# Neural Network Quantum States

Nico Fransaert

*2020-2021*

Ghent University

GHENT UNIVERSITY

Department of Physics and Astronomy
Research Unit: Theoretical Nuclear and Statistical Physics

# Neural Network Quantum States

Nico Fransaert

*1. Promotor* Prof. Dr. Jan Ryckebusch
Department of Physics and Astronomy
Ghent University

*2. Supervisor* Dr. Jannes Nys
Department of Physics and Astronomy
Ghent University

*3. Supervisor* Tom Vieijra
Department of Physics and Astronomy
Ghent University

FACULTY OF SCIENCES

2020-2021

**Nico Fransaert**
*Neural Network Quantum States*
Machine Learning in Quantum Many-Body Physics, 2020-2021
Reviewers: Prof. Dr. Jan Ryckebusch and Prof. Dr. Jutho Haegeman
Supervisors: Prof. Dr. Jan Ryckebusch, Dr. Jannes Nys and Tom Vieijra

**Ghent University**
*Theoretical Nuclear and Statistical Physics*
Department of Physics and Astronomy
Proeftuinstraat 86, building N3
B-9000 Ghent

# Abstract

> *The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.*
>
> **— Paul M. Dirac [1]**

The physical laws that describe the foundations of Nature are considered largely understood. In principle, most observed phenomena are accounted for, meaning we can reap the benefits from the vast body of knowledge we have to our disposal. However, the mathematical formulations of physics, which usually appear as simple and compact equations, are rarely exactly solvable. Furthermore, the equations do not explicitly, or rather, straightforwardly, provide a picture of reality that matches observed phenomena. In practice, we circumvent these issues by i) composing approximate solutions to the underlying laws of Nature and ii) describing observations phenomenologically. Combining both strategies allows us to predict and understand Nature's behaviour in a practical manner.

In chapter 1, we provide an introduction to many-body physics. This framework describes complex systems of many interacting degrees of freedom. The connection between macroscopic properties of a system and its individual constituents are formally studied in classical statistical physics, which is the first topic of discussion. Next, we summarize the essentials of quantum many-body physics; the study of quantum systems at the limit of zero temperature. Following the ideas and

procedures of classical statistical physics, the formalism is extended to include non-zero temperature quantum systems; this is the study of quantum statistical physics. To finalize the chapter, symmetries and their usefulness in analyzing many-body quantum systems are discussed.

With the rapid increase of computational resources came the powerful technology of machine learning. In chapter 2, we introduce the methodology of machine learning, summarize its most prominent algorithms, and discuss the typical problems for which we use it. In chapter 3, we provide an overview of how to approximate the quantum many-body wave function by use of a restricted Boltzmann machine or recurrent neural network. It is shown how to implement SU(2) symmetry utilizing a basis of coupled spins. In chapter 4, we continue on the pioneering work in this field of research, and take advantage of the capabilities of machine learning to solve some selected problems of quantum many-body physics. In particular, we study the antiferromagnetic Heisenberg model and the $J_1 - J_2$ model in one and two dimensions using the previously mentioned networks. Among other things, it is shown that those networks represent ground states better when they satisfy SU(2) symmetry. The limits and the scaling of the method is investigated, computational tractability is considered, and comparisons with recent literature are provided.

# Nederlandse samenvatting

De natuurkundige wetten die de fundamenten van de natuur beschrijven worden als grotendeels begrepen beschouwd. In principe kunnen we de meeste waarnemingen verklaren, wat betekent dat we de vruchten kunnen plukken van de enorme hoeveelheid kennis die we tot onze beschikking hebben. De wiskundige formuleringen van de natuurkunde, die meestal verschijnen als eenvoudige en compacte vergelijkingen, zijn echter zelden exact oplosbaar. Bovendien geven de vergelijkingen niet expliciet, of beter gezegd, rechtstreeks een beeld van de werkelijkheid dat overeenkomt met waargenomen verschijnselen. In de praktijk omzeilen we deze problemen door i) benaderingen te bedenken voor de onderliggende natuurwetten en ii) waarnemingen fenomenologisch te beschrijven. Door beide strategieën te combineren kunnen we het gedrag van de natuur op een praktische manier voorspellen en begrijpen.

In hoofdstuk 1 geven we een inleiding tot de veeldeeltjesfysica. Dit onderzoeksgebied bestudeert complexe systemen die bestaan uit veel interagerende vrijheidsgraden. Het verband tussen macroscopische eigenschappen van een systeem en zijn individuele bestanddelen wordt formeel bestudeerd in de klassieke statistische fysica, wat het eerste onderwerp van discussie is. Vervolgens vatten we de essentie van de kwantumveeldeeltjestheorie samen; de studie van kwantumsystemen met een temperatuur op het absolute nulpunt. In navolging van de ideeën en procedures van de klassieke statistische fysica wordt het formalisme uitgebreid tot kwantumsystemen die een temperatuur verschillend van nul hebben; dit is de studie van kwantumstatistische fysica. We ronden het hoofdstuk af met een kijk op symmetrieën en bespreken hun bruikbaarheid bij het analyseren van kwantumveeldeeltjessystemen.

Ten gevolge van de snelle toename van computationele middelen verscheen een krachtige technologie die bekend staat als machinaal leren. In hoofdstuk 2 introduceren we de methodologie van machinaal leren, vatten we de meest prominente algoritmen samen, en bespreken we de typische problemen waarvoor we deze technologie gebruiken. In hoofdstuk 3 geven we een overzicht van hoe de kwantum golffunctie kan worden benaderd met behulp van een *restricted Boltzmann machine* of een *recurrent neural network*. Er wordt getoond hoe SU(2)-symmetrie kan worden geïmplementeerd door gebruik te maken van een basis van gekoppelde spins. In hoofdstuk 4 treden we in de voetsporen van de pioniers in dit domein, en maken we gebruik van de mogelijkheden van machinaal leren om een aantal geselecteerde problemen van de kwantumveeldeeltjesfysica op te lossen. Meer specifiek bestuderen we het antiferromagnetische Heisenberg model en het $J_1 - J_2$ model in één en twee dimensies met behulp van de eerder genoemde netwerken. Er wordt onder andere aangetoond dat die netwerken grondtoestanden beter weergeven als ze voldoen aan

de SU(2)-symmetrie. De limieten en de uitbreidbaarheid van de methode worden onderzocht, computationele geschiktheid wordt overwogen, en de resultaten worden vergeleken met recente literatuur.

# Acknowledgement

I want to thank Prof. Dr. Jan Ryckebusch, Dr. Jannes Nys, and Tom Vieijra for giving me the opportunity to perform this interesting study. Even in these strange times of the coronavirus you found ways to guide me.

I want to thank my parents, Claudia and Dirk, for always being there for me. I am eternally grateful for your encouragement and for enabling me to study physics.
I want to thank my sister, Nora, for supporting me.

I want to thank my significant other, Jessica, for her patience and love.
If my life were a book, then you would be its beautiful colours.

I want to thank my neighbours and closest friends, Alessandro and Talha, who after $15+$ years are more like brothers to me.

I want to thank Siebe and Yannick, with whom I have spend most of my time with the last couple of years.

And to basically everyone,
thank you for listening to my endless streams of physics talk!

# Contents

# 1 Introduction to many-body physics

The famous phrase "The whole is more than the sum of its parts" suggests that the interaction between many degrees of freedom gives rise to interesting emergent phenomena. An isolated molecule of water can not undergo a phase transition, a single bird can not fly in formation, and a single human cell can not feel pain. However, collective behaviour emerges when bodies (particles, birds, cells...) are brought into contact with each other.

In order to understand this process, one searches for the connection between the individual pieces and the collective. We refer to these pieces as the degrees of freedom, since they are the dynamical components that can move, vibrate, etc. In general, two approaches can be employed: the top-down method derives the underlying structure by decomposing the collective, with the goal of obtaining a tractable description. The bottom-up approach instead starts with the individual degrees of freedom and tries to capture the emergent behaviour. In any case, some rule or interaction is fundamental to the transition between the scale of the isolated degrees of freedom and that of the collective. The complexity of these interactions make many-body problems particularly hard to solve.

Many-body physics is the field of study that deals with physical problems involving a large amount of interacting degrees of freedom. The meaning of large can range between the number of atoms in macroscopic materials ($\sim 10^{23}$), and the number of electrons in atoms. Many-body physics is not limited to a particular time or spatial scale: it is a general framework for understanding the collective behaviour of many interacting constituents. While the underlying physical laws of the constituents are relatively simple, it is often challenging to understand the interplay between them. Sometimes, entirely new physics is discovered when a large amount of simple constituents interact. This leads to emergent phenomena observed in Nature. What makes emergence interesting is that it is not encoded in the physical laws — the emergent phenomena may come as a surprise.

## 1.1 Classical statistical physics

The aim of classical statistical physics is to understand and predict the behaviour of the collective by combining the procedures of statistics with the physical laws that govern the constituents. Many of its ideas stem from thermodynamics: the notions of work, temperature, energy, and entropy. It goes further than thermodynamics in that it relates macroscopic observations to the properties of the microscopic constituents. For example, in classical statistical mechanics, temperature is the quantitative measure of how energy is shared among particles. Statistics is used in order to realize this correspondence. Instead of keeping track of the positions and velocities of all the particles, one relies on the laws of probability to estimate the averages and the fluctuations of observables. Thermodynamic quantities are thus connected to microscopic behaviour, i.e., knowledge of microscopic parameters allows the prediction of macroscopic observables.

