variational states for large systems with $L = 16$ and $L = 18$, we consider a *local* attention with a 5×5 filter and architectures with $n_l = 4$ and $n_l = 8$ layers to have a structure that connects all patches, albeit indirectly. The actual energies for $L = 14, 16$ and 18 are reported in Table 5.1

We mention that the energies obtained by the ViT wave function reveal a $1/L^2$ term as the leading correction, whereas the DMRG results exhibit an additional $1/L$ term. Most importantly, the energy extrapolated in the thermodynamic limit is compatible within the two approaches. In addition, we emphasize that this approach allows us to achieve larger sizes than are currently feasible with state-of-the-art methods such as the DMRG.

Table 5.1: Ground-state variational energy (in unit of J') for different number of layers n_l at $J/J' = 0.8$. The Monte Carlo error due to finite sampling effects is on the last digit. In the case of a 6×6 lattice, the ground-state energy per site from exact diagonalization is $E = -0.4517531$.

	4 layers	8 layers
6×6	-0.451699	-0.451707
14×14	-0.448839	-0.448925
16×16	-0.448822	-0.448882
18×18	-0.448813	-0.448859

5.2.3 Phase diagram

Having proved the high accuracy of our *Ansatz*, we now focus on the region $0.7 \leq J/J' \leq 0.9$, which is expected to include both antiferromagnetic and plaquette phases, as well as the putative spin-liquid one. All calculations are done on $L \times L$ clusters with $L \leq 18$. The presence of antiferromagnetic order is extracted from the thermodynamic limit of the staggered magnetization $m^2(L) = S(\pi, \pi)/L^2$ [180], where

$$S(\boldsymbol{k}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} \langle \hat{\boldsymbol{S}}_{\boldsymbol{0}} \cdot \hat{\boldsymbol{S}}_{\boldsymbol{R}} \rangle \tag{5.3}$$

is the spin structure factor. Notice that $S(\boldsymbol{k})$ is defined by the Fourier transform on the square lattice denoted by the sites $\boldsymbol{R}$, i.e., *without* considering the basis of the Shastry-Sutherland lattice. In addition, the insurgence of the plaquette order is detected by a suitably defined order parameter

$$m_p(L) = |C(L/2, L/2) - C(L/2 - 1, L/2 - 1)| \ , \tag{5.4}$$

where the function $C(\boldsymbol{R})$ is defined as follows: starting from the operator $\hat{P}_{\boldsymbol{R}}$, which performs a cyclic permutation of the four spins of a plaquette with the top-right site at $\boldsymbol{R}$ [180], the following correlation functions are evaluated:

$$C(\boldsymbol{R}) = \frac{1}{4} \langle [\hat{P}_{\boldsymbol{R}} + \hat{P}_{\boldsymbol{R}}^{-1}][\hat{P}_{\boldsymbol{0}} + \hat{P}_{\boldsymbol{0}}^{-1}] \rangle \ . \tag{5.5}$$

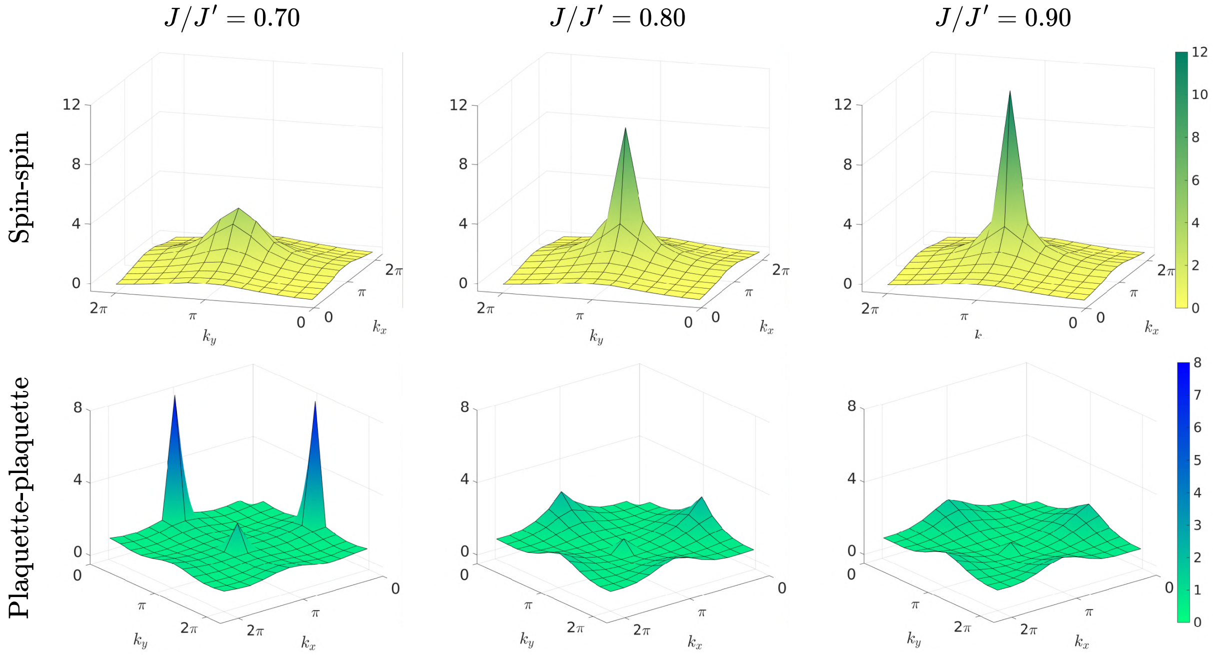

Figure 5.3: Fourier transform of the spin-spin (upper panels) and plaquette-plaquette (lower panels) correlations for $L = 12$ for different values of the frustration ratio J/J'. The calculations are performed with a Vision Transformer characterized by a number of heads equal to $h = 12$, an embedding dimension $d = 72$, and number of layers $n_l = 8$.

Therefore, the plaquette order parameter $m_p(L)$ of Eq. (5.4) measures the difference, along the diagonal, of the plaquette correlation at the maximum distance and the second maximum distance; whenever the plaquette order is present, the correlation along the diagonal does not decay to zero, implying a non-vanishing value of $m_p(L)$ for large L. Similarly, the Fourier transform of the correlation functions in Eq. (5.5) (with the same conventions as for spins) denoted by $C(\boldsymbol{k})$ can be analyzed. The presence of the plaquette order can be identified by a diverging peak at $\boldsymbol{k_p} = (0, \pi)$ or $(\pi, 0)$. The results for $L = 12$ are shown in Fig. 5.3, for three values of the frustration ratio: for $J/J' = 0.7$ the ground state has strong peaks in $C(\boldsymbol{k})$ and a rather smooth spin structure factor $S(\boldsymbol{k})$, which is typical of a state with plaquette order; by contrast, for $J/J' = 0.9$ there are strong spin-spin correlations and weak plaquette-plaquette ones, which is characteristic of antiferromagnetic states. In between, for $J/J' = 0.8$, the spin-spin correlations still have a peak, with moderate plaquette correlations. In order to get information on the thermodynamic limit, a size scaling is necessary. In general, if magnetic order is stabilized,

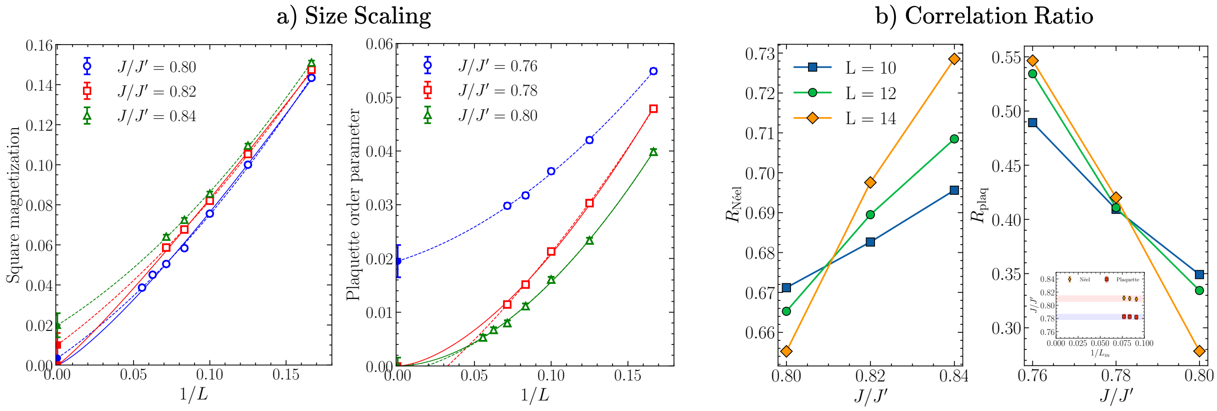

Figure 5.4: **Panel a:** Size scaling of the square magnetization $m^2(L)$ (left panel), and the plaquette order parameter $m_p(L)$ (right panel) as a function of $1/L$ from $L = 6$ up to $L = 14$. For the frustration ratio $J/J' = 0.80$ we include also $L = 16$ and 18. The values reported for each size L are obtained by extrapolating to an infinite number of layers. The error bars of the extrapolated values in the thermodynamic limit are estimated via a resampling technique with gaussian noise. The fits associated to dashed curves are obtained using second-order polynomials in $1/L$, while solid curves are obtained using the critical form in Eq. (5.7) of the main text. **Panel b:** In the left (right) panel we show the correlation ratio $R_{\text{Néel}}$ (R_{plaq}) for the antiferromagnetic (plaquette) order in the interval $J/J' \in [0.80, 0.84]$ ($J/J' \in [0.76, 0.80]$). System sizes from $L = 10$ to $L = 14$ are used. Inset: Crossing points of the correlation ratio for Néel (orange diamond) and plaquette (red squares) order parameter as a function of the system size. The crossing points are obtained using $L_1 \times L_1$ and $L_2 \times L_2$ clusters with $(L_1, L_2) = (10, 12), (10, 14), (12, 14)$, with $L_m = (L_1 + L_2)/2$.

the square magnetization scales asintotically as [147, 148]:

$$m^2(L) \approx m_0^2 + \frac{A_1}{L} + \frac{A_2}{L^2} \ , \tag{5.6}$$

where m_0 is the magnetization in the thermodinamic limit. In a disordered phase, the magnetization vanishes in the thermodynamic limit. The size corrections can be either exponential (for a gapped state) or power law (for a gapless one). In the vicinity of the Néel transition, the gap is relatively small and we use the "critical" form [37]:

$$m^2(L) \approx L^{-(1+\eta)} \ . \tag{5.7}$$

Similar scaling behaviors are considered for $m_p(L)$ (within the plaquette phase, exponential corrections should be present, but no appreciable differences in the fits are observed with respect to the choice of a polynomial fit). In Fig. 5.4a, we perform a size-scaling extrapolation of both order parameters. For $J'/J = 0.84$ ($J'/J = 0.76$), the numerical values of the square magnetization (plaquette order parameter) fit well with a second-order