### 1.1.1 Introduction to the formalism

**Microstates, macrostates, and the coin flipping game** Consider a coin flipping game that involves three coins. The player wins the game if the flips end up in either all heads or all tails. The player loses in any of the other cases. The outcomes of the coin flipping can be written as a sequence $(f_1, f_2, f_3)$ of coin flips $f$, where $f_i = +1$ ($f_i = -1$) denotes heads (tails) on the $i$-th flip. Note that there are a total of $2^3 = 8$ possible outcomes, two of which lead to the player winning, namely $(+1, +1, +1)$ and $(-1, -1, -1)$. The other $6$ outcomes lead to the player losing the game. In the statistical physics context, the specific sequences $(f_1, f_2, f_3)$ correspond to so-called microstates. The two states of the player either winning or losing correspond to the macrostates. To see if playing the game is beneficial, one is only interested in the number of microstates associated to each of the macrostates. One does not really care about the specific sequences. For example, if losing costs $1$ and winning gives $2$, the expected profit is $((-1 \times 6) + (2 \times 2))/8 = -0.25$ per game.

The typical number of particles in macroscopic materials is of order $10^{23}$. In principle, if we knew the positions and momenta of the particles at any given instant of time, we could predict their trajectory using the laws of physics. Not only is it impossible to keep track of this information, exact knowledge of the particles' properties is often uninteresting. Similar to the coin flipping game, one is interested in the distribution of the microstates and their associated macroscopic properties. With this information, the expectation values of these properties can be calculated.

**Ensembles and the partition function** The coin flipping game reveals the main idea of statistical physics: macroscopic properties of the system are calculated as expectation values over the probability distribution $p(\boldsymbol{\sigma})$ defined over the possible microstates $|\boldsymbol{\sigma}\rangle$. The expectation value of an observable $O$ is

$$\langle O \rangle = \sum_{\boldsymbol{\sigma}} p(\boldsymbol{\sigma}) O_{\boldsymbol{\sigma}}, \tag{1.1}$$

with $O_{\boldsymbol{\sigma}}$ the value of the observable associated to state $|\boldsymbol{\sigma}\rangle$, and $\sum_{\boldsymbol{\sigma}}$ the sum over all microstates. In equilibrium, the probability distribution $p(\boldsymbol{\sigma})$ is independent of time. This ensures that expectation values of macroscopic observables are also time independent. Which probability distribution $p(\boldsymbol{\sigma})$ is used depends on the situation, and situations are categorized in terms of ensembles. For example, a system that has a fixed number of degrees of freedom $N$, a constant energy $E$, and a constant volume $V$, is said to belong to the $(N, V, E)$ or microcanonical ensemble. The fundamental postulate of statistical mechanics states that all microstates $|\boldsymbol{\sigma}\rangle$ are equally probable. The $(N, V, E)$ ensemble corresponds to an isolated system in equilibrium. As such, all states $|\boldsymbol{\sigma}\rangle$ that have energy $E$ have equal probability, such that

$$p_{(N,V,E)}(\boldsymbol{\sigma}) = \frac{1}{\Omega(E)}, \tag{1.2}$$

where $\Omega(E)$ is the number of states with energy $E$. In the $(N, V, E)$ ensemble, the states $|\boldsymbol{\sigma}\rangle$ for which $E_{\boldsymbol{\sigma}} \neq E$ have zero occupation probability. In practice, however, the total energy $E$ often fluctuates, whereas the temperature $T$ is constant. This is the case when the system can exchange energy with its environment. Such systems belong to the $(N, V, T)$ or canonical ensemble, with the probability distribution $p_{(N,V,T)}(\boldsymbol{\sigma})$ given by the Boltzmann distribution

$$p_{(N,V,T)}(\boldsymbol{\sigma}) = \frac{e^{-E_{\boldsymbol{\sigma}}/k_B T}}{\sum_{\boldsymbol{\sigma}} e^{-E_{\boldsymbol{\sigma}}/k_B T}} \equiv \frac{e^{-E_{\boldsymbol{\sigma}}/k_B T}}{Z}, \tag{1.3}$$

with $k_B$ the Boltzmann constant and the partition function $Z$ defined as $Z \equiv \sum_{\boldsymbol{\sigma}} e^{-E_{\boldsymbol{\sigma}}/k_B T}$. It is customary to define $\beta \equiv 1/(k_B T)$ the inverse temperature. The partition function sums over all microstates $|\boldsymbol{\sigma}\rangle$ and associates a Boltzmann weight $e^{-\beta E_{\boldsymbol{\sigma}}}$ to each of them. It is the central object of statistical physics, since it contains all the information about the system. The exponential suppression in Eq. (1.3) means that it is unlikely that any of the states with $E_{\boldsymbol{\sigma}} \gg k_B T$ are occupied, whereas the states with $E_{\boldsymbol{\sigma}} \leq k_B T$ have a decent chance of being populated. In the zero temperature limit $T \to 0$, the probability distribution is dominated by the ground state, i.e. the state with lowest energy $E_{gs} = \min_{\boldsymbol{\sigma}}(E_{\boldsymbol{\sigma}})$.

**Entropy and free energy** Entropy is often seen as a measure of disorder, randomness, or uncertainty. It is defined by the formula [2]

$$S = -k_B \sum_{\boldsymbol{\sigma}} p(\boldsymbol{\sigma}) \log p(\boldsymbol{\sigma}). \tag{1.4}$$

In the microcanonical ensemble Eq. (1.2), the expression reduces to $S = k_B \log \Omega$, such that it counts the (log of the) number of microstates with fixed energy. The second law of thermodynamics states that the entropy always increases. An isolated system has reached thermodynamic equilibrium whenever its entropy is maximal. The increase of entropy leads to the concept of irreversible and spontaneous processes; as such, entropy is said to determine the arrow of time. The different ensembles are unified by the fact that they maximize the entropy subject to a set of constraints (e.g. constant $N, V$ and $T$ in the canonical ensemble).

An important quantity of the canonical ensemble is the free energy, defined by

$$F = \langle E \rangle - TS. \tag{1.5}$$

Loosely speaking, the minimization of the free energy captures the idea of two competing forces. On the one hand, the system moves towards the state that minimizes the energy. On the other hand, the system tries to maximize the entropy. Interestingly, a decrease in energy is often met by a decrease in entropy and *vice versa*. The same goes for increasing the energy and the entropy. The relative importance of the competing forces is controlled by the temperature $T$; at high temperatures, it is more beneficial for the system to attain a state of high entropy. The large number of high energy states outweigh the low number of low-lying states. At low temperatures $T \approx 0$ the competition vanishes, and minimization of the free energy $F$ corresponds to minimization of the energy $E$.

### 1.1.2 The Ising model

The Ising model [3] is a prototypical model for magnetism in statistical physics. It consists of $N$ sites in a $d$-dimensional lattice. The degrees of freedom live on the lattice sites, i.e. the motion is frozen such that only the direction of magnetic spins remain. The model has an apparent simplicity, but it exhibits non-trivial behaviour. Therefore, it forms the ideal testbed for new methods. The Ising model is defined by the energy function

$$E = -J \sum_{\langle ij \rangle} s_i s_j - h_z \sum_i s_i, \tag{1.6}$$

where $\sum_{\langle ij \rangle}$ denotes the sum over all nearest neighbour pairs in the lattice. Two spins are nearest neighbours if they are closer (or equally close) to each other than to any other spins. The notation $\sum_i \equiv \sum_{i=1}^{N}$ denotes the sum over all possible values of $i$, but the limits are often omitted to simplify the notation. The spins $s_{i \in \{1,\dots,N\}}$ are discrete variables $s_i = \pm 1$ and can be considered to point up $(+1)$ or to point down $(-1)$ the $z$-axis. The first term of Eq. (1.6) corresponds to interactions between pairs of spins. Depending on the sign of $J$, the interaction is either ferromagnetic $(J > 0)$ or antiferromagnetic $(J < 0)$. The second term is due to an external magnetic field of strength $h_z$, and it tries to align the spins in a certain direction along the $z$-axis (depending on the sign of $h_z$). The microstates of the system are the $2^N$ realizable configurations of spins.

**Phase transitions and the order parameter** In statistical physics, macroscopic behaviour is described by the phase of the system. An order parameter is introduced to measure in which phase the system resides. For the Ising model with $J > 0$, this corresponds to the average magnetization $m$, given by

$$m = \frac{1}{N} \sum_i \langle s_i \rangle \,. \tag{1.7}$$

If the spin directions are random, i.e. the system is in the disordered phase, then the average magnetization is approximately zero $m \approx 0$. If the spins are aligned, the system has a non-zero magnetization $m$, indicating the ordered phase.

A phase transition is identified by a drastic change of the order parameter $(m \approx 0 \to 0 < |m| \leq 1)$ in response to a small perturbation of the system. For the Ising model with $h_z = 0$ and $J > 0$, the phase transition is related to the temperature $T$. Above some critical temperature $T > T_c$, the system is in the disordered phase $m \approx 0$ (the entropy term of Eq. (1.5) dominates). If the temperature is decreased such that $T < T_c$, the system transitions into the ordered phase of aligned spins $m \approx \pm 1$ (the energy term of Eq. (1.5) dominates). The two-dimensional $(d = 2)$ model has a continuous phase transition at a finite critical temperature $T_c > 0$. Interestingly, the one-dimensional $(d = 1)$ model has no phase transition. Thermal fluctuations drive the system in the disordered phase for any non-zero temperature $T$.