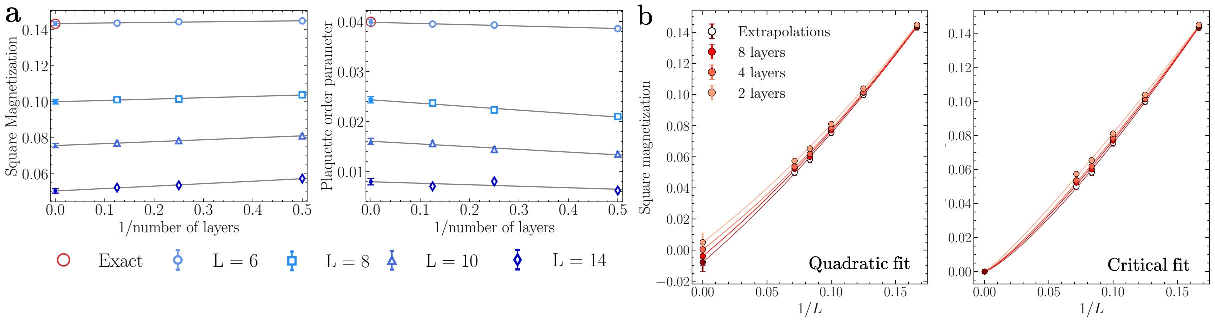

Figure 5.5: **Panel a:** The values of the square magnetization $m^2(L)$ (plaquette order parameter $m_p(L)$) are reported in the left (right) panel by increasing the number of layers n_l from 2 to 8 at $J/J' = 0.8$ from $L = 6$ to $L = 14$ (empty symbols). The extrapolated results in the limit of an infinite number of layers are also shown (full symbols). The exact values for $L = 6$ are reported in both cases for comparison (red circle). The error bars on the extrapolated values are obtained via resampling techniques with gaussian noise. **Panel b:** Size scaling of the square magnetization $m^2(L)$ as a function of $1/L$ from $L = 6$ up to $L = 14$ at $J'/J = 0.8$. The numerical data encompasses varying numbers of layers, specifically from $n_l = 2$ to $n_l = 8$, along with the extrapolated values for an infinite number of layers (see Fig. 5.5). The curves for the extrapolations in the thermodynamic limit are performed using as fitting curve a second-order polynomial in $1/L$ [see Eq. (5.6)] (left panel) and the critical form of Eq. (5.7) (right panel). The error bars of the extrapolated values in the thermodynamic limit are obtained with resampling techniques with gaussian noise.

polynomial in $1/L$ and suggest the existence of long-range order in the thermodynamic limit. By contrast, for $J'/J = 0.78, 0.8$ ($J'/J = 0.82, 0.84$), a more appropriate description of the scaling behavior of m^2 (m_p) is obtained by the critical relation of Eq. (5.7). Interestingly, fitting the data of the square magnetization at $J'/J = 0.8$ with $m^2 \approx L^{-(1+\eta)}$, we get $\eta \approx 0.3$, in agreement with the DMRG calculations of Ref. [180]. Remarkably, for the most challenging point $J/J' = 0.8$ we estimate the order parameters also for larger lattices, in particular $L = 16$ and $L = 18$ (see panel (a) of Fig. 5.4). These further calculations provide convincing evidence that for $J/J' = 0.8$ both order parameters vanish in the thermodynamic limit, strongly suggesting the existence of an intermediate phase compatible with a liquid spin phase [180]. The results in the panel (a) of Fig. 5.4 are obtained by measuring the order parameters as a function of the number of layers n_l, and then extrapolating their values for a network with an infinite number of layers $n_l \to \infty$ (see panel (a) of Fig. 5.5). For $L = 6$, we validate these numerical extrapolations against exact diagonalization results, finding excellent agreement. Additionally, the extrapolated results exhibit minimal deviation from the results for $n_l = 8$ layers, underscoring the robustness of the calculations.

Moreover, in the panel (b) of Fig. 5.5, we show the size scaling of the square magnetization in $1/L$ as the number of layers increases. Then we perform extrapolations in the thermodynamic limit, considering the square magnetization values obtained for a fixed number of layers n_l. Specifically, as cross validation we use both a second-order polynomial in $1/L$ [see Eq. (5.6)] and the critical form in Eq. (5.7) to carry out the extrapolations. The resulting fitting curves display remarkably similar behaviors, further confirming the consistency and reliability of our extrapolated results.

In summary, we find that the magnetization (plaquette order) vanishes for $J/J' \approx 0.82$ ($J/J' \approx 0.77$). These results suggest that a spin liquid exists between $(J/J')_{\text{plaq}} \approx 0.77$ and $(J/J')_{\text{Néel}} \approx 0.82$. To further support the present outcome, we measure the correlation ratio for the plaquette order as $R_{\text{plaq}} = 1 - C(\boldsymbol{k_p} + \delta\boldsymbol{k})/C(\boldsymbol{k_p})$, and for the magnetic order as $R_{\text{Néel}} = 1 - S(\boldsymbol{k_p} + \delta\boldsymbol{k})/S(\boldsymbol{k_p})$, where $||\delta\boldsymbol{k}|| = 2\pi/L$. When plaquette (magnetic) order is not present, $C(\boldsymbol{k})$ ($S(\boldsymbol{k})$) is a smooth function of $\boldsymbol{k}$, which implies that $R_{\text{plaq}} \to 0$ ($R_{\text{Néel}} \to 0$) in the thermodynamic limit; instead, when plaquette (magnetic) order settles down, $C(\boldsymbol{k})$ ($S(\boldsymbol{k})$) is finite for all the momenta except for $\boldsymbol{k_p}$, leading to $R_{\text{plaq}} \to 1$ ($R_{\text{Néel}} \to 1$). Then, the transition point may be accurately determined by locating the crossing point of the correlation ratio curves for different system sizes. The results for the plaquette (magnetic) order are shown in Fig. 5.4b, in the relevant interval $J/J' \in [0.76, 0.80]$ ($J/J' \in [0.80, 0.84]$), increasing the system size, i.e., for $L = 10$, 12, and 14. The various curves cross at $(J/J')_{\text{plaq}} \approx 0.78$ ($(J/J')_{\text{Néel}} \approx 0.81$), validating the phase boundary derived from the extrapolations of the order parameters.

5.2.4 Nature of the Spin Liquid Phase

The difficulty of the problem resides in the smallness of the spin liquid region, which require extremely accurate calculations and large system sizes. The present definition of the ViT wave function (that combines a real-valued attention mechanism and a final complex-valued fully-connected layer) allows us to detect the existence of a finite region $0.78 \lesssim J/J' \lesssim 0.82$ in which both magnetic and plaquette orders vanish in the thermodynamic limit, then supporting the presence of the intermediate spin-liquid phase [180]. Our results are important because they show that the magnetically ordered Néel phase is melted into a spin liquid, similar to what happens in the J_1-J_2 Heisenberg model on the square lattice [184]. This suggests that this kind of (continuous) transition is rather

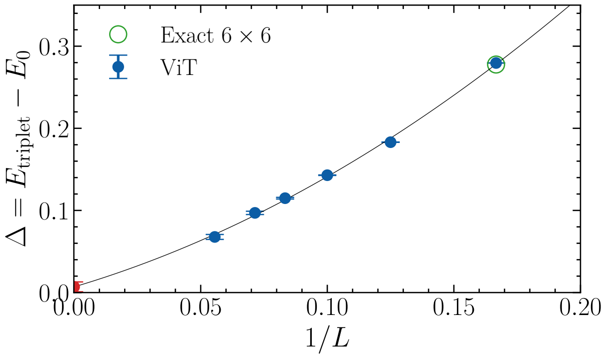

Figure 5.6: Energy gap Δ obtained with the ViT wave function between the ground state with total spin along the z direction $S^z = 0$ at momentum $\boldsymbol{k} = (0,0)$ and the excited state with $S^z = 1$ at momentum $\boldsymbol{k} = (0,0)$ as a function of inverse linear length $1/L$ from $L = 6$ to $L = 18$ at $J/J' = 0.8$. The exact gap for the 6×6 lattice is also reported for comparison (green empty circle).

generic and may represent the habit, and not the exception, for the melting of the Néel order due to magnetic frustration.

A further step in comprehending the physical properties of the Shastry-Sutherland model is the characterization of the spin liquid phase. Of particular significance is the question of whether the liquid spin state is *gapless* or *gapped*. However, the spin-liquid region is rather small and extracting the gap is extremely difficult. Indeed, the gap is expected to be small, thus implying that very large clusters must be considered. We focus on $J/J' = 0.8$, namely the most challenging point in the middle of the exotic spin liquid phase. We study the energy gap between the triplet state ($S = 1$) at momentum $\boldsymbol{k} = (0,0)$, which represents the lowest excitation in the Néel phase (within the folded Brillouin zone), and is expected to remain the lowest near the transition to the spin liquid phase, and the singlet state ($S = 0$) at momentum $\boldsymbol{k} = (0,0)$, which correspond to the symmetry sectors of the ground state.

As described in Sec. 5.2.1 the ViT wave function has by construction zero momentum by choosing as input patches the unit cell of the Shastry-Sutherland lattice. However, since the ViT wave function is constructed in a basis aligned with the z-axis, it explicitly breaks the $SU(2)$ symmetry [149, 180, 185–187]. For this reason, to approximate the triplet state we restrict the wave function to the sector of the Hilbert space with $S^z = 1$, which can be easily implemented in the Monte Carlo sampling. Instead for the approximation of the ground state we restrict the sampling to the $S^z = 0$ sector as detailed in Sec. 5.2.1. In Fig. 5.6 we show the energy gap $\Delta = E_{\text{triplet}} - E_0$ obtained with the ViT wave function as a function of inverse linear length $1/L$ from $L = 6$ to $L = 18$ at $J/J' = 0.8$. The extrapolation with a quadratic fit $\Delta = a + b/L + c/L^2$ produces, with a small fitting error, a vanishing gap $\Delta = 0.00(7)$ in the thermodynamic limit. This result provides, for the first time, compelling evidence about the *gapless* nature of liquid spin in the Shastry-Sutherland model [28].

Conclusions and Future Directions

The research work presented in this Thesis has focused on the characterization of frustrated spin models using variational approaches grounded on neural networks. Over the past three decades, this challenging task has been addressed through the introduction and development of several numerical methods. Although a comprehensive understanding of the physical properties of strongly correlated systems in the highly-frustrated regime remains an open challenge, we have demonstrated that our methods provide accurate results to previously inaccessible problems. This underscores the potential of Neural-Network Quantum States as a valuable tool for probing uncharted phases of matter, opening opportunities to establish the properties of many-body systems.