**Exact solution of the Ising chain** Consider the Ising chain, the one-dimensional Ising model $(d = 1)$. We impose periodic boundary conditions, so that $s_{N+1} \equiv s_1$ and the lattice is circular. The second term of Eq. (1.6) can be rewritten as $h_z \sum_i s_i =$

$\frac{h_z}{2}\sum_i(s_i+s_{i+1})$. We work in the $(N, h_z, T)$ ensemble, with a probability distribution similar to that of Eq. (1.3). The partition function is [4]

$$Z = \sum_{s_1=\pm1} \cdots \sum_{s_N=\pm1} \prod_{i=1}^{N} \exp\left(\beta J s_i s_{i+1} + \frac{\beta h_z}{2}(s_i + s_{i+1})\right). \tag{1.8}$$

This can be written as a product of matrices. Using the notation of quantum mechanics, we write

$$\langle s_i|\boldsymbol{T}|s_{i+1}\rangle \equiv \exp\left(\beta J s_i s_{i+1} + \frac{\beta h_z}{2}(s_i + s_{i+1})\right), \tag{1.9}$$

where we defined the transfer matrix $\boldsymbol{T}$ as

$$\boldsymbol{T} = \begin{pmatrix} e^{\beta J+\beta h_z} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J-\beta h_z} \end{pmatrix}. \tag{1.10}$$

With this identification of matrices, the partition function Eq. (1.8) becomes

$$Z = \mathrm{Tr}(\langle s_1|\boldsymbol{T}|s_2\rangle \langle s_2|\boldsymbol{T}|s_3\rangle \ldots \langle s_N|\boldsymbol{T}|s_1\rangle) = \mathrm{Tr}(\boldsymbol{T}^N), \tag{1.11}$$

where the trace reflects the periodic boundary conditions. To obtain the partition function, one computes the eigenvalues of the transfer matrix $\boldsymbol{T}$, which are

$$\lambda_\pm = e^{\beta J}\cosh\beta h_z \pm \sqrt{e^{2\beta J}\cosh^2\beta h_z - 2\sinh 2\beta J}, \tag{1.12}$$

with $\lambda_- < \lambda_+$. The partition function is thus

$$Z = \lambda_+^N + \lambda_-^N = \lambda_+^N\left(1 + \frac{\lambda_-^N}{\lambda_+^N}\right) \approx \lambda_+^N, \tag{1.13}$$

where we used $\lambda_-^N/\lambda_+^N \approx 0$ for large $N$. Using the partition function, we can calculate all kinds of interesting quantities. For example, the average magnetization Eq. (1.7) at $h_z = 0$ is given by

$$m = \frac{1}{N\beta}\frac{\partial \log Z}{\partial h_z}\bigg|_{h_z=0} = \frac{1}{\lambda_+\beta}\frac{\partial \lambda_+}{\partial h_z}\bigg|_{h_z=0} = 0\,. \tag{1.14}$$

This means that if there is no external field $h_z = 0$, then there is no magnetization $m = 0$. Even though the $J > 0$ interaction term minimizes the energy by having aligned spins, thermal fluctuations dominate at any temperature, and the system is in the disordered phase. For more information on the Ising chain and the transfer matrix approach, we refer to Refs. [4, 5].

**Exact solutions are rare** An exact solution for the two-dimensional Ising model for $h_z = 0$ has been found by Onsager [6], and it is notoriously complex. There is a large body of literature about "toy-models" such as the Ising model, but there are only a small number of exact solutions. The sum in the partition function, e.g. in Eq. (1.3), runs over all possible microstates. The number of microstates grows exponentially with the number of degrees of freedom $N$. This means that an exact evaluation of the partition functions becomes intractable for large systems. Therefore, one often resorts to approximations, e.g. mean field theory or cluster expansions [4]. A modern approach is to tackle the problem using computational methods, which often rely on Monte Carlo simulation [7]. Contemporary research also explores machine learning approaches to the Ising model [8].

## 1.2 Quantum many-body physics

### 1.2.1 Introduction to the formalism

**The wave function and the product basis** Quantum mechanics describes the physics of microscopic degrees of freedom, e.g. the properties of electrons and atoms. A quantum mechanical object is described by a quantum state or wave function $|\Psi\rangle$, which lives in a Hilbert space $\mathcal{H}$. The wave function of a single particle can be expanded in a single particle basis $|i\rangle$ of $\mathcal{H}$. In this work, we assume $i = 1, \ldots, D$, with a finite dimension $D = \dim(\mathcal{H})$. Quantum many-body physics deals with zero-temperature systems of interacting quantum mechanical particles. The Hilbert space $\mathcal{H}^{(N)}$ of $N$ distinguishable particles is the tensor product of $N$ copies of the single particle Hilbert space $\mathcal{H}^{(N)} = \mathcal{H}^{\otimes N}$. A convenient basis for $\mathcal{H}^{(N)}$ is given by the tensor-product basis

$$|i_1\, i_2\, \ldots\, i_N\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_N\rangle\,, \tag{1.15}$$

and we have $\mathcal{D} = \dim(\mathcal{H}^{(N)}) = D^N$. A general wave function of the $N$-particle system can be expanded in the tensor product basis

$$|\Psi\rangle = \sum_{\{i_1, i_2, \ldots, i_N\}}^{\mathcal{D}} \psi(i_1, i_2, \ldots, i_N)\, |i_1\, i_2\, \ldots\, i_N\rangle\,, \tag{1.16}$$

where $\sum_{\{i_1,i_2,\ldots,i_N\}}^{\mathcal{D}}$ denotes the sum over all basis states, and $\psi(i_1, i_2, \ldots, i_N)$ are complex expansion coefficients. To fully determine the wave function, one has to specify the complex amplitudes. Equation 1.16 reveals the difficulty of dealing

with quantum systems that involve a large number of particles. Namely, the dimension of the $N$-particle system $\mathcal{D} = D^N$ scales exponentially with the number of particles $N$. For indistinguishable particles, an (anti-)symmetrization procedure of the tensor product basis is in order, which reduces the number of physically realizable states.

**Lattice problems and spin** In this thesis, we deal with spatially localized spins on a lattice, therefore the degrees of freedom are distinguishable. For spin-$1/2$ particles, the local Hilbert space dimension is $\dim(\mathcal{H}) = 2$. In this case, the standard single particle basis is the $s_z$-basis, where the state of the particle is labeled by the eigenvalues of the $\hat{s}_z$-operator, i.e. the spin-projection operator on the $z$-axis. The spin operators in the three $(x, y, z)$-directions are related to the Pauli operators. In the standard $s_z$-basis, these are given by

$$\hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.17}$$

The Pauli operators satisfy the commutation relations

$$[\hat{\sigma}^a, \hat{\sigma}^b] = 2i\varepsilon^{abc}\hat{\sigma}^c, \tag{1.18}$$

where $\varepsilon^{abc}$ is the Levi-Civita symbol. The anti-commutation relations are

$$\{\hat{\sigma}^a, \hat{\sigma}^b\} = 2\delta_{a,b}\hat{I}, \tag{1.19}$$

where $\delta_{a,b}$ is the Kronecker delta and $\hat{I}$ is the identity operator. The spin operators are the Pauli operators multiplied by a factor of $\frac{\hbar}{2}$, such that we have $\hat{s}_{x,y,z} = \frac{\hbar}{2}\hat{\sigma}^{x,y,z}$. It is customary to set the (reduced) Planck constant to unity $\hbar = 1$, which is also done here. The two eigenvalues of the $\hat{s}_z$-operator are $s_z = \pm\frac{1}{2}$, and the corresponding states are interpreted as either spin up ($|s_z = +\frac{1}{2}\rangle = |\uparrow\rangle$) or spin down ($|s_z = -\frac{1}{2}\rangle = |\downarrow\rangle$). Depending on the context, we will denote the states by the eigenvalues of the $\hat{\sigma}^z$-operators. We write the direction as a superscript to preserve the subscript location for the lattice index. The many-body wave function is written as a superposition

$$|\Psi\rangle \equiv \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma})\,|\boldsymbol{\sigma}\rangle = \sum_{\{\sigma_i^z\}} \psi(\sigma_1^z, \sigma_2^z, \ldots, \sigma_N^z)\,|\sigma_1^z\sigma_2^z\ldots\sigma_N^z\rangle = \sum_{n=1}^{2^N} \psi(\boldsymbol{\sigma}_n)\,|\boldsymbol{\sigma}_n\rangle\,, \tag{1.20}$$

where a configuration $\boldsymbol{\sigma}$ is expressed in terms of the eigenvalues $\sigma_i^z = \pm 1$ of the single spin $\hat{\sigma}_i^z$-eigenfunctions at sites $i$.

If an operator acts on one or more spins, we implicitly assume that the identity operator acts on the remaining spins. For example, the $\hat{\sigma}_i^z$-operator acts on a many-body state as

$$\hat{\sigma}_i^z = \hat{I}_1 \otimes \hat{I}_2 \otimes \cdots \otimes \hat{I}_{i-1} \otimes \hat{\sigma}_i^z \otimes \hat{I}_{i+1} \otimes \cdots \otimes \hat{I}_N, \tag{1.21}$$

where $\hat{I}_j$ is the identity operator acting on the $j$-th spin.

**Hamiltonians and the Schrödinger equation** Models are specified by a Hamiltonian $\hat{H}$. Together with the identity operator $\hat{I}$, the Pauli operators Eq. (1.17) form a complete basis for the Hermitian operators on $\mathcal{H}^{(N)}$. Therefore, the Hamiltonian can be expressed in terms of Pauli operators. If $\hat{H}$ is time independent, the non-relativistic many-body problem is solved by the time-independent Schrödinger equation

$$\hat{H} \left|\Psi_n\right\rangle = E_n \left|\Psi_n\right\rangle . \tag{1.22}$$

Upon choosing a suitable basis $|\boldsymbol{\sigma}\rangle$, the Hamiltonian $\hat{H}$ can be written as a matrix $H_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = \langle\boldsymbol{\sigma}'|\hat{H}|\boldsymbol{\sigma}\rangle$. In the language of linear algebra, the Schrödinger equation Eq. (1.22) is an eigenvalue problem. The eigenvectors are states $|\Psi_n\rangle$ and the eigenvalues are corresponding energies $E_n$. The problem is thus solved by diagonalization of the $\mathcal{D} \times \mathcal{D}$ Hamiltonian matrix. In general, the number of operations needed to diagonalize a matrix of dimension $\mathcal{D}$ scales as $\mathcal{O}(\mathcal{D}^3)$. Therefore, one often resorts to approximations, e.g. perturbation theory [9]. In this work, we will approximate the quantum wave function by using neural networks as variational ansätze, which was first introduced in Ref. [10].

### 1.2.2 The transverse field Ising model

A direct translation of the classical Ising model Eq. (1.6) into the language of quantum mechanics gives a Hamiltonian $\hat{H} = -J \sum_{\langle ij \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h_z \sum_i \hat{\sigma}_i^z$. The eigenstates of this Hamiltonian correspond to product states (Eq. (1.15)) in the eigenbasis of the $\hat{\sigma}^z$ operators, i.e. there is no superposition of the form Eq. (1.20). True quantum behaviour is obtained by adding a non-commutative term. From Eq. (1.18), it is clear that the $\hat{\sigma}^x$ operator does not commute with $\hat{\sigma}^z$. By adding a term involving $\hat{\sigma}^x$ and putting the longitudinal field term to zero $(h_z = 0)$, we arrive at the transverse field Ising model (TFIM)

$$\hat{H}_{\text{TFIM}} = -J \sum_{\langle ij \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h_x \sum_i \hat{\sigma}_i^x , \tag{1.23}$$

where $J$ is the interaction strength between pairs of spins and $h_x$ is the transverse field strength. The fact that the operators $\hat{\sigma}^z$ and $\hat{\sigma}^x$ do not commute (Eq. (1.18)) means that the individual terms of the Hamiltonian do not have a common eigenbasis. Thus, a state that minimizes one of the terms cannot simultaneously minimize the other term. In order to minimize the total Hamiltonian $\hat{H}_{\text{TFIM}}$ in a given basis, we need a superposition of states as in Eq. (1.20). One says that non-commuting terms lead to frustration at the quantum level.

The superposition of the form Eq. (1.20) means that expectation values of quantum wave functions are calculated as weighted averages over the basis states. This is remarkably similar to how expectation values are calculated in statistical physics, namely by use of the partition function Eq. (1.3). However, the uncertainty in the quantum system is not due to thermal fluctuations, since we are dealing with the system at zero temperature $T = 0$. Rather, the thermal fluctuations are replaced by quantum fluctuations, and quantum fluctuations arise whenever the eigenstates of the Hamiltonian are superpositions of basis states.

We can analyse the TFI model by introducing the dimensionless parameter $g = h_x/J$ [11]. In the limit $g = 0$, we recover the classical Ising model. The energy is minimized by the two states in which all spins are aligned, i.e. the product states $|\Psi_\pm\rangle = |\sigma^z = \pm 1\rangle^{\otimes N}$. This corresponds to the ordered phase, with a magnetization $\langle\Psi_\pm|\sum_i \hat{\sigma}_i^z|\Psi_\pm\rangle = \pm N$. In the limit $g \gg 1$, the transverse field $h_x$ dominates. In this case, there is a unique ground state given by $|\Psi_\rightarrow\rangle = |\rightarrow\rangle^{\otimes N}$, where $|\rightarrow\rangle = (|\uparrow\rangle + |\downarrow\rangle)/\sqrt{2}$. This ground state is an equal superposition of all possible spin states in the $s_z$-basis, so the regime corresponds to the disordered phase with $\langle\Psi_\rightarrow|\sum_i \hat{\sigma}_i^z|\Psi_\rightarrow\rangle = 0$. Between the two limits of $g$, we expect a phase transition. Note that the role of the temperature in the classical Ising model is now fulfilled by the parameter $g$, or rather, the transverse field $h_x$. Instead of controlling the competition between energy minimization and entropy maximization in Eq. (1.5), we now control the importance of non-commuting terms in the Hamiltonian.

## 1.3 Quantum statistical physics

In order to describe quantum systems at finite temperatures $T > 0$, one combines the procedures of statistical physics (section 1.1) and quantum many-body physics (section 1.2). This enables one to express classical probability distributions over quantum states. Two sources of uncertainty are thereby combined: i) the uncertainty due to temperature $T$, i.e., states of different energies are accessible, and ii) the uncertainty due to quantum fluctuations, i.e., quantum states are expressed as a superposition over the basis states.

### 1.3.1 Introduction to the formalism

In principle, the description of $T > 0$ quantum systems does not require a new formalism. However, when dealing with a quantum system whose state is not completely known, it proves to be convenient to introduce the density operator language. Suppose that the quantum system is in a state $|\Psi_i\rangle$ with probability $p_i$, where $i$ is an index that runs over the different possible states. The density operator $\hat{\rho}$ is defined by [12]

$$\hat{\rho} \equiv \sum_i p_i \left|\Psi_i\right\rangle \left\langle\Psi_i\right| . \tag{1.24}$$

The density operator $\hat{\rho}$ is a non-negative operator with trace one. If the state of the system is known exactly, we call it a pure state; in this case, the density operator is $\hat{\rho} = |\Psi\rangle \langle\Psi|$, with $|\Psi\rangle$ the state of the system. A pure state satisfies $\mathrm{Tr}(\hat{\rho}^2) = 1$. In all other cases, the system is said to be in a mixed state, and $\mathrm{Tr}(\hat{\rho}^2) < 1$. The expectation value of an observable $\hat{O}$ is calculated as

$$\langle \hat{O} \rangle = \mathrm{Tr}(\hat{O}\hat{\rho}). \tag{1.25}$$

The density operator formalism is especially useful for describing the subsystems of a composite quantum system. Suppose that the quantum system can be divided into two subsystems $A$ and $B$. The density matrix of the composite system is $\hat{\rho}^{AB}$. We define the reduced density matrix for subsystem $A$ to be

$$\hat{\rho}^A \equiv \mathrm{Tr}_B(\hat{\rho}^{AB}), \tag{1.26}$$

where $\mathrm{Tr}_B$ is the partial trace over subsystem $B$. This partial trace is defined by

$$\mathrm{Tr}_B(|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2|) \equiv |a_1\rangle \langle a_2| \, \mathrm{Tr}_B(|b_1\rangle \langle b_2|), \tag{1.27}$$

where $|a_1\rangle$ and $|a_2\rangle$ ($|b_1\rangle$ and $|b_2\rangle$) are any two vectors in the Hilbert space of $A$ ($B$). The fact that the reduced density matrix $\hat{\rho}^A$ is a description for subsystem $A$ comes from the observation that it provides the correct measurement statistics for measurements on $A$ [12].

A quantum system in thermal equilibrium is described by the density operator in the canonical ensemble (section 1.1.1)

$$\hat{\rho}_{(N,V,T)} = \frac{e^{-\beta\hat{H}}}{\mathrm{Tr}(e^{-\beta\hat{H}})} = \frac{e^{-\beta\hat{H}}}{Z_q}, \tag{1.28}$$

where we defined the quantum mechanical partition function

$$Z_q \equiv \mathrm{Tr}(e^{-\beta\hat{H}})\,. \tag{1.29}$$

The exponential function of an operator $\hat{O}$ is interpreted as a power series

$$\exp(\hat{O}) = \sum_{i=0}^{\infty} \frac{1}{i!}\hat{O}^i. \tag{1.30}$$

The quantum mechanical partition function Eq. (1.29) is similar to the classical partition function Eq. (1.3). In the eigenbasis of $\hat{H}$, the diagonal elements of $\hat{\rho}_{(N,V,T)}$ are the populations of the energy eigenstates at thermal equilibrium. In the limit $T = 0$, only the ground state is populated and the ensemble is in a pure state (if there is no ground state degeneracy). On the other hand, each energy eigenstate is equally populated when $T \to \infty$.

### 1.3.2 Quantum-classical correspondence

**Derivation for the single spin TFI model** Consider a zero-dimensional quantum system ($d = 0$), i.e. a single spin, described by the Hamiltonian [11]

$$\hat{H} = -h_x\hat{\sigma}^x - h_z\hat{\sigma}^z\,. \tag{1.31}$$

This Hamiltonian can be interpreted as the single spin TFI model Eq. (1.23) with non-zero longitudinal field $h_z$. This Hamiltonian consists of two terms $\hat{h}_0 = -h_x\hat{\sigma}^x$ and $\hat{h}_1 = -h_z\hat{\sigma}^z$, such that the canonical partition function can be written as

$$Z_q = \mathrm{Tr}(e^{-\beta\hat{H}}) = \mathrm{Tr}(e^{-\beta\sum_{i=0}^{1}\hat{h}_i}). \tag{1.32}$$

We can relate the exponential of a sum of non-commuting terms to the product of the individual exponentials by using the Suzuki-Trotter decomposition [13]

$$e^{-\beta \sum_i \hat{h}_i} = \lim_{n\to\infty} \left( \prod_i e^{-\frac{\beta}{n}\hat{h}_i} \right)^n . \tag{1.33}$$