While NQS have achieved remarkable results in the unfrustrated regime [7], where the knowledge of the sign structure significantly simplified the numerical computations [34], the study of frustrated systems has posed considerable complications. In Chapter 3 we first looked into what happens when the ground state is not positive definite in the computational basis. Specifically, we wanted to understand how neural networks can figure out the sign structure of ground state and low-energy excited states. We started with a simple neural network architecture, namely a single fully-connected layer, analogous to the one utilized in the original work of Carleo and Troyer [7]. We focused on small clusters for the J_1-J_2 Heisenberg model on a chain, comparing our results with the exact ones. We checked how well this kind of neural network can understand the tricky sign structure of the ground state during training, especially in different frustration scenarios [38]. It turns out, while this simple architecture works well for small systems, it is not suitable for larger clusters. That is why we need to consider more complicated architectures. Specifically, we propose a variational state based on Transformer architectures [31], which are advanced models known for their effectiveness in natural language processing tasks [18]. Transformers are constructed to efficiently capture long-range correlations, a challenge for other types of neural networks. Describing these correlations accurately is crucial in understanding the physical properties of many complicated models. We tested our approach on the J_1-J_2 Heisenberg model on a chain with one hundred sites. Our results compete well with Density Matrix Renormalization Group calculations, indicating that our approach accurately describes ground state properties for both gapless and gapped states with incommensurate correlations.

However, using large-scale neural networks creates challenges for traditional optimization methods. Specifically, the Stochastic Reconfiguration, a powerful method for optimizing variational quantum states, becomes impractical when dealing with a large number of parameters. This is because it requires inverting a matrix with a side length equal to the number of parameters. To address this issue, in Chapter 2 we discussed a method that we recently proposed, that relies on a simple linear algebra identity. This identity reduces the problem to inverting a smaller matrix with a side length equal to the number of samples used for stochastic estimations [27]. This approach is particularly useful in deep learning scenarios where the number of parameters is much greater than the number of samples, enabling the efficient optimization of variational states with millions of parameters. In Chapter 4 we showcase the efficacy of this method by achieving *state-of-the-art* ground state energy for the J_1-J_2 Heisenberg model on a 10×10 lattice at $J_2/J_1 = 0.5$, a challenging benchmark in highly-frustrated magnetism, with a variational *Ansatz* based on a Deep Vision Transformer.

In Chapter 5 we further demonstrated the effectiveness of our methodology focusing on the Shastry-Sutherland lattice, a frustrated two-dimensional system that mimics the low-energy properties of $\mathrm{SrCu_2(BO_3)_2}$. This material exhibits remarkable behavior under an external magnetic field. Our study focused on determining the ground state properties. We found a region where the order parameters describing well-established orders, such as plaquette valence bond and antiferromagnetic orders, both approach zero in the thermodynamic limit. This evidence suggests the presence of a small region consistent with a spin liquid state, a conclusion also reached in a recent study using DMRG [180]. This result, achieved by combining state-of-the-art Neural-Network Quantum States with a powerful optimization technique, not only highlights the utility of neural networks as effective parametrizations for obtaining reliable benchmarks but also unveils new physical phenomena in complicated models.

Future directions can take various paths. On one hand, we are particularly interested in using these variational states for exploring more complicated problems, such as the Kagome lattice [35, 188]. Additionally, there is an interest in applying them for determining physical properties which are relevant from an experimental perspective, such as the magnetization plateaus in the Shastry-Sutherland lattice [169]. On the other hand, the success in describing quantum systems of spins through variational states based on neural networks has no counterpart in correlated models of fermions [189]. For example, the Hubbard model on the square lattice, which represents the most iconic model of interacting fermions, has been the subject of intense numerical and theoretical studies; nevertheless, a complete understanding of its properties has not been achieved yet [51, 190]. Considering the results obtained in studying one and two dimensional spin models with the Transformer architecture, we would like to extend this approach to deal with fermionic ones. To do that, a generalization of the architecture is necessary.

Generally, deep neural networks process inputs (like the physical configurations of a model) to create hidden representations in high-dimensional spaces. As presented in Chapter 4 their success lies in the simplicity of problem-solving within this space, often characterized by clusters that are linearly separable. Consequently, even simple networks operating in this representation can effectively tackle complicated problems. This philosophy guided our approach in defining the *Ansatz* for spin models [27].

In this case, a spin configuration is mapped into a vector $\boldsymbol{z} \in \mathbb{R}^d$ by the Transformer. Subsequently, a straightforward network can predict the amplitude of the corresponding configuration. For fermionic problems, an additional step is needed. Starting from the hidden representation $\boldsymbol{z}$, we construct a matrix of orbitals, and the final amplitude is determined by the determinant of the matrix. This approach offers an advantage as the obtained orbitals are configuration-dependent, similar to *backflow* methods [191]. In principle, a single determinant proves sufficient to construct highly accurate variational states by refining the hidden representations. Additionally, a similar strategy can be applied to address fermionic problems on the continuum with minor adjustments [102, 103, 107].

Moreover, from a methodological perspective, the approach introduced for optimizing networks with a large number of parameters [27], making minor modifications to the equations describing parameter updates, enables to describe the unitary time evolution of quantum many-body systems according to the *time-dependent variational principle* [75, 192–194]. Consequently it can find application in the determination of dynamical structure factors [108]. Specifically, the latter ones can be estimated as the Fourier transform of a dynamical correlation function $\langle\Psi_0|\hat{O}_1(t)\hat{O}_2|\Psi_0\rangle = \langle\Psi_0|e^{i\hat{H}t}\hat{O}_1e^{-i\hat{H}t}\hat{O}_2|\Psi_0\rangle$. Thus, it is essential to have an efficient way to accurately estimate $e^{-i\hat{H}t}\,|\Psi_0\rangle$. Dynamical structure factors offer the possibility to build a connection between theoretical/numerical results and experimental analysis such as inelastic neutron scattering. This makes them crucial methods for characterizing two dimensional quantum many-body systems. In particular they play an important role for the experimental detection of significant properties (e.g., fractionalized excitations) in candidate materials able to realize exotic phases of matter.

In conclusion, the results presented in this Thesis demonstrate that Neural-Network Quantum States are an effective tool for investigating the ground-state properties of frustrated quantum magnets, competing, and also surpassing, existing state-of-the-art numerical methods developed over the past three decades. We believe that future research focused on improving these techniques will be crucial for advancing our understanding of the physical properties of strongly interacting quantum systems.

Acknowledgements

Prima di tutto, vorrei ringraziare Federico (soprannominato *Torquemada* da qualche spiritoso studente che mi ha preceduto) per avermi sopportato in tutti questi anni. Il suo approccio, diretto e schietto, sia a livello personale che lavorativo, è stato molto formativo (e, devo ammettere, spesso molto invidiato dagli altri dottorandi). L'uso di metafore e di "esclamazioni colorite" hanno reso il dottorato un'esperienza unica e piena di aneddoti memorabili. Lo ringrazio per il quotidiano supporto e per le innumerevoli spiegazioni che mi hanno permesso di affrontare al meglio questo percorso. Con orgoglio, ammetto che è stato un vero onore essere definito lo studente più "rompicojoni" che abbia mai avuto!

Desidero inoltre ringraziare tutte le persone con cui ho avuto il piacere di collaborare in questi anni, che mi hanno fatto crescere sia dal punto di vista umano che professionale. Un ringraziamento va a Francesco Ferrari, per la sua immensa pazienza nei miei confronti; ad Alberto Parola, per la sua professionalità e il suo incredibile senso dell'umorismo; a Sebastian Goldt, per gli utili consigli che mi ha sempre offerto; e ad Alessandro Laio, per avermi trasmesso la sua profonda passione per la ricerca.

Vorrei poi ringraziare le persone che hanno reso il mio soggiorno triestino un'esperienza indimenticabile.

Un ringraziamento speciale va a Riccardo. Ho avuto il piacere di conoscerlo ben prima che diventasse il grande chef affermato che è oggi, noto a tutti per la celebre "pasta col pepe". Lo ringrazio per le tante discussioni scientifiche e non (gli amici piu audaci possono comprendere gli altri argomenti di discussione). Grazie per le risate, il continuo confronto e per tutti i weekend trascorsi insieme all'ICTP, che tutti ci invidiavano.

Un immenso grazie a Roberta. Ho sempre potuto contare su di lei, e so che sarà così anche in futuro. Sono passati quasi nove anni da quando ho iniziato a scroccarle pranzi e cene nelle campagne di Arcavacata, e nonostante abbia spesso messo a dura prova la sua pazienza, non mi ha mai abbandonato. Anche se sono consapevole di essere stato il miglior coinquilino che si potesse desiderare, devo ammettere che anche lei è stata un'ottima compagna di viaggio.

Un grazie a Vittorio, non si molla mai, nonostante i tantissimi swipes, le aspettive bassissime e le poche speranze. Ringrazio Fra Slongo per essermi amico nonostante la mia totale avversione per l'arrampicata e Mauro, per il suo grande umorismo e la sua invidiabile chioma bionda. A Zeno a la Lu per la grande gentilezza e disponibilità, sono impaziente di conoscere il nuovo membro della famiglia, il temibile pesce combattente. Ringrazio tutti loro anche per la grande ospitalità che mi hanno riservato negli anni, accogliendomi nel soggiorno con vista più esclusivo di Trieste.

Un doveroso ringraziamento ai *Kagopizza boys*. Più esplicitamente al maestro Giovanni, punto di riferimento spirituale e architetto del parcheggio selvaggio; al fervente napoletano Giulio, esperto di Barren Plateaus e stimato personal trainer e a Michele Vinsky, magnate del quanto e artista contemporaneo dotato di humour impararareggiabile.

Un ringraziamento a Leo, la persona più gentile che abbia mai conosciuto, il miglior compagno di ufficio che si possa desiderare. Un grazie a Davide Piccioni per il suo travolgente entusiasmo e alla Mary per essere la migliore chef di Trieste (pari merito con Rici). Un grazie a Marti che ha sempre un sorriso da regalare, ad Anirudh per le "pizzette" e a Laria (senza la "i") per i sempre preziosi consigli e per essermi amica nonostante le divergenti opinioni politiche.

Un ringraziamento va a quel pazzo di Matteo Gallo, che riesce a mettermi di buon umore ogni volta che lo vedo, e che contro la mia volontà, mi ha fatto diventare padre di Tippy, insieme alle grandi menti di Vittorio e Rici. Mi sembra obbligatorio ringraziare anche Ghery e Isa, per essere degli incredibili genitori adottivi e per prendersi cura di lui in mia assenza. Grazie a Giada e Davide per i tanti momenti divertenti trascorsi insieme, attendo ancora di assaggiare le carotine fatte con la friggitrice ad aria (che anche Davide dovrà provare). Un ringraziamento speciale va poi a Claudia, per avermi fatto scoprire l'incredibile Hugo dell'Aldi e per tutte le volte che è venuta a trovarci a Trieste, superando le mille difficoltà logistiche.