Henceforth, we denote $\varepsilon = \frac{\beta}{n}$. The trace in Eq. (1.32) is evaluated in the $\sigma^z$-basis, and we substitute Eq. (1.33) in Eq. (1.32) to obtain

$$Z_q = \mathrm{Tr}(e^{-\beta\hat{H}}) = \lim_{n\to\infty} \sum_{\sigma_0^z} \langle \sigma_0^z | \left( \prod_i e^{-\varepsilon \hat{h}_i} \right)^n | \sigma_0^z \rangle , \tag{1.34}$$

where $\sum_{\sigma_0^z}$ is the sum over spin configurations $\sigma_0^z = \pm 1$. For each of the $n$ factors in Eq. (1.34), we insert a resolution of the identity $\hat{I} = \sum_{\sigma_i^z} |\sigma_i^z\rangle \langle \sigma_i^z|$. The partition function now becomes

$$Z_q = \mathrm{Tr}(e^{-\beta\hat{H}}) = \lim_{n\to\infty} \sum_{\{\sigma_i^z\}} \langle \sigma_0^z | e^{-\varepsilon\hat{H}} | \sigma_1^z \rangle \langle \sigma_1^z | e^{-\varepsilon\hat{H}} | \sigma_2^z \rangle \ldots \langle \sigma_n^z | e^{-\varepsilon\hat{H}} | \sigma_0^z \rangle , \tag{1.35}$$

where $\sum_{\{\sigma_i^z\}}$ is the sum over spin configurations $\sigma_i^z = \pm 1$ for each $i = 0, 1, \ldots, n$. We now take one of the factors in Eq. (1.35), and write out the exponential

$$\begin{aligned} \langle \sigma_i^z | e^{-\varepsilon\hat{H}} | \sigma_{i+1}^z \rangle &= \langle \sigma_i^z | e^{\varepsilon h_z \hat{\sigma}^z} e^{\varepsilon h_x \hat{\sigma}^x} | \sigma_{i+1}^z \rangle \\ &= e^{\varepsilon h_z \sigma_i^z} \langle \sigma_i^z | \cosh(\varepsilon h_x)\, \hat{I} + \sinh(\varepsilon h_x)\, \hat{\sigma}^x | \sigma_{i+1}^z \rangle \\ &= e^{J'\sigma_i^z \sigma_{i+1}^z + h'\sigma_i^z + constant} . \end{aligned} \tag{1.36}$$

In Eq. (1.36), we used the power series of $\exp(\varepsilon h_x \hat{\sigma}^x)$ combined with the fact that $(\hat{\sigma}^x)^i = \hat{I}$ if $i$ is even and $(\hat{\sigma}^x)^i = \hat{\sigma}^x$ otherwise. We also introduced new variables $J' = -\frac{1}{2}\log(\tanh(\varepsilon h_x))$ and $h' = \varepsilon h_z$. The additional constant introduces a proportionality factor which can be omitted. We are thus left with

$$Z_q = \lim_{n\to\infty} \sum_{\{\sigma_i^z\}} e^{\sum_{i=1}^n J'\sigma_i^z\sigma_{i+1}^z + h'\sigma_i^z} , \tag{1.37}$$

which is the partition function of the one-dimensional classical Ising model with periodic boundary conditions Eq. (1.8), in the thermodynamic limit $n \to \infty$. Indeed, we identify the number of spins $n = N$, the nearest neighbour interaction $J' = J$, and the external field $h' = h_z$. The result of Eq. (1.36) corresponds to the transfer matrix Eq. (1.9) of the one-dimensional Ising model (section 1.1.2).

**Generalization and sign problem** The ground state regime of a quantum system is obtained in the limit $\beta \to \infty$ (section 1.3.1). Because of the identification $N = \beta/\varepsilon$, we see, in our example above, that this corresponds to the thermodynamic limit $N \to \infty$ of the classical system. The correspondence between the ground state of a $d$-dimensional quantum system and a $(d+1)$-dimensional finite temperature classical system in the thermodynamic limit holds for any dimension $d$ and for any Hamiltonian $\hat{H}$ [14].

One might now ask why we would be interested in studying quantum ground states, since evidently they can all be mapped to classical systems. A first reason is that the concepts that arise in the language of quantum mechanics are extremely useful (e.g. the concept of entanglement). But there is a more important aspect; it is not guaranteed that the Boltzmann weights in the classical partition function are positive or real. Conceptually, this is frustrating, since the interpretation of a classical probability distribution is lost. The consequences are however more profound; numerical methods such as quantum Monte Carlo [7] rely explicitly on the probabilities obtained by the quantum-classical mapping. Negative Boltzmann weights come with an exponential growth of the statistical error, which defeats the advantage of Monte Carlo methods. This is known as the sign problem, which has been proven to be NP-hard [15].

## 1.4 Symmetries in physics

### 1.4.1 General description

A feature, an object, or physical law obeys a symmetry if it remains invariant or unchanged under some transformation. Symmetries can be found in our everyday lives, for example the bilateral symmetry of the human face. The cosmological principle states that the universe is isotropic and homogeneous; roughly speaking, this means that the universe is the same for all observers, and it looks the same in all directions. In the Standard Model, particles are categorized based on their (approximate) symmetries. We find symmetries everywhere in Nature; they are a necessary component of theories of physics.

**Group theory** Symmetries are described mathematically in terms of group theory. A group is a set $G = \{g_1, g_2, \dots\}$ of group elements $g_i \in G$, that is structured according to the following rules [16]

- multiplication of two group elements yields a group element $g_i g_j = g_k \in G$,
- multiplication is associative $(g_i g_j) g_k = g_i (g_j g_k)$,
- there exists an identity element $I \in G$ such that $g_i I = I g_i : \forall g_i \in G$,
- each element $g_i \in G$ has an inverse $g_i^{-1} \in G$ such that $g_i g_i^{-1} = g_i^{-1} g_i = I$.

To be entirely correct, the multiplication is to be understood as a general binary operator $(\circ)$. The binary operator can be either additive $g_i \circ g_j = g_i + g_j$ or, as we had assumed here, multiplicative $g_i \circ g_j = g_i g_j$.

There are many different types of groups. If the number of group elements $g \in G$ is finite, we say that the group is discrete. An example is the group that describes the symmetries of a square: the dihedral group $D_4$. Continuous symmetries with an infinite amount of elements are described by Lie groups, and their infinitesimal symmetry motions appear as Lie algebras [17]: the elements of the continuous group are described as functions of a certain number of continuous parameters. The group elements can be written in terms of an equal number of generators, infinitesimal operators, which satisfy the multiplication law represented by so-called Lie brackets. In this Lie algebra, there is a structure constant that characterizes the structure of the group. An example will be discussed in section 1.4.2.

Furthermore, we can differentiate between abelian groups, where $g_i g_j = g_j g_i$, and non-abelian groups, where the commutative relation does not hold $g_i g_j \neq g_j g_i$. For example, rotations about different axes are non-commutative.

**Representations of groups** The groups themselves are abstract notions. In order to work with them, we define representations. A representation is a homomorphism $G \to GL_n(\mathcal{V})$ to the general linear group of a vector space $\mathcal{V}$. Roughly speaking, a representation relates elements $g \in G$ of the abstract group to $n \times n$ invertible matrices, which allows us to understand $G$ by how it acts on $\mathcal{V}$ [16]. The binary operator $\circ$ gets translated into matrix multiplication. If the vector space $\mathcal{V}$ corresponds to the complex vector space $\mathbb{C}$, then the coefficients of the matrices are complex numbers. We call the representation irreducible if it has no non-trivial invariant subspaces. In terms of invertible matrices, irreducible means that the matrix cannot be divided into blocks. Irreducible representations (irreps) are especially useful, since reducible representations can be built from them.

The names of the groups are related to the properties of the associated matrices. For example, the "special orthogonal" group SO(3) of rotations about the origin of three-dimensional Euclidean space $\mathbb{R}^3$ is represented by $3 \times 3$ orthogonal matrices with determinant $1$. The "unitary group of degree n" is denoted U(n), and corresponds to the group of $n \times n$ unitary matrices. In the simplest case U(1), it consists of all complex numbers with absolute value $1$.

**Conservation laws** Noether's theorem [18] states that every differentiable symmetry of the action of a physical system with conservative forces has a corresponding conservation law. Without delving into any details or proofs, we appreciate the fundamental consequences of this theorem. Put simply, if a system obeys a continuous symmetry, there is a corresponding quantity whose value is conserved in time. This allows us to search for symmetries whenever we observe conserved quantities and *vice versa*. Since conserved quantities are an important concept in virtually all of physics [19], the theorem provides another motivation for studying the symmetries of a system. Some notable examples are (spatial) translational symmetry $\leftrightarrow$ conservation of momentum, time translational symmetry $\leftrightarrow$ conservation of energy, and rotational symmetry $\leftrightarrow$ conservation of angular momentum.

### 1.4.2 Symmetries in quantum mechanics

In the language of quantum mechanics, the matrix representations of symmetry groups correspond to unitary operators $\hat{U}$ that act on the state space (i.e. the wave function Eq. (1.16)) [20]. A state $|\Psi\rangle$ is symmetric with respect to a unitary operator $\hat{U}$ if the transformed state is identical to the original state, up to a phase factor $\hat{U}|\Psi\rangle = e^{i\theta}|\Psi\rangle$. The unitary transformation $\hat{U}$ is a symmetry of the Hamiltonian $\hat{H}$ if their commutator vanishes $[\hat{H}, \hat{U}] = 0$. This implies that $\hat{H}$ and $\hat{U}$ have a common eigenbasis. Thus, the eigenstates of the Hamiltonian can be given a label for the particular irrep of the symmetry group according to which they transform. These labels are referred to as quantum numbers.

For indistinguishable particles, the Hamiltonians are invariant under permutations. The permutation group has only two irreps, and these correspond to fully symmetric and antisymmetric wave functions. This is at the core of the so-called quantum statistics and gives rise to the different statistical behaviour of bosons (symmetric) and fermions (antisymmetric) [21]. Measurements of quantum systems are insensitive to global phase changes, i.e., the wave function is symmetric under the U(1) circle group. This U(1) symmetry is related to the conservation of charge.

**An example: symmetry of the TFI model** Consider the transverse field Ising model Eq. (1.23). The TFI model has spin-flip symmetry, since $\hat{H}_{TFIM}$ commutes with the operator $\hat{P} = \otimes_i \hat{\sigma}_i^x$. The action of $\hat{\sigma}_i^x$ on the eigenstates of the $\hat{\sigma}_i^z$ operators is to "flip" the $i$-th spin, i.e. $\hat{\sigma}^x |\sigma^z = \pm 1\rangle = |\sigma^z = \mp 1\rangle$. This is a $\mathbb{Z}_2$ symmetry, since $\{\hat{I}, \hat{P}\}$ is closed under multiplication. If we introduce a non-zero longitudinal field $h_z \neq 0$, then this relation no longer holds; this is an example of explicit symmetry breaking. Putting $h_z = 0$, the order parameter $\hat{O} = \sum_i \hat{\sigma}_i^z$ (total magnetization) transforms as $\hat{P}\hat{O}\hat{P}^\dagger = -\hat{O}$. Therefore, if the ground state does not break the symmetry, the ground state expectation value $\langle \hat{O} \rangle$ should be zero. Recall the two ground states in the $(g = 0)$ ferromagnetic regime $|\Psi_\pm\rangle = |\pm 1\rangle^{\otimes N}$. The ground state expectation value of the total magnetization $\langle \Psi_\pm | \hat{O} | \Psi_\pm \rangle = \pm N$ is clearly non-zero. We can, however, construct the symmetric and antisymmetric superpositions $(|\Psi_+\rangle \pm |\Psi_-\rangle)/\sqrt{2}$ for which $\langle \hat{O} \rangle = 0$. The important observation is that the symmetry acts non-trivially within the ground state subspace; this is an example of spontaneous symmetry breaking. We identify this with the system being in the ordered phase. In the regime $g \gg 1$, the unique ground state $\Psi_\rightarrow = |\rightarrow\rangle^{\otimes N}$ is symmetric under $\hat{P}$ such that $\langle \Psi_\rightarrow | \hat{O} | \Psi_\rightarrow \rangle = 0$. There is thus no symmetry breaking in the paramagnetic disordered phase [11].

**SU(2) spin symmetry** The SU(2) symmetry group is a continuous non-abelian Lie group. It describes the rotational symmetry of half-integer spins, e.g. spin-$1/2$. The spin-rotation symmetry of integer spins is described by the classical rotational symmetry group SO(3), which is closely related to SU(2). The action of the symmetry operations can be written in terms of unitary operators [16]

$$\hat{R}(\boldsymbol{\theta}) = e^{i(\theta_x \hat{s}_x + \theta_y \hat{s}_y + \theta_z \hat{s}_z)} = e^{i\boldsymbol{\theta}\cdot\hat{\boldsymbol{s}}}, \tag{1.38}$$

where $\boldsymbol{\theta}$ is a normalised vector in three-dimensional space, and $\hat{s}_{x,y,z}$ are the generators of SU(2). The exponential is to be interpreted as a power series Eq. (1.30). The vector $\boldsymbol{\theta}$ denotes a rotation of magnitude $\|\boldsymbol{\theta}\| = \theta$ around the axis pointing in the direction of $\boldsymbol{\theta}$. For spin-$1/2$ degrees of freedom, which is the only case we will be considering, the generators correspond to the Pauli matrices Eq. (1.17). The corresponding Lie algebra is presented in Eq. (1.18). The Lie bracket is the usual commutator, and the structure constant is the Levi-Civita symbol. The action on a composite system of $N$ spin degrees of freedom is the direct product of the single-particle actions

$$\hat{R}(\boldsymbol{\theta}) = e^{i\boldsymbol{\theta}\cdot(\hat{\boldsymbol{s}}_1 + \hat{\boldsymbol{s}}_2 + \cdots + \hat{\boldsymbol{s}}_N)} = e^{i\boldsymbol{\theta}\cdot\hat{\boldsymbol{J}}}, \tag{1.39}$$

with $\hat{\boldsymbol{J}}$ the total angular momentum operator. Eigenstates of the SU(2) symmetry operator must therefore have well-defined total angular momentum. The spin-rotational symmetry comes with the conserved quantity of total spin $\hat{\boldsymbol{s}}_{tot} = \sum_i \hat{\boldsymbol{s}}_i$. The SU(2) symmetry and angular momentum states are discussed more extensively in section 3.3.2.

# Introduction to machine learning

# 2

Machine learning (ML) is the branch of artificial intelligence focused on computer algorithms that solve a problem without explicitly being coded by the logical steps which lead to the solution. For this to be achieved, the algorithm processes data in a guided manner. A formal definition was provided by Tom M. Mitchell: "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*" [22]. Machine learning algorithms often require a lot of data in order to perform reliably. Furthermore, some of the algorithms are computationally expensive. The usability of machine learning is thus strongly dependent on computational resources and availability of suitable datasets.

In the past decade, machine learning techniques have been applied to various areas of research and enabled innovative technologies. Self-driving cars, customer-recommendation systems, speech-recognition, and virtual assistance devices rely explicitly on ideas from the ML industry. The disruption of the novel technology has been compared to the impact of computers in the 1980s and 1990s, and it is expected to be of ever increasing relevance in perspective of a society based around big data, automisation, and predictive capabilities [23]. The power of ML comes from the exceptional ability to recognize complex patterns and the considerable speed-up of classification or decision-making tasks as compared to classical methods. Machine learning has gained attention from influential companies like Google, Apple, Intel and Microsoft, both because of its short-term productivity and its promising long-term insightfulness in many interesting exploratory fields. Conceptual developments in ML are often motivated by ideas from mathematics, statistics and statistical physics, therefore it is especially synergistic with the physical sciences [24]. Physical insights and applications to domains of physics allow cross-fertilization between the two fields, realizing unprecedented advances in artificial intelligence and physical problems [23].

## 2.1 The machine learning methodology

In data science, an algorithm is a sequence of statistical processing steps. In machine learning, algorithms are *trained* by processing data, namely the training set, in a guided but independent manner. The goal of a ML algorithm is to provide a model which can be generalized to new (unseen) data, the so-called test set. First, the general workflow of machine learning will be discussed. Second, we will discuss what it means to train a model. Finally, an example is given in which the theoretical notions are translated into a practical application.

### 2.1.1 Typical machine learning workflow

Usual machine learning workflows look as follows:

1. **Problem definition.** Firstly, data has to be gathered and the problem (or task $T$) has to be defined and translated into mathematical terms. This means that a target has to be formulated, together with a measure that indicates how well the machine handles the task. The latter is most often referred to as the cost function (or performance measure $P$), and can be seen as the value that indicates the error of the machine.

2. **Model selection.** The next step is selecting the appropriate model (or algorithm) for the task. This step also depends on the amount and the quality of the data that is available, as some algorithms need more data to perform reliably as compared to others. When a suitable model is chosen, the hyperparameters (if any) need to be carefully considered. The so-called bias-variance tradeoff plays a crucial role during model selection: this tradeoff implies that a model which is less biased often shows greater variance and *vice versa*. High bias models do not change their prediction much when new data points are introduced, meaning the model shows low variance. However, high bias means the search space (or hypothesis space) is restricted and thus the expressiveness of the model is limited, often leading to inadequate performance. If a low bias model is chosen, the reverse reasoning applies: the model performs well because it is more adaptive, but introducing a new data point changes the outcome of the algorithm drastically (meaning low bias and high variance). The ideal scenario is having a model with low bias and low variance, such that it can accurately approximate the underlying distribution of the data and yet be robust and generalizable.

3. **Training.** Machine learning algorithms contain variational parameters. Training the model means feeding training data (experiences $E$) and adjusting the variational parameters in a way that improves the performance measure $P$. Training is thus the phase wherein the algorithm "learns", i.e., it is tweaked (often iteratively) in order to perform better at the provided task.

4. **Testing.** Extensive training typically leads to a model which performs well on the training data. However, practical uses of machine learning algorithms require the model to perform well on unseen test data. It is therefore of great interest to acquire the best possible performance on the test set. In many cases, too much training leads to overfitting: extensively training a model leads to ever-increasing performance on the training set, but the performance on the test set starts to decrease. The main reason for this behaviour is that the model learns the noise in the training data.

We now discuss three additional remarks on the outline provided above. (i) Since models need to perform well on unseen data, the original dataset is often separated in three parts, namely training - validation - and test sets. The validation set is then used to validate the model performance during training, possibly guiding subsequent actions, e.g. in model refinement. The test set is only used at the very last phase of the machine learning pipeline, that is during final performance measurements. If at any point the test set slips into the model, the final results are biased. (ii) Most algorithms come with hyperparameters, i.e., parameters not varied during training but possibly important for the model behaviour. The optimal hyperparameters are then selected by doing a parameter sweep: a series of simulations (steps 3 & 4) are carried out using sets of hyperparameters and the results are compared. The set that performs best on the validation set is then taken to define the model. (iii) Methods such as early stopping are used in order to avoid overfitting: as the training error generally decreases monotonically during training, the error on the validation set is the appropriate measure of performance. Early stopping is a method that stops training of the model when the validation error increases, thereby averting overfitting.

### 2.1.2 The concept of learning

As we have already noted in step 3, we say that the model "learns" during the training phase. In order to introduce the concept, denote the output of the model for the input $\mathbf{x}_i$ by $\hat{\mathbf{y}}_i = f_{\mathcal{W}}(\mathbf{x}_i)$, where data points (or samples) are labeled by index $i$. Note that the output of the model depends on its variational parameters $\mathcal{W}$. The loss

function $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ takes as inputs the target $\mathbf{y}_i$ and the prediction $\hat{\mathbf{y}}_i$, and returns a single number. This number indicates the error of the model on the given data point. When the target $\mathbf{y}_i$ is not readily available, we define the loss by other means. The distinction between these two cases is set out in more detail in the forthcoming section (section 2.2).

Usually we are more interested in the overall performance of the model than its performance on an individual data point. This performance is represented by the single number returned by the cost function $\mathcal{C}$. The cost function can be obtained by extending the definition of loss functions to more data points, for example by setting $\mathcal{C} = \sum_i \mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ and normalizing. The latter can be either minimized (in which case the cost function outputs a value called the cost, error, or loss) or maximized (in which case the value is called the reward). Choosing the correct cost function is a crucial part of the machine learning workflow, and depends on the specific situation.