Un sincero ringraziamento a Trotta, Fabio e Lorenzo per la loro autentica amicizia. Nonostante ciascuno di noi abbia la propria vita e la distanza che ci separa, ogni volta che ci rivediamo è come tornare tra i banchi di scuola. Ringrazio Vins, Samantha e Federica per la loro amicizia e per esserci sempre.

Un ringraziamento speciale va a Francesca, che, sebbene sia entrata nella mia vita da poco, ha già assunto un ruolo molto importante per me. Nonostante le mie innumerevoli insicurezze, mi sta accanto, sempre pronta ad aiutarmi, offrendomi parole di conforto e incoraggiandomi a migliorare, ad accettare le mie fragilità e a volermi bene. Sappi che, anche se la distanza ci separa, ti sento sempre vicina.

Ringrazio la mia seconda famiglia: Gianluca, Alessandra, Lorenzo e Sasà. Grazie per il vostro supporto costante e per farmi sentire sempre a casa ogni volta che sono con voi.

Un ringraziamento alla mia famiglia, in particolare ai miei genitori per il loro sostegno nelle mie scelte di vita e per avermi sempre insegnato l'importanza dello studio e della perseveranza. Dopo tanti anni, ricordo ancora lo sguardo di approvazione di mio padre quando, in "Nuovo Cinema Paradiso", Alfredo si rivolge a Salvatore pronunciando la frase: "Non tornare più, non ci pensare mai a noi, non ti voltare, non scrivere. Non ti fare fottere dalla nostalgia, dimenticaci tutti. Se non resisti e torni indietro, non venirmi a trovare, non ti faccio entrare a casa mia.". Queste parole risuonano sempre nella mia testa e mi rendono orgoglioso perché mi fanno ricordare da dove vengo.

Voglio esplicitamente ringraziare mia sorella (per evitare che si offenda nuovamente) per sopportarmi e per lasciarmi sedere sul divano nel mio posto preferito quando ritorno a casa.

Infine, un ringraziamento particolare va alle mie nonne, per il loro immenso amore.

Appendix A

Marshall Sign Rule Prior

For any bipartite lattice the exact signs of the ground state wave function of an Hamiltonian with nearest-neighbor interactions (e.g., the Heisenberg model) satisfy the so-called Marshall-sign rule [34]. Specifically, the sign of the wave function in the computational basis is given by $\mathrm{sign}[\Psi_0(\sigma)] = (-1)^{N_{\uparrow,A}(\sigma)}$, where $N_{\uparrow,A}(\sigma)$ is the number of up spins on the A sublattice. Motivated by this fact, it is common to consider variational states in which $(-1)^{N_{\uparrow,A}(\sigma)}$ is attached to the amplitudes of the variational states for studying frustrated systems (see Table 4.2), although the Marhsall-sign rule gives the exact signs of the ground state only in the unfrustrated limit. However, it still turns out to constitute a reasonable approximation for the sign structure of the exact wave function in a certain regime of frustration (see Fig. A.1). The accuracy of the Marshall-sign rule can be assessed by evaluating the following average on relatively small clusters, such as the ones that can be tackled by exact diagonalization:

$$\langle s_{\mathrm{MSR}} \rangle = \left| \sum_{\{\sigma\}} |\Psi_0(\sigma)|^2 \mathrm{sign}\left[\Psi_0(\sigma)\right] \mathcal{M}(\sigma) \right| \, , \tag{A.1}$$

where $\Psi_0(\sigma)$ is the exact ground-state amplitude and $\mathcal{M}(\sigma) = (-1)^{N_{\uparrow,A}(\sigma)}$ is the Marshall sign of the configuration σ. The absolute value is taken to overcome a possible global sign in the exact state. Whenever the Marshall-sign rule is exact, $\langle s_{\mathrm{MSR}} \rangle = 1$, otherwise $0 \leq \langle s_{\mathrm{MSR}} \rangle < 1$.

In panels (a) and (b) of Fig. A.1, we show the values of $\langle s_{\mathrm{MSR}} \rangle$ for the one-dimensional J_1-J_2 Heisenberg model [see Eq. (3.3)] for a cluster of $N = 20$ and $N = 30$ sites. The momentum of the ground state is either $k = 0$ or π: while for $J_2/J_1 \leq 0.5$ it does not

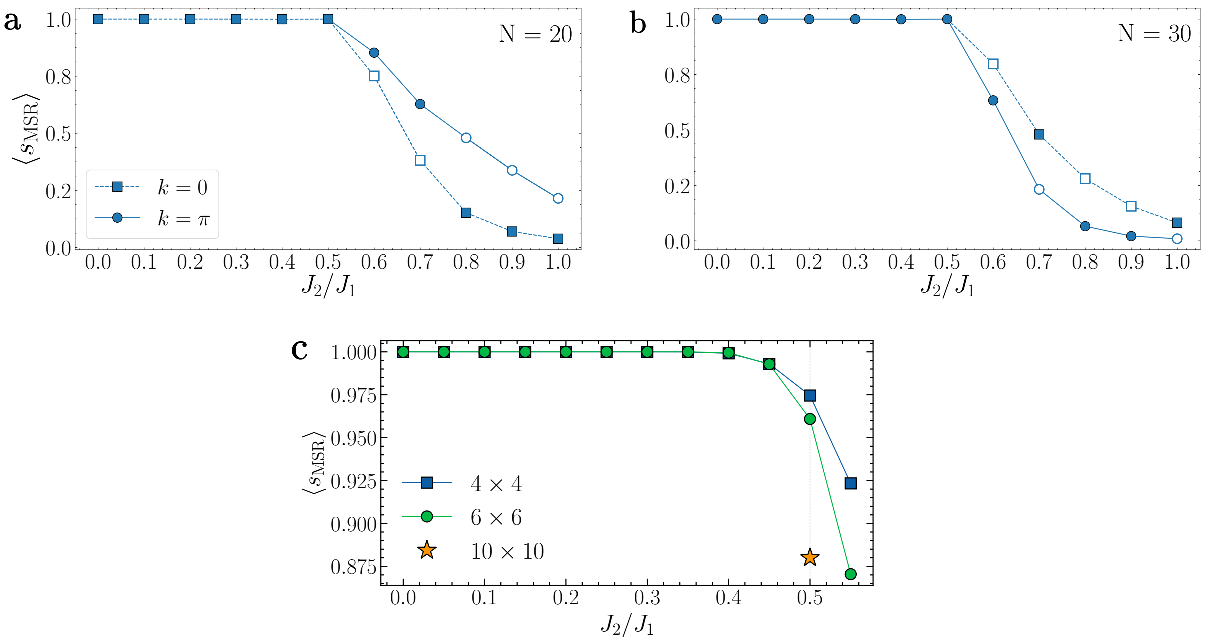

Figure A.1: **Panel a and b:** Average Marshall-sign for the one-dimensional chain as a function of J_2/J_1 for $N = 20$ [see panel (a)] and 30 [see panel (b)]. For $J_2/J_1 < 0.5$ the ground state has momentum $k = 0$ (for $N = 20$) and $k = \pi$ (for $N = 30$); for $J_2/J_1 > 0.5$, the momentum of the ground state is not fixed. For that reason, we report both states, the actual ground state is marked by a filled symbol. **Panel c:** Average Marshall-sign for the square lattice as a function of J_2/J_1 for 4×4 (blue squares) and 6×6 (green circles) and the extrapolated value at $J_2/J_1 = 0.5$ from Ref. [128] for the 10×10 (orange star) lattice.

depend on J_2/J_1 but only on the parity of $N/2$, for $J_2/J_1 > 0.5$ it changes with the frustrating ratio and N. Therefore, for this latter case, we compute $\langle s_{\text{MSR}} \rangle$ for both the lowest-energy wave functions with $k = 0$ and π. The remarkable outcome is that, even on a relatively large cluster, $\langle s_{\text{MSR}} \rangle$ is very close to 1 in the whole region $0 \leq J_2/J_1 \leq 0.5$ (it is exactly 1 for $J_2/J_1 = 0$ and 0.5), while it rapidly drops to zero for $J_2/J_1 > 0.5$. As an example, on $N = 30$ sites, $\langle s_{\text{MSR}} \rangle = 0.99994$ for $J_2/J_1 = 0.3$ and $\langle s_{\text{MSR}} \rangle = 0.08195$ for $J_2/J_1 = 1$ [38].

As shown in the panel (c) of Fig. A.1, a similar behavior is observed in the J_1-J_2 Heisenberg model on the square lattice [see Eq. (4.4)]. Around $J_2/J_1 = 0.5$, the average Marshall-sign, $\langle s_{\text{MSR}} \rangle$, begins to deviate significantly from 1.0 [195]. In Ref. [128], this value for a 10×10 lattice is extrapolated from smaller system sizes under the assumption of exponential decay with the number of spins, yielding an estimate of $\langle s_{\text{MSR}} \rangle \approx 0.88$.

Appendix B

Quantum Number Projection

Variational wave functions do not necessarily possess the symmetries of the physical model under investigation. In principle, the correct symmetries of the exact ground state can be potentially recovered by the variational state in the limit of a large number of parameters, since NQS states have the property of being universal approximators of arbitrary functions. However, in practice, the optimized variational wave functions do not fulfill exactly the symmetries of the Hamiltonian. A possible way to overcome this issue is applying a projection operator to enforce the desired symmetries with definite quantum numbers [196]. In general, for a symmetry group G and a variational state $|\Psi_\theta\rangle$, which does not exhibit the desired symmetry, we can enforce it by applying the projection operator:

$$\hat{\mathcal{P}}_G = \frac{1}{|G|} \sum_{g \in G} \chi_g^* \; \hat{g} \; , \tag{B.1}$$

where $\{\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{|G|-1}\}$ is the set of operators corresponding to the elements of the group, and $\{\chi_g\}$ represent the characters of the group. This operator projects the non-symmetric state onto the subspace that is invariant under the group action. The symmetry-enforced invariant state is defined as $|\tilde{\Psi}_\theta\rangle = \hat{\mathcal{P}}_G \, |\Psi_\theta\rangle$, whose corresponding amplitudes (neglecting the irrelevant scaling factor $1/|G|$) are given by :

$$\tilde{\Psi}_\theta(\sigma) = \sum_{g \in G} \chi_g^* \Psi_\theta(\sigma_g), \tag{B.2}$$

where $\langle \sigma | \hat{g} | \Psi_\theta \rangle = \Psi_\theta(\sigma_g)$, meaning that σ_g represents the transformed configuration under the group element $\hat{g}$. It is straightforward to demonstrate that $\tilde{\Psi}_\theta(\sigma_{g'}) = \chi_{g'}^* \tilde{\Psi}_\theta(\sigma)$ by exploiting the property of group characters, which satisfies $\chi_{g'g} = \chi_{g'} \chi_g$. This ensures that the projected state correctly transforms according to the the action of the group elements.