Thus "learning" means adjusting the variational parameters $\mathcal{W}$ in order to decrease/increase the cost function. In case of cost function minimization, this can be expressed as

$$\min_{\mathcal{W}} \left[\mathcal{C}(\mathcal{W})\right]. \tag{2.1}$$

This expression leads to a set of equations involving the derivatives of the cost function and the function representing the action of the machine $f_{\mathcal{W}}(\mathbf{x}_i)$. These equations can be hard to solve analytically, especially if the machine contains many variational parameters and is non-linear. Therefore, iterative optimization techniques are widely used. Perhaps the most well-known is the gradient descent algorithm, which is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. It defines a parameter update rule in the direction of steepest descent

$$\mathcal{W}^{t+1} = \mathcal{W}^t - \eta \cdot \nabla_{\mathcal{W}} \mathcal{C}(\mathcal{W})\Big|_{\mathcal{W}=\mathcal{W}^t}, \tag{2.2}$$

where $\eta$ denotes the learning rate, and $t$ is the iteration step. The learning rate is a chosen value which dictates the magnitude of the parameter variation. If the learning rate is big, the algorithm might miss the (local) minimum. If it is small, the optimization is slow and the algorithm might get stuck in a local minimum. The update rule Eq. (2.2) is applied iteratively until convergence is reached, meaning the variational parameters $\mathcal{W}$ do no longer change considerably during an update.

## 2.2 Machine learning tasks

Machine learning has been applied to a wide variety of tasks, each of which can be divided into one of three paradigms:

- **Supervised learning.** In supervised learning, the data points are accompanied by labels, i.e. the ground truths. Labels are the correct answers and the target output for the model. The label information is used to calculate the cost function and thus enables estimating model performance during optimization and in the testing phase. The main obstacle of supervised learning is the need for labeled datasets. Generating labeled datasets is typically time consuming and expensive, as it requires expert analysis.

- **Unsupervised learning.** On the contrary, unsupervised learning uses unlabeled datasets. The training dataset is a set of examples without a desired outcome or target associated to them. Rather, the goal of unsupervised learning is the extraction of features or patterns in the dataset and to elucidate structure. As there is no obvious target for the model to aim at, defining a meaningful cost function is one of the main difficulties in unsupervised learning. A meaningful cost function forces the model to learn the underlying structure of the dataset during optimization.

- **Reinforcement learning.** In reinforcement learning, an intelligent agent (IA) takes actions based on its environment and accumulative reward. An IA has sensors and actuators to observe and change its environment, respectively. The IA maximizes the reward by balancing exploration of uncharted territory and exploitation of current knowledge. It learns from past experiences and trains by trial and error. The typical model in reinforcement learning is the Markov decision process, which is an extension of the Markov chain by addition of actions and rewards.

Falling between supervised and unsupervised learning we have semi-supervised learning, in which the model is trained with a partially labeled dataset. This means that some amount of data points have labels whereas the rest of the data points do not. In a largely unsupervised setting, the labels can for example be incorporated in pattern detection. Supervised models can learn to extrapolate the label information to unlabeled data points and use this to improve the model.

Many different machine learning algorithms have been devised, but they usually solve similar problems. Most machine learning tasks fall into one of the following categories.

- **Classification.** Perhaps the most well-known machine learning task is supervised classification, in which a model is trained to predict the class to which a data point belongs. An example is the popular Modified National Institute of Standards and Technology (MNIST) database [25] consisting of 70 000 images of handwritten numbers. Customarily, the images are classified by a convolutional neural network which systematically extracts patterns in the pixel distributions [26]. Another popular classification model is the support vector machine (SVM) [27] (discussed in section 2.3.2).

- **Regression.** Regression analysis is a type of supervised predictive modelling which searches for correlations between the independent variables (the predictors $\mathbf{x}$) and the dependent variable (the target $y$). The simplest form is linear regression, where the fit $y = \mathbf{m}\mathbf{x} + b$ to the data points is optimized by minimizing the sum of distances between exact values and predicted values $y$. A popular example of regression problems is price prediction, for example in stocks.

- **Clustering.** The idea behind clustering is that there is some natural grouping of the data, and that this grouping can be found in an unsupervised setting. After defining a distance measure in feature space (e.g. the euclidean distance between points $\mathbf{x}$), data points are placed in the same group when they are close and/or satisfy other criteria. The clustering can function as a label, as it marks similarity. Other benefits include outlier detection and pattern recognition.

- **Dimensionality reduction.** This unsupervised technique reduces the dimensionality of a dataset by removing superfluous variables or by transforming the variables. This is interesting for data compression and preprocessing, as clustering algorithms are known to perform worse in higher dimensions. A prominent example is principle component analysis (PCA) [28], in which variables of highest variance are created as linear combinations of the original ones. A certain number of new variables (the principle components) are then kept whereas the rest of them are discarded, resulting in a dataset containing most of the information but of lower dimensionality (see section 2.3.1).

- **Function approximation.** Many machine learning models, especially neural networks, are excellent function approximators. In some sense, all machine learning models approximate functions. For example, classification models try to find the function which maps data points $\mathbf{x}$ to their corresponding classes $f(\mathbf{x}) \in \{c_1, c_2, \ldots, c_N\}$. Regression models are then the generalization to continuous classes. In these cases we are mostly interested in the outputs of

the function for given data points. As we will see later when we discuss neural networks as representations for quantum wave functions, we are sometimes interested in the function in and of itself, because it might contain valuable information.

## 2.3 Machine learning algorithms

Perhaps the most important step in the machine learning workflow is the choice of model. This choice depends mainly on the situation, e.g. the type of task and data availability. Additional criteria for model/algorithm selection are (i) Interpretability/explainability: in order to obtain insight, or to increase model performance, it is beneficial to be able to follow along the process by which the model reaches its output. For example, when using a decision tree we can percolate a sample through the tree manually, meaning the processing steps can closely be followed. Contrarily, a neural network behaves more as a black box. (ii) Transferability: when we want the model to be more broadly applicable, e.g. to samples stemming from different distributions, then we need the model to be transferable [29]. A classic example is a neural network trained to classify dogs and cats in images, and extending its use to classify other animals. (iii) Many others: further criteria include the average training time of the model, the expected accuracy, the memory requirements, etc. Some of the most well-known machine learning algorithms are briefly discussed below.

### 2.3.1 Principle component analysis

Principle component analysis (PCA) is one of the oldest and most widely used dimensionality reduction techniques, and it is used in various disciplines. The idea behind PCA is to reduce the dimensionality of the data while preserving the statistical information [28]. This is done by finding new variables that are linear transformations of those in the original dataset, such that they maximize variance and are uncorrelated.

In practice, the $(n \times p)$ dataset $\mathbf{X}$ containing $n$ samples with $p$ variables is first normalized to have zero mean and unit variance for each variable (i.e. the columns). Subsequently, the covariance matrix $\mathbf{S}$ is calculated, which holds information about the correlations between variables in the dataset. Finding the new variables of highest variance then amounts to solving the eigenvalue problem $\mathbf{S}\mathbf{a} = \lambda\mathbf{a}$, where $\mathbf{a}$

is an eigenvector of the covariance matrix $\mathbf{S}$ and $\lambda$ is the corresponding eigenvalue. We are interested in the eigenvectors that have the largest eigenvalues, since the eigenvalues are the variances of the linear combinations defined by the eigenvectors $\mathrm{Var}(\mathbf{X}\mathbf{a}_i) = \lambda_i$. After ordering the eigenvectors, we choose the first $n$ to be the principle components, and discard the rest. Finally, we transform the data to the new coordinate system of the chosen principle components. In this way, we have reduced the number of variables, but kept most of the variance in the dataset.

### 2.3.2 Support vector machines

Support vector machines (SVM) are supervised classifiers that maximize the margin (i.e. the distance between the classes) in order to assure generalizability [30]. Assume we have a linearly separable dataset $\{\mathbf{x_i}, y_i\}, i = 0, 1, ..., N$, where the possible values of $y_i \in \{-1, 1\}$ represent the two classes. The decision boundary is a line if the vector $\mathbf{x} \in \mathbb{R}^d$ is two-dimensional $(d = 2)$. If we have data with three variables $(d = 3)$, the decision boundary is a plane.

For arbitrary dimensionality, we search for the separating hyperplane $\mathbf{w}\mathbf{x}+b = 0$ with maximal margin. In the linearly separable case we can write $y_i(\mathbf{w}\mathbf{x}_i + b) - 1 \geq 0, \forall i$. Now consider two auxiliary hyperplanes $H_1$ and $H_2$ defined as in Fig. 2.1, such that the margin is the distance between the two. Take a point $\mathbf{x}^-$ on $H_1$ and let $\mathbf{x}^+$ be the closest point on $H_2$. The margin can be calculated as $M = |\mathbf{x}^+ - \mathbf{x}^-|$, where $\mathbf{x}^+ - \mathbf{x}^- = \lambda\mathbf{w}$. By combining this with $\mathbf{w}\mathbf{x}^\pm + b = \pm 1$ and rewriting, we find $\lambda = \frac{2}{\mathbf{w}\mathbf{w}}$. We thus end up with the margin $M = |\lambda\mathbf{w}| = \frac{2\sqrt{\mathbf{w}\mathbf{w}}}{\mathbf{w}\mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}\mathbf{w}}}$. The problem can now be stated as a constrained optimization problem: maximize M (equivalent to minimize $\frac{\mathbf{w}\mathbf{w}}{2}$) with the constraint that all data points are classified correctly. This leads to a quadratic optimization problem with $N$ linear constraints, which can be solved using the method of Lagrange multipliers.

The data points that define the auxiliary hyperplanes $H_1$ and $H_2$ are termed the support vectors. Note that the support vectors form a compact representation of the model (the SVM model is completely defined by the support vectors). In order to handle non-separable cases, slack-variables $\xi_i$ that penalize errors are introduced. These variables are included in the optimization criterion, so that now the sum $\frac{\mathbf{w}\mathbf{w}}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$ is minimized. This can again be solved using Lagrange multipliers, with $N$ additional inequality constraints $\xi_i \geq 0, \forall i$.