To be concrete, as an example we consider a non-translational invariant wave function, such as the RBM discussed in Sec. 3.3. The translational symmetry can be restored by defining:

$$\Psi_\theta(\sigma) = \sum_{R=0}^{N-1} e^{ikR} \Psi_\theta(\sigma_R) \ , \tag{B.3}$$

where, for simplicity, we restrict ourselves to the one-dimensional case. Here, $k = (2\pi/N)n$ with $n = 0, \dots, N-1$ is the crystal momentum, e^{ikR} represents the character of the translation group, with $R = 0, \dots, N-1$ the lattice vectors of the one-dimensional lattice. The term $\Psi_\theta(\sigma_R) = \langle \sigma | \hat{T}_R | \Psi_\theta \rangle$ is the wave function evaluated in the translated configuration, where $\{\hat{T}_R\}$ denotes the set of translation operators.

We emphasize that this projection procedure not only leads to a substantial improvement in the accuracy of the variational *Ansätze* [31, 36, 37, 106, 185], but also gives the possibility of approximating excited states, for example varying the momentum k in Eq. (B.3) [106].

However, from a practical standpoint, careful attention must be given to the implementation of the symmetrized wave function in Eq. (B.2) to avoid numerical instabilities. Typically, we parametrize the logarithm of the wave function, $\mathrm{Log}[\Psi_\theta(\sigma)]$ (see Sec. 1.4.3), and for symmetrized states, the idea is to evaluate the logarithm of the wave function $\mathrm{Log}[\tilde{\Psi}_\theta(\sigma)]$ by exploiting the known values of $\mathrm{Log}[\Psi_\theta(\sigma_g)]$, without explicitly exponentiating the wave function. To achieve this, we select a reference element of the group $\bar{g}$, and rewrite the sum in Eq. (B.2) as follows:

$$\mathrm{Log}[\tilde{\Psi}_\theta(\sigma)] = \mathrm{Log}[\Psi_\theta(\sigma_{\bar{g}})] + \mathrm{Log}\left(\chi^*_{\bar{g}} + \sum_{g \neq \bar{g}} \chi^*_g \ e^{\mathrm{Log}[\Psi_\theta(\sigma_g)] - \mathrm{Log}[\Psi_\theta(\sigma_{\bar{g}})]} \right) \ . \tag{B.4}$$

The standard procedure to optimize the symmetrized state involves first optimizing the non-symmetric state $\Psi_\theta(\sigma)$. If the state is sufficiently accurate, we expect it to approximately recover the symmetries of the Hamiltonian. In this case, the differences in the exponents $\mathrm{Log}[\Psi_\theta(\sigma_g)] - \mathrm{Log}[\Psi_\theta(\sigma_{\bar{g}})]$ will be $O(1)$, preventing numerical instabilities when evaluating their exponentials. After the initial optimization of the non-symmetric state, we can use the optimized parameters to further train the symmetrized state. This procedure can be iterated for the different symmetries of the Hamiltonian, as shown in inset of the left panel of Fig. 4.7.

Appendix C

Analytical Representation of Physical Ground States with Transformer Wave Functions

In this Appendix we examine an analytical solvable quantum many-body Hamiltonian, developing an exact mapping between its ground state and a single layer of two-headed Factored attention. Building upon this result, we extend our analysis to scenarios where the ground state lacks analytical solutions, providing insights into why attention mechanisms including queries and keys [as in Eq. (4.9) and Eq. (4.10)] should converge to positional only solutions when studying large systems.

As illustrative example of solvable quantum many-body Hamiltonian, we consider the Shastry-Sutherland model [164] (see Chapter 5 for a detailed description). In a finite range of the frustration ratio ($J/J' \lesssim 0.675$), the ground state of this model is represented as a product of singlets between next-nearest-neighbor spins arranged on a square lattice [164], refer to Fig. C.1 for a graphical representation. Here, we want to show that a single-layer ViT with Factored attention [see Eq. (4.11)] can represent exactly this ground state. Working on a $L \times L$ square lattice with periodic boundary conditions, we partition input spin configurations into $b \times b$ patches, with $b = 2$ (see Fig. C.1), which are then flattened to construct input sequences. Assuming an embedding dimension of $d = b^2 = 4$ and choosing the embedding matrix to be the identity, the i-th input vector is $\boldsymbol{x}_i = (\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \sigma_{i,4})^T$, where $i = 1, \dots, L^2/b^2$. Then, we apply the Multi-Head attention mechanism [18] with

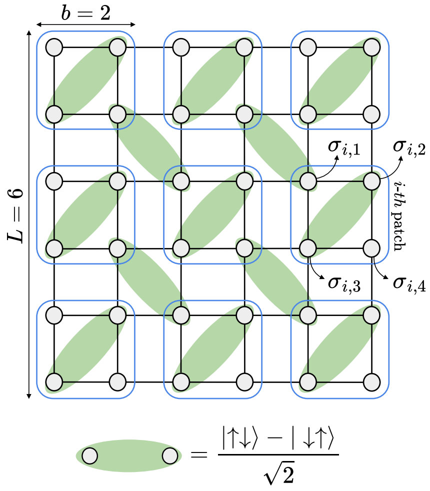

Figure C.1: Graphical representation of the ground state of the Shastry-Sutherland model in the dimer phase [164] on a 6×6 lattice (periodic boundary connections not shown for clarity). The green shaded regions denote singlet states between two next-nearest neighbors spins. The blue squares $b \times b$ indicate the patches used to construct the input set of vectors for the Transformer.

$h = 2$ heads. Considering the value matrices:

$$V^{(1)} = \begin{pmatrix} 0 & 0 & 0 & V^{(1)}_{11} \\ 0 & V^{(1)}_{22} & V^{(1)}_{23} & 0 \end{pmatrix} \qquad V^{(2)} = \begin{pmatrix} V^{(2)}_{14} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} , \tag{C.1}$$

the value vectors are computed as $\boldsymbol{v}_i^{(\mu)} = V^{(\mu)}\boldsymbol{x}_i \in \mathbb{R}^{d/h}$:

$$\boldsymbol{v}_i^{(1)} = \left(V^{(1)}_{11}\sigma_{i,4}, V^{(1)}_{22}\sigma_{i,2} + V^{(1)}_{23}\sigma_{i,3}\right)^T \qquad \boldsymbol{v}_i^{(2)} = \left(V^{(2)}_{14}\sigma_{i,1}, 0\right)^T . \tag{C.2}$$

Now, we assume the $L^2/b^2 \times L^2/b^2$ attention matrices to be $\alpha^{(1)}_{ij} = \delta_{i,j}$ and $\alpha^{(2)}_{ij} = \delta_{i,S(i)}$, where:

$$S(i) = \begin{cases} (i+1)\%(L^2/b^2) & \text{if } \ i\%(L/b) = 0, \\ (i + L/b)\%(L^2/b^2) + 1 & \text{otherwise,} \end{cases} \tag{C.3}$$

to take into account the periodic boundary conditions. Notably, the role of the two different heads is to encode the intra-patches correlations through the attention matrix $\alpha^{(1)}$ and the inter-patches correlations through $\alpha^{(2)}$. It is worth noting that, to reproduce the same attention maps with T5 [see Eq. (4.9)] or Decoupled [see Eq. (4.10)] attention mechanisms, we have to set $Q = K = 0$. The resulting attention vectors are:

$$\boldsymbol{A}_i^{(1)} = \left(V^{(1)}_{11}\sigma_{i,4}, V^{(1)}_{22}\sigma_{i,2} + V^{(1)}_{23}\sigma_{i,3}\right)^T \qquad \boldsymbol{A}_i^{(2)} = \left(V^{(2)}_{14}\sigma_{S(i),1}, 0\right)^T . \tag{C.4}$$

Following the Multi-Head mechanism [18], we concatenate the vectors $\boldsymbol{A}_i^{(\mu)}$ of the different heads and apply another matrix $W \in \mathbb{R}^{d\times d}$ to mix the different representations. Choosing W to be:

$$W = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \;, \tag{C.5}$$

we obtain:

$$\boldsymbol{A}_i = \left(V_{11}^{(1)}\sigma_{i,4} + V_{14}^{(2)}\sigma_{S(i),1}, V_{22}^{(1)}\sigma_{i,2} + V_{23}^{(1)}\sigma_{i,3}, 0, 0\right)^T \;. \tag{C.6}$$

At this point, in the standard architecture each attention vector is fed to a MLP; in our analytical computations, we substitute it with a generic nonlinearity $F(\boldsymbol{A}_i + c)$, where c is a constant bias. The output of this operation is the sequence of vectors:

$$\boldsymbol{y}_i = \left(F(V_{11}^{(1)}\sigma_{i,4} + V_{14}^{(2)}\sigma_{S(i),1} + c), F(V_{22}^{(1)}\sigma_{i,2} + V_{23}^{(1)}\sigma_{i,3} + c), 0, 0\right)^T \;. \tag{C.7}$$

The hidden representation is obtained by summing all the output vectors $\boldsymbol{z} = \sum_{i=1}^{L^2/b^2} \boldsymbol{y}_i$, where $\boldsymbol{z} \in \mathbb{R}^d$:

$$\boldsymbol{z} = \left(\sum_{i=1}^{L^2/b^2} F(V_{11}^{(1)}\sigma_{i,4} + V_{14}^{(2)}\sigma_{S(i),1} + c), \sum_{i=1}^{L^2/b^2} F(V_{22}^{(1)}\sigma_{i,2} + V_{23}^{(1)}\sigma_{i,3} + c), 0, 0\right)^T \;. \tag{C.8}$$

Replacing the fully-connected network that acts on $\boldsymbol{z}$ [27, 28, 39] with a simpler sum, we get the amplitude of the input spin configuration :

$$\mathrm{Log}[\Psi_\theta(\sigma)] = \sum_{i=1}^{L^2/b^2} \left[F(V_{11}^{(1)}\sigma_{i,4} + V_{14}^{(2)}\sigma_{S(i),1} + c) + F(V_{22}^{(1)}\sigma_{i,2} + V_{23}^{(1)}\sigma_{i,3} + c)\right] \;. \tag{C.9}$$

At the end, by choosing $F(\cdot) = \mathrm{logcos}(\cdot)$ and setting $V_{11}^{(1)} = V_{23}^{(1)} = \pi/4$, $V_{14}^{(2)} = V_{22}^{(1)} = 3\pi/4$ and $c = \pi/2$ we obtain an exact representation that fully complies with the ground state of the model, specifically a product of singlets arranged on a square lattice, as illustrated in Fig. C.1:

$$\Psi_0(\sigma) = \prod_{i=1}^{L^2/4} \cos\left(\frac{\pi}{2} + \pi(\sigma_{i,4} + 3\sigma_{S(i),1})\right) \cos\left(\frac{\pi}{2} + \pi(\sigma_{i,2} + 3\sigma_{i,3})\right) \;. \tag{C.10}$$

We want to emphasize that, to keep the analytical calculation manageable, we did not to include Layer Norm and skip connections. The mapping between the exact ground state of the Shastry-Sutherland model and the Transformer wave function highlights the role played by the different components of the architecture. In particular, this example reveals that the attention weights are used to describe the correlations in the ground state, and the attention weights connecting two patches containing uncorrelated spins should be zero to have an exact representation of the ground state.

In general, physical events that are sufficiently far apart (either in space or time) are essentially independent or uncorrelated. From a mathematical perspective, this fundamental concept is formalized through the *cluster property* [197, 198]:

$$\lim_{|i-j|\to+\infty} \langle \hat{A}_i \hat{A}_j \rangle = \langle \hat{A}_i \rangle \langle \hat{A}_j \rangle \ , \tag{C.11}$$

where $\hat{A}_i$ is a generic local operator. According to the cluster property, correlations must decay with distance and, in the thermodynamic limit, sites that are infinitely distant become uncorrelated. As shown in the previous mapping, the role of the attention weights is to connect correlated inputs. Therefore, for systems for which the property in Eq. (C.11) holds, we expect the attention weights connecting spins far apart in the system to be close to zero, regardless the specific values of the spins. Interestingly, to reproduce this long-distance behavior using standard T5 [Eq. (4.9)] or Decoupled [see Eq. (4.10)] attention mechanisms we have to require $Q = K = 0$. In other words, the standard attention mechanisms should converge to positional only solutions, thereby to the Factored version [see Eq. (4.11)].

Bibliography

[1] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, “Theory of superconductivity”, Phys. Rev. **108**, 1175–1204 (1957).

[2] R. B. Laughlin, “Anomalous quantum hall effect: an incompressible quantum fluid with fractionally charged excitations”, Phys. Rev. Lett. **50**, 1395–1398 (1983).

[3] P. Anderson, “The resonating valence bond state in la$_2$cuo$_4$ and superconductivity”, Science **235**, 1196–1198 (1987).

[4] L. Savary and L. Balents, “Quantum spin liquids: a review”, Reports on Progress in Physics **80**, 016502 (2016).

[5] S. White, “Density matrix formulation for quantum renormalization groups”, Phys. Rev. Lett. **69**, 2863–2866 (1992).

[6] F. Verstraete and J. Cirac, “Renormalization algorithms for quantum-many body systems in two and higher dimensions”, (2004).

[7] G. Carleo and M. Troyer, “Solving the quantum many-body problem with artificial neural networks”, Science **355**, 602–606 (2017).

[8] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, “Neural-network quantum states, string-bond states, and chiral topological states”, Phys. Rev. X **8**, 011006 (2018).

[9] H. Lange, A. V. de Walle, A. Abedinnia, and A. Bohrdt, *From architectures to applications: a review of neural quantum states*, 2024.

[10] K. Choo, T. Neupert, and G. Carleo, “Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states”, Phys. Rev. B **100**, 125124 (2019).

[11] X. Liang, W.-Y. Liu, P.-Z. Lin, G.-C. Guo, Y.-S. Zhang, and L. He, "Solving frustrated quantum many-particle models with convolutional neural networks", Phys. Rev. B **98**, 104426 (2018).

[12] A. Chen, K. Choo, N. Astrakhantsev, and T. Neupert, "Neural network evolution strategy for solving quantum sign structures", Phys. Rev. Research **4**, L022026 (2022).

[13] C. Roth and A. MacDonald, "Group convolutional neural networks improve quantum state accuracy", (2021).

[14] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions", Phys. Rev. Res. **2**, 023358 (2020).

[15] M. Hibat-Allah, R. G. Melko, and J. Carrasquilla, *Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy*, 2022.

[16] Y. Nomura, A. Darmawan, Y. Yamaji, and M. Imada, "Restricted boltzmann machine learning for solving strongly correlated quantum systems", Phys. Rev. B **96**, 205152 (2017).

[17] F. Ferrari, F. Becca, and J. Carrasquilla, "Neural gutzwiller-projected variational wave functions", Phys. Rev. B **100**, 125131 (2019).

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", Advances in neural information processing systems **30** (2017).

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805 (2019).

[20] A. Radford, K. Narasimhan, et al., *Improving language understanding by generative pre-training*, 2018.

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

[22] 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 **596**, 583–589 (2021).

[23] OpenAI, "Gpt-4 technical report", arXiv preprint arXiv:2303.08774 (2024).

[24] 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", arXiv preprint arXiv:2010.11929 (2021).

[25] R. G. Melko and J. Carrasquilla, "Language models for quantum simulation", Nature Computat. Sci. **4**, 11–18 (2024).

[26] K. Sprague and S. Czischek, "Variational monte carlo with large patched transformers", Communications Physics **7**, `10.1038/s42005-024-01584-y` (2024).

[27] 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", Communications Physics **7**, `10.1038/s42005-024-01732-4` (2024).

[28] 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*, 2024.

[29] 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).

[30] 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).

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

[32] I. von Glehn, J. S. Spencer, and D. Pfau, "A self-attention ansatz for ab-initio quantum chemistry", arXiv preprint arXiv:2211.13672 (2023).

[33] H. Lange, G. Bornet, G. Emperauger, C. Chen, T. Lahaye, S. Kienle, A. Browaeys, and A. Bohrdt, *Transformer neural networks and quantum simulators: a hybrid approach for simulating strongly correlated systems*, 2024.

[34] W. Marshall, "Antiferromagnetism", Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **232**, 48–68 (1955).

[35] S. Yan, D. A. Huse, and S. R. White, "Spin-liquid ground state of the ¡i¿s¡/i¿ = 1/2 kagome heisenberg antiferromagnet", Science **332**, 1173–1176 (2011).

[36] A. Chen and M. Heyl, “Empowering deep neural quantum states through efficient optimization”, Nature Physics **20**, 1476–1481 (2024).

[37] 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).

[38] 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).

[39] R. Rende, S. Goldt, F. Becca, and L. L. Viteritti, “Fine-tuning neural network quantum states”, arXiv preprint arXiv:2403.07795 (2024).

[40] A. J. Leggett, “Superfluidity”, Rev. Mod. Phys. **71**, S318–S323 (1999).

[41] H. L. Stormer, “Nobel lecture: the fractional quantum hall effect”, Rev. Mod. Phys. **71**, 875–889 (1999).

[42] J. G. Bednorz and K. A. Müller, “Possible high t_c superconductivity in the ba-la-cu-o system”, Zeitschrift für Physik B Condensed Matter **64**, 189–193 (1986).

[43] E. Dagotto, “Correlated electrons in high-temperature superconductors”, Rev. Mod. Phys. **66**, 763–840 (1994).

[44] W. Kohn and L. J. Sham, “Self-consistent equations including exchange and correlation effects”, Phys. Rev. **140**, A1133–A1138 (1965).

[45] R. O. Jones, “Density functional theory: its origins, rise to prominence, and future”, Rev. Mod. Phys. **87**, 897–923 (2015).

[46] J. Hubbard, “Electron correlations in narrow energy bands”, Proc. R. Soc. London, Ser. A **276**, 238–257 (1963).

[47] A. Zagoskin, *Quantum theory of many-body systems: techniques and applications*, 2nd (Springer Publishing Company, Incorporated, 2016).

[48] C. Broholm, R. J. Cava, S. A. Kivelson, D. G. Nocera, M. R. Norman, and T. Senthil, “Quantum spin liquids”, Science **367**, eaay0668 (2020).

[49] L. Balents, “Spin liquids in frustrated magnets”, Nature **464**, 199–208 (2010).

[50] L. Savary and L. Balents, “Quantum spin liquids: a review”, Rep. Prog. Phys. **80**, 016502 (2017).

[51] H. Xu, C.-M. Chung, M. Qin, U. Schollwöck, S. R. White, and S. Zhang, "Coexistence of superconductivity with partially filled stripes in the hubbard model", Science **384**, 10.1126/science.adh7691 (2024).

[52] F. Simkovic, R. Rossi, A. Georges, and M. Ferrero, "Origin and fate of the pseudogap in the doped hubbard model", Science **385**, eade9194 (2024).

[53] J. P. F. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K.-L. Chan, C.-M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jiménez-Hoyos, E. Kozik, X.-W. Liu, A. J. Millis, N. V. Prokof'ev, M. Qin, G. E. Scuseria, H. Shi, B. V. Svistunov, L. F. Tocchio, I. S. Tupitsyn, S. R. White, S. Zhang, B.-X. Zheng, Z. Zhu, and E. Gull (Simons Collaboration on the Many-Electron Problem), "Solutions of the two-dimensional hubbard model: benchmarks and results from a wide range of numerical algorithms", Phys. Rev. X **5**, 041041 (2015).

[54] A. W. Sandvik, A. Avella, and F. Mancini, "Computational studies of quantum spin systems", in Aip conference proceedings (2010).

[55] A. Wietek, S. Capponi, and A. M. Läuchli, "Quantum electrodynamics in $2+1$ dimensions as the organizing principle of a triangular lattice antiferromagnet", Phys. Rev. X **14**, 021010 (2024).

[56] A. Dawid, J. Arnold, B. Requena, A. Gresch, M. Płodzień, K. Donatella, K. A. Nicoli, P. Stornati, R. Koch, M. Büttner, R. Okuła, G. Muñoz-Gil, R. A. Vargas-Hernández, A. Cervera-Lierta, J. Carrasquilla, V. Dunjko, M. Gabrié, P. Huembeli, E. van Nieuwenburg, F. Vicentini, L. Wang, S. J. Wetzel, G. Carleo, E. Greplová, R. Krems, F. Marquardt, M. Tomza, M. Lewenstein, and A. Dauphin, *Modern applications of machine learning in quantum sciences*, 2023.

[57] O. Sharir, A. Shashua, and G. Carleo, "Neural tensor contractions and the expressive power of deep neural quantum states", Phys. Rev. B **106**, 205136 (2022).

[58] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states", Annals of Physics **326**, 96–192 (2011).

[59] R. Orús, "A practical introduction to tensor networks: matrix product states and projected entangled pair states", Annals of Physics **349**, 117–158 (2014).

[60] F. Verstraete, V. Murg, and J. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems", Advances in Physics **57**, 143–224 (2008).

[61] E. Stoudenmire and S. R. White, “Studying two-dimensional systems with the density matrix renormalization group”, Annual Review of Condensed Matter Physics **3**, 111–128 (2012).

[62] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, “Quantum entanglement in deep learning architectures”, Phys. Rev. Lett. **122**, 065301 (2019).

[63] F. Becca and S. Sorella, *Quantum monte carlo approaches for correlated systems* (Cambridge University Press, 2017).

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

[65] C. Robert and G. Casella, *Monte Carlo statistical methods* (Springer Verlag, 2004).

[66] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, “Adaptive monte carlo augmented with normalizing flows”, Proceedings of the National Academy of Sciences **119**, 10.1073/pnas.2109420119 (2022).

[67] L. Galliano, R. Rende, and D. Coslovich, “Policy-guided monte carlo on general state spaces: application to glass-forming mixtures”, The Journal of Chemical Physics **161**, 10.1063/5.0221221 (2024).

[68] S. Sorella, “Green function monte carlo with stochastic reconfiguration”, Phys. Rev. Lett. **80**, 4558–4561 (1998).

[69] S. Sorella, “Wave function optimization in the variational monte carlo method”, Phys. Rev. B **71**, 241103 (2005).

[70] D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, 2017.

[71] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2019.

[72] S. Amari and S. Douglas, “Why natural gradient?”, in Proceedings of the 1998 ieee international conference on acoustics, speech and signal processing, icassp ’98 (cat. no.98ch36181), Vol. 2 (1998), 1213–1216 vol.2.

[73] S. Amari, R. Karakida, and M. Oizumi, “Fisher information and natural gradient learning in random deep networks”, in Proceedings of the twenty-second international conference on artificial intelligence and statistics, Vol. 89, edited by K. Chaudhuri and M. Sugiyama, Proceedings of Machine Learning Research (16–18 Apr 2019), pp. 694–702.

[74] Y. Nomura, “Boltzmann machines and quantum many-body problems”, Journal of Physics: Condensed Matter **36**, 073001 (2023).

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

[76] C.-Y. Park and M. J. Kastoryano, “Geometry of learning neural quantum states”, Phys. Rev. Res. **2**, 023232 (2020).

[77] H. V. Henderson and S. R. Searle, “On deriving the inverse of a sum of matrices”, Siam Review **23**, 53–60 (1981).

[78] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*, Nov. 2012.

[79] 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, Vol. 162, Proceedings of Machine Learning Research (17–23 Jul 2022), pp. 17018–17044.

[80] 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 Physics Codebases, `10.21468/scipostphyscodeb.7` (2022).

[81] 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).

[82] J. Bradbury, R. Frostig, P. Hawkins, M. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, *JAX: composable transformations of Python+NumPy programs*, version 0.3.13, 2018.

[83] T. Neupert, M. Fischer, E. Greplova, K. Choo, and M. Denner, *Introduction to machine learning for the sciences*, 2021.

[84] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer normalization*, 2016.

[85] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015.

[86] A. Barron, “Universal approximation bounds for superpositions of a sigmoidal function”, IEEE Transactions on Information Theory **39**, 930–945 (1993).

[87] G. Cybenko, “Approximation by superpositions of a sigmoidal function”, Mathematics of Control, Signals, and Systems (MCSS) **2**, 303–314 (1989).

[88] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, *The expressive power of neural networks: a view from the width*, 2017.

[89] 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", Physics Reports **810**, 1–124 (2019).

[90] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, *On the number of linear regions of deep neural networks*, 2014.

[91] R. Eldan and O. Shamir, *The power of depth for feedforward neural networks*, 2016.

[92] M. Telgarsky, *Benefits of depth in neural networks*, 2016.

[93] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos", in Advances in neural information processing systems, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (2016).

[94] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, *On the expressive power of deep neural networks*, 2017.

[95] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in Advances in neural information processing systems, Vol. 25, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (2012).

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in 2016 ieee conference on computer vision and pattern recognition (cvpr) (2016), pp. 770–778.

[97] W.-Y. Liu, X.-T. Zhang, Z. Wang, S.-S. Gong, W.-Q. Chen, and Z.-C. Gu, *Deconfined quantum criticality with emergent symmetry in the extended shastry-sutherland model*, 2023.

[98] 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 (2013).

[99] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, "Deep autoregressive models for the efficient variational simulation of many-body quantum systems", Phys. Rev. Lett. **124**, 020503 (2020).

[100] J. Robledo Moreno, G. Carleo, A. Georges, and J. Stokes, "Fermionic wave functions from neural-network constrained hidden states", Proceedings of the National Academy of Sciences **119**, `10.1073/pnas.2122059119` (2022).

[101] 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 preprint arXiv:2305.08831 (2023).

[102] 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).

[103] J. Nys, G. Pescia, and G. Carleo, "Ab-initio variational wave functions for the time-dependent many-electron schrödinger equation", arXiv preprint arXiv:2403.07447 (2024).

[104] Z. Denis and G. Carleo, "Accurate neural quantum states for interacting lattice bosons", arXiv preprint arXiv:2404.07869 (2024).

[105] K. Choo, G. Carleo, N. Regnault, and T. Neupert, "Symmetries and many-body excitations with neural-network quantum states", Phys. Rev. Lett. **121**, 167204 (2018).

[106] Y. Nomura, "Helping restricted boltzmann machines with quantum-state representation by restoring symmetry", Journal of Physics: Condensed Matter **33**, 174003 (2021).

[107] D. Pfau, S. Axelrod, H. Sutterud, I. von Glehn, and J. S. Spencer, "Accurate computation of quantum excited states with neural networks", Science **385**, eadn0137 (2024).

[108] 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).

[109] S. White and I. Affleck, "Dimerization and incommensurate spiral spin correlations in the zigzag spin chain: analogies to the kondo lattice", Phys. Rev. B **54**, 9862–9869 (1996).

[110] S. Eggert, "Numerical evidence for multiplicative logarithmic corrections from marginal operators", Phys. Rev. B **54**, R9612–R9615 (1996).

[111] N. Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks", Neural computation **20**, 1631–49 (2008).

[112] G. Montufar and N. Ay, "Refinements of universal approximation results for deep belief networks and restricted boltzmann machines", Neural computation **23**, 1306–19 (2011).

[113] A. Chen, K. Choo, N. Astrakhantsev, and T. Neupert, "Neural network evolution strategy for solving quantum sign structures", (2021).

[114] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography", Nature Physics **14**, 447–450 (2018).

[115] Y. Nomura, "Machine learning quantum states extensions to fermion-boson coupled systems and excited-state calculations", Journal of the Physical Society of Japan **89**, 054706 (2020).

[116] F. Ferrari, A. Parola, S. Sorella, and F. Becca, "Dynamical structure factor of the $J_1 - J_2$ heisenberg model in one dimension: the variational monte carlo approach", Phys. Rev. B **97**, 235103 (2018).

[117] C. M. Bishop and H. Bishop, *Deep learning - foundations and concepts*, edited by S. Cham, 1st ed. (2023).

[118] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations", arXiv preprint arXiv:1803.02155 (2018).

[119] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", arXiv preprint arXiv:1910.10683 (2023).

[120] H. Cui, F. Behrens, F. Krzakala, and L. Zdeborová, *A phase transition between positional and semantic learning in a solvable model of dot-product attention*, 2024.

[121] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu, "On layer normalization in the transformer architecture", arXiv preprint arXiv:2002.04745 (2020).

[122] 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*, June 2021.

[123] N. Bhattacharya, N. Thomas, R. Rao, J. Dauparas, P. K. Koo, D. Baker, Y. S. Song, and S. Ovchinnikov, "Interpreting potts and transformer protein models through the lens of simplified attention", in *Biocomputing 2022* (), pp. 34–45.

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

[125] S. Jelassi, M. E. Sander, and Y. Li, "Vision transformers provably learn spatial structure", in Advances in neural information processing systems, edited by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho (2022).

[126] P. Pippan, S. White, and H. Evertz, "Efficient matrix-product state method for periodic boundary conditions", Phys. Rev. B **81**, 081103 (2010).

[127] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, "Generalization properties of neural network approximations to frustrated magnet ground states", Nature Communications **11**, 10.1038/s41467-020-15402-w (2020).

[128] A. Szabó and C. Castelnovo, "Neural network wave functions and the sign problem", Phys. Rev. Res. **2**, 033075 (2020).

[129] C.-Y. Park and M. Kastoryano, *Expressive power of complex-valued restricted boltzmann machines for solving non-stoquastic hamiltonians*, Aug. 2021.

[130] M. Bukov, M. Schmitt, and M. Dupont, "Learning the ground state of a non-stoquastic quantum hamiltonian in a rugged neural network landscape", SciPost Phys. **10**, 147 (2021).

[131] 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).

[132] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, http://www.deeplearningbook.org (MIT Press, Cambridge, MA, USA, 2016).

[133] Y. Bengio, A. Courville, and P. Vincent, *Representation learning: a review and new perspectives*, 2014.

[134] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in Advances in neural information processing systems, Vol. 25, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (2012).

[135] M. Li, J. Chen, Q. Xiao, F. Wang, Q. Jiang, X. Zhao, R. Lin, H. An, X. Liang, and L. He, "Bridging the gap between deep learning and frustrated quantum spin system for extreme-scale simulations on new generation of sunway supercomputer", IEEE Transactions on Parallel amp; Distributed Systems **33**, 2846–2859 (2022).

[136] X. Liang, M. Li, Q. Xiao, H. An, L. He, X. Zhao, J. Chen, C. Yang, F. Wang, H. Qian, L. Shen, D. Jia, Y. Gu, X. Liu, and Z. Wei, 2^{1296} *exponentially complex quantum many-body simulation via scalable deep learning method*, 2022.

[137] 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).

[138] A. F. Agarap, *Deep learning using rectified linear units (relu)*, 2019.

[139] 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*, 2021.

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

[141] R. Rende and L. L. Viteritti, *Are queries and keys always relevant? a case study on transformer wave functions*, 2024.

[142] Y. Nomura, "Helping restricted boltzmann machines with quantum-state representation by restoring symmetry", Journal of Physics: Condensed Matter **33**, 174003 (2021).

[143] M. Reh, M. Schmitt, and M. Gärttner, "Optimizing design choices for neural quantum states", Phys. Rev. B **107**, 195115 (2023).

[144] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, A. Mohamed, M. Philipose, M. Richardson, and R. Caruana, "Do deep convolutional nets really need to be deep and convolutional?", in International conference on learning representations (2017).

[145] S. d'Ascoli, L. Sagun, G. Biroli, and J. Bruna, "Finding the needle in the haystack with convolutions: on the benefits of architectural bias", in Advances in neural information processing systems, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (2019).

[146] A. Ingrosso and S. Goldt, "Data-driven emergence of convolutional structure in neural networks", Proceedings of the National Academy of Sciences **119**, e2201854119 (2022).

[147] 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**, 11446–11456 (1998).

[148] A. W. Sandvik, "Finite-size scaling of the ground-state parameters of the two-dimensional heisenberg model", Phys. Rev. B **56**, 11678–11690 (1997).

[149] 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).

[150] X. Liang, M. Li, Q. Xiao, H. An, L. He, X. Zhao, J. Chen, C. Yang, F. Wang, H. Qian, L. Shen, D. Jia, Y. Gu, X. Liu, and Z. Wei, 2^{1296} *exponentially complex quantum many-body simulation via scalable deep learning method*, 2022.

[151] 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).

[152] M. Li, J. Chen, Q. Xiao, F. Wang, Q. Jiang, X. Zhao, R. Lin, H. An, X. Liang, and L. He, "Bridging the gap between deep learning and frustrated quantum spin system for extreme-scale simulations on new generation of sunway supercomputer", IEEE Transactions on Parallel amp; Distributed Systems **33**, 2846–2859 (2022).

[153] H. Chen, D. Hendry, P. Weinberg, and A. Feiguin, "Systematic improvement of neural network quantum states using lanczos", in Advances in neural information processing systems, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (2022), pp. 7490–7503.

[154] J.-Q. Wang, R.-Q. He, and Z.-Y. Lu, *Variational optimization of the amplitude of neural-network quantum many-body ground states*, 2023.

[155] X. Liang, S.-J. Dong, and L. He, "Hybrid convolutional neural network and projected entangled pair states wave functions for quantum many-particle states", Phys. Rev. B **103**, 035138 (2021).

[156] E. Ledinauskas and E. Anisimovas, *Scalable imaginary time evolution with neural network quantum states*, 2023.

[157] Z. Dai, H. Liu, Q. V. Le, and M. Tan, *Coatnet: marrying convolution and attention for all data sizes*, 2021.

[158] U. Wennberg and G. E. Henter, "The case for translation-invariant self-attention in transformer-based language models", arXiv preprint arXiv:2106.01950 (2021).

[159] G. Ke, D. He, and T.-Y. Liu, "Rethinking positional encoding in language pre-training", in International conference on learning representations (2021).

[160] M. Reh, M. Schmitt, and M. Gärttner, "Optimizing design choices for neural quantum states", Phys. Rev. B **107**, 195115 (2023).

[161] L. McInnes, J. Healy, and J. Melville, *Umap: uniform manifold approximation and projection for dimension reduction*, 2020.

[162] Y. Tang, J. Liu, J. Zhang, and P. Zhang, “Learning nonequilibrium statistical mechanics and dynamical phase transitions”, Nature Communications **15**, 1117 (2024).

[163] C. Lacroix, P. Mendels, and F. Mila, *Introduction to frustrated magnetism: materials, experiments, theory* (Jan. 2011).

[164] B. Shastry and B. Sutherland, “Exact ground state of a quantum mechanical antiferromagnet”, Physica B+C **108**, 1069–1070 (1981).

[165] H. Kageyama, K. Yoshimura, R. Stern, N. V. Mushnikov, K. Onizuka, M. Kato, K. Kosuge, C. P. Slichter, T. Goto, and Y. Ueda, “Exact dimer ground state and quantized magnetization plateaus in the two-dimensional spin system $SrCu_2(BO_3)_2$”, Phys. Rev. Lett. **82**, 3168–3171 (1999).

[166] S. Miyahara and K. Ueda, “Exact dimer ground state of the two dimensional heisenberg spin system $SrCu_2(BO_3)_2$”, Phys. Rev. Lett. **82**, 3701–3704 (1999).

[167] K. Onizuka, H. Kageyama, Y. Narumi, K. Kindo, Y. Ueda, and T. Goto, “1/3 magnetization plateau in srcu 2(bo 3) 2 - stripe order of excited triplets -”, Journal of the Physical Society of Japan **69**, 1016–1018 (2000).

[168] K. Kodama, M. Takigawa, M. Horvatić, C. Berthier, H. Kageyama, Y. Ueda, S. Miyahara, F. Becca, and F. Mila, “Magnetic superstructure in the two-dimensional quantum antiferromagnet srcu¡sub¿2¡/sub¿(bo¡sub¿3¡/sub¿)¡sub¿2¡/sub¿”, Science **298**, 395–399 (2002).

[169] P. Corboz and F. Mila, “Crystals of bound states in the magnetization plateaus of the shastry-sutherland model”, Phys. Rev. Lett. **112**, 147203 (2014).

[170] M. Albrecht and F. Mila, “First-order transition between magnetic order and valence bond order in a 2d frustrated heisenberg model”, Europhysics Letters **34**, 145 (1996).

[171] Z. Weihong, C. J. Hamer, and J. Oitmaa, “Series expansions for a heisenberg antiferromagnetic model for $SrCu_2(BO_3)_2$”, Phys. Rev. B **60**, 6608–6616 (1999).

[172] A. Koga and N. Kawakami, “Quantum phase transitions in the shastry-sutherland model for $SrCu_2(BO_3)_2$”, Phys. Rev. Lett. **84**, 4461–4464 (2000).

[173] C. H. Chung, J. B. Marston, and S. Sachdev, “Quantum phases of the shastry-sutherland antiferromagnet: application to $SrCu_2(BO_3)_2$”, Phys. Rev. B **64**, 134407 (2001).

[174] A. Läuchli, S. Wessel, and M. Sigrist, “Phase diagram of the quadrumerized shastry-sutherland model”, Phys. Rev. B **66**, 014401 (2002).

[175] P. Corboz and F. Mila, “Tensor network study of the shastry-sutherland model in zero magnetic field”, Phys. Rev. B **87**, 115144 (2013).

[176] T. Waki, K. Arai, M. Takigawa, Y. Saiga, Y. Uwatoko, H. Kageyama, and Y. Ueda, “A novel ordered phase in srcu2(bo3)2 under high pressure”, Journal of the Physical Society of Japan **76**, 073710 (2007).

[177] M. E. Zayed, C. Rüegg, J. Larrea J., A. M. Läuchli, C. Panagopoulos, S. S. Saxena, M. Ellerby, D. F. McMorrow, T. Strässle, S. Klotz, G. Hamel, R. A. Sadykov, V. Pomjakushin, M. Boehm, M. Jiménez–Ruiz, A. Schneidewind, E. Pomjakushina, M. Stingaciu, K. Conder, and H. M. Rønnow, “4-spin plaquette singlet state in the shastry–sutherland compound $\mathrm{srcu_2(bo_3)_2}$”, Nature Physics **13**, 962–966 (2017).

[178] J. Guo, G. Sun, B. Zhao, L. Wang, W. Hong, V. A. Sidorov, N. Ma, Q. Wu, S. Li, Z. Y. Meng, A. W. Sandvik, and L. Sun, “Quantum phases of $\mathrm{SrCu_2(BO_3)_2}$ from high-pressure thermodynamics”, Phys. Rev. Lett. **124**, 206602 (2020).

[179] J. Y. Lee, Y.-Z. You, S. Sachdev, and A. Vishwanath, “Signatures of a deconfined phase transition on the shastry-sutherland lattice: applications to quantum critical $\mathrm{SrCu_2(BO_3)_2}$”, Phys. Rev. X **9**, 041037 (2019).

[180] J. Yang, A. W. Sandvik, and L. Wang, “Quantum criticality and spin liquid phase in the shastry-sutherland model”, Phys. Rev. B **105**, L060409 (2022).

[181] L. Wang, Y. Zhang, and A. W. Sandvik, “Quantum spin liquid phase in the shastry-sutherland model detected by an improved level spectroscopic method”, Chinese Physics Letters **39**, 077502 (2022).

[182] A. Kele ş and E. Zhao, “Rise and fall of plaquette order in the shastry-sutherland magnet revealed by pseudofermion functional renormalization group”, Phys. Rev. B **105**, L041115 (2022).

[183] M. Reh, M. Schmitt, and M. Gärttner, “Optimizing design choices for neural quantum states”, Phys. Rev. B **107**, 195115 (2023).

[184] 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 (2013).

[185] T. Vieijra, C. Casert, J. Nys, W. De Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete, “Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries”, Phys. Rev. Lett. **124**, 097201 (2020).

[186]T. Vieijra and J. Nys, “Many-body quantum states with exact conservation of non-abelian and lattice symmetries through variational monte carlo”, Phys. Rev. B **104**, 045123 (2021).

[187]G. Crognaletti, G. D. Bartolomeo, M. Vischi, and L. L. Viteritti, “Equivariant variational quantum eigensolver to detect phase transitions through energy level crossings”, arXiv preprint arXiv:2403.07100 (2024).

[188]D. Wu, R. Rossi, F. Vicentini, N. Astrakhantsev, F. Becca, X. Cao, J. Carrasquilla, F. Ferrari, A. Georges, M. Hibat-Allah, M. Imada, A. M. Läuchli, G. Mazzola, A. Mezzacapo, A. Millis, J. R. Moreno, T. Neupert, Y. Nomura, J. Nys, O. Parcollet, R. Pohle, I. Romero, M. Schmid, J. M. Silvester, S. Sorella, L. F. Tocchio, L. Wang, S. R. White, A. Wietek, Q. Yang, Y. Yang, S. Zhang, and G. Carleo, “Variational benchmarks for quantum many-body problems”, Science **386**, 296–301 (2024).

[189]J. Robledo Moreno, G. Carleo, A. Georges, and J. Stokes, “Fermionic wave functions from neural-network constrained hidden states”, Proceedings of the National Academy of Sciences **119**, `10.1073/pnas.2122059119` (2022).

[190]J. P. F. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K.-L. Chan, C.-M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jiménez-Hoyos, E. Kozik, X.-W. Liu, A. J. Millis, N. V. Prokof’ev, M. Qin, G. E. Scuseria, H. Shi, B. V. Svistunov, L. F. Tocchio, I. S. Tupitsyn, S. R. White, S. Zhang, B.-X. Zheng, Z. Zhu, and E. Gull (Simons Collaboration on the Many-Electron Problem), “Solutions of the two-dimensional hubbard model: benchmarks and results from a wide range of numerical algorithms”, Phys. Rev. X **5**, 041041 (2015).

[191]L. F. Tocchio, F. Becca, A. Parola, and S. Sorella, “Role of backflow correlations for the nonmagnetic phase of the t–t' hubbard model”, Phys. Rev. B **78**, 041101 (2008).

[192]A. Sinibaldi, C. Giuliani, G. Carleo, and F. Vicentini, “Unbiasing time-dependent variational monte carlo by projected quantum evolution”, Quantum **7**, 1131 (2023).

[193]W. Zhang, B. Xing, X. Xu, and D. Poletti, *Paths towards time evolution with larger neural-network quantum states*, 2024.

[194]L. Gravina, V. Savona, and F. Vicentini, *Neural projected quantum dynamics: a systematic study*, 2024.

[195]L. Capriotti, F. Becca, A. Parola, and S. Sorella, “Resonating valence bond wave functions for strongly frustrated spin systems”, Phys. Rev. Lett. **87**, 097201 (2001).

[196] T. Mizusaki and M. Imada, “Quantum-number projection in the path-integral renormalization group method”, Phys. Rev. B **69**, 125110 (2004).

[197] E. H. Wichmann and J. H. Crichton, “Cluster decomposition properties of the S matrix”, Phys. Rev. **132**, 2788–2799 (1963).

[198] S. Weinberg, “What is quantum field theory, and what did we think it was?”, in *Conceptual foundations of quantum field theory*, edited by T. Y. Cao (Cambridge University Press, 1999), pp. 241–251.