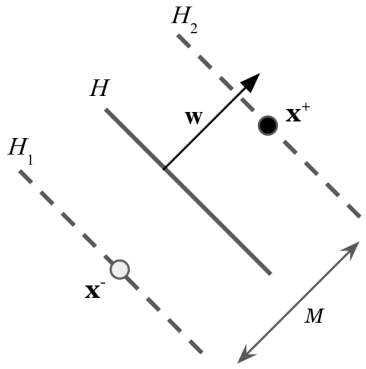



**Fig. 2.1.:** Graphical representation of the construction in SVMs. We have the separating hyperplane $H$ defined by $\mathbf{w}\mathbf{x} + b = 0$, and the auxiliary hyperplanes $H_1$ and $H_2$. The margin $M$ is the distance between the two points $\mathbf{x}^+$ and $\mathbf{x}^-$, which is used to define the optimization problem in the main text.

Non-linear problems are solved by transforming the data points $\mathbf{x_i}$ into a higher dimensional feature space $\mathbf{x_i} \rightarrow \mathbf{z}_i = \phi(\mathbf{x_i})$, with $\dim(\mathbf{z}_i) > \dim(\mathbf{x}_i)$. The data is linearly separable in this embedding, meaning the SVM algorithm can be applied. In practice, the transformation is not explicitly performed. Rather, the efficient kernel trick is applied, where a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)$ implicitly transforms the data [31]. This can be done since only the dot products enter in the optimization problem. The approach of implicitly transforming the data into a linearly separable feature space is generally applicable, and has been used to generalize other linear models (e.g. kernel PCA).

### 2.3.3 Neural networks

Artificial neural networks (ANNs) are the machine learning tool used in this thesis, therefore we introduce them in more detail. We call these models "neural networks" because of their resemblance to interconnected biological neurons in e.g. the human brain. Neural networks generally model a function, and the goal is then to find the parameters that yield the correct mapping. In the past years, ANNs have successfully been applied to various fields, thereby enabling innovative technologies ranging from speech recognition to self-driving cars. In the coming sections, we will discuss how to use them to represent quantum many-body wave functions. Now, we will start investigating the basic building blocks of these models, and gradually increase their complexity in a step-by-step fashion. We will discuss the initialization of neural networks, and how the parameters are updated using backpropagation.

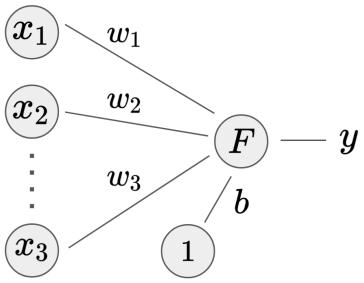



**Fig. 2.2.:** A depiction of the perceptron model, the fundamental building block of neural networks. It performs a linear combination of the inputs $x_i$ weighted by the weights $w_i$, and adds a bias $b$. The result is given as input to an activation function $F$, which gives us the output of the model $y$. The bias can be seen as an additional term in the linear combination: the additional virtual input is fixed to 1 and the corresponding weight is $b$.

**Perceptrons** The fundamental building block of neural networks is the so-called perceptron, which is a simplified single neuron model as shown in Fig. 2.2. Mathematically, its output $y$ is given by

$$y = F(\mathbf{w} \cdot \mathbf{x} + b), \qquad F(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where $\mathbf{x}$ is a real-valued input vector, $\mathbf{w}$ is a real-valued weight vector, $b$ is a bias, and $F$ is a (piecewise) *linear* activation function. The perceptron models a treshold function, as it is activated (returns 1) only if the argument $x$ of the activation function $F(x)$ is greater than zero. With this definition, it is clear that the perceptron is a binary classifier that distinguishes linearly separable data. Geometrically, the bias $b$ shifts the decision boundary $\sum_i^n w_i x_i + b = 0$, whereas $\mathbf{w}$ determines its slope. Otherwise, the weight vector $\mathbf{w}$ can be interpreted as a feature detector. When a given input feature $x_i$ is irrelevant for the output $y$, the corresponding weight $w_i$ will be zero. Conversely, a relatively high weight can be assigned to meaningful input features.

In a supervised setting with labels $d(\mathbf{x})$, optimizing the variational parameters $\mathcal{W} = \{\mathbf{w}, b\}$ is done iteratively using one of three rules. In these rules, $\eta$ denotes the learning rate and $\Delta w_i(t)$ is the update of parameter $w_i$ at step $t$ such that $w_i(t+1) = w_i(t) + \Delta w_i(t)$. The rules are

1. Hebbian rule: $\Delta w_i(t) = \eta\, y x_i$ ,
2. Perceptron rule: $\Delta w_i(t) = d(\mathbf{x}) x_i \quad \text{if } y \neq d(\mathbf{x}) \text{ else } 0$,
3. Delta rule: $\Delta w_i(t) = \eta(d(\mathbf{x}) - y) x_i$ .

The bias $b$ is handled as an additional weight $b = w_{n+1}$ connected to a constant (virtual) input $x_{n+1} = 1$, where $n$ is the real input dimension. The Hebbian rule states that the connection should be strengthened when units are activated simultaneously. The perceptron rule uses a sign activation function such that $d(\mathbf{x}) = \pm 1$ and only updates if $y \neq d(\mathbf{x})$. This rule has an important convergence property: *If the data is linearly separable, the perceptron learning rule will converge to a solution in a finite number of steps for any initial choice of weights* [32]. These kind of theorems are interesting from both a theoretical and an experimental standpoint, as they can guide further developments and practical usage of the model. Note that the premise of the theorem is that the data is linearly separable. If this is not the case, the update rule never leads to convergence. The Delta rule (also called the Widrow-Hoff rule) circumvents this issue by finding a local optimum in these cases. It can be derived using the gradient descent algorithm Eq. (2.2) on a least squares loss function $\mathcal{C} = \sum_i (d(\mathbf{x}_i) - y_i)^2$. These considerations show the importance of a good optimization algorithm.

More complex data distributions can be modeled by combining the decision boundaries of multiple output nodes, also called single layer perceptrons. Note that the individual boundaries themselves remain straight lines. A natural extension of the single layer perceptron is to include an extra "hidden" layer in addition to the input and output layers. The multi-layer perceptron (MLP), having at least three layers, is the result of sequentially stacking single layer perceptrons. Except for the input layer, the nodes in each layer have linear activation functions.[1] The usefulness of this model is however limited, since linear algebra shows that an MLP with any amount of layers can be reduced to a two-layer input-output model.

**Feedforward Neural Networks** The expressive power of neural networks can drastically be increased by generalizing the MLPs to include non-linear activation functions. This extension leads to a class of models known as feedforward neural networks (FFNN). As is illustrated in figure 2.3, the FFNN consists of an input layer, a given number of hidden layers, and the output layer. In its most basic form, each layer is fully connected to the next one, and there are no connections between nodes in the same layer. We call the network "deep" if it contains more than one hidden layer. Except for the nodes in the input layer, each node represents a perceptron with a non-linear activation function. The variational parameters $\mathcal{W}$ are given by the weights $w_{ij}^p$ and the biases $b_i^p$. In this notation, subscripts and superscripts correspond to node and hidden layer numbers, respectively.

[1] Actually, the term MLP is used ambiguously. In some works, the MLP has non-linear activation functions — here we reserve this feature for feedforward neural networks.

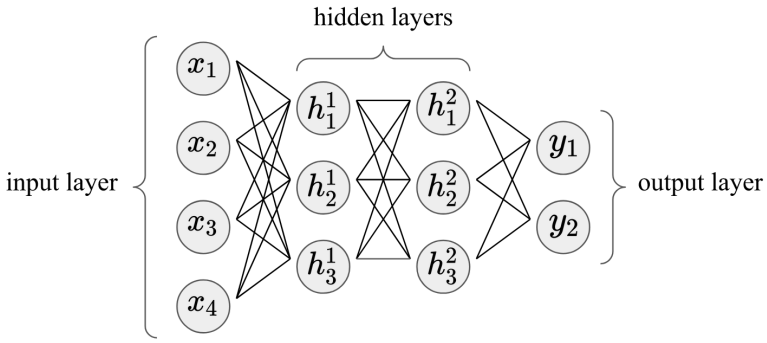



**Fig. 2.3.:** A depiction of a (fully connected) feedforward neural network consisting of 4 layers total: we have an input layer of 4 input nodes, 2 hidden layers of size 3, and the output layer that has 2 output nodes. The biases are not shown explicitly.

Feedforward denotes that information flows only in one direction, namely from input to output. The input nodes are connected to the nodes of the first hidden layer, whose outputs are then again the inputs of the nodes in the next layer. We can then write the various (intermediate) outputs as

$$\text{output nodes:} \qquad y_k = F_o\left(\sum_{j=1}^{N_h^p} h_j^p w_{jk}^p + b_k^p\right), \tag{2.4a}$$

$$\text{intermediate hidden nodes:} \qquad h_j^{q+1} = F_h\left(\sum_{l=1}^{N_h^q} h_l^q w_{lj}^q + b_j^q\right), \tag{2.4b}$$

$$\text{initial hidden nodes:} \qquad h_l^1 = F_h\left(\sum_{i=1}^{N_v} x_i w_{il}^1 + b_l^1\right), \tag{2.4c}$$

where we denoted the number of nodes in hidden layer $q \in \{1, \ldots, p\}$ by $N_h^q$, and the number of inputs by $N_v$. The advantage of non-linear activation functions is evident from the following.

**The Universal Approximation Theorem:**

Only one layer of hidden units suffices to approximate any function with finitely many discontinuities to arbitrary precision, provided the activation functions of the hidden units are non-linear [33].

Multiple hidden layers are however used in practice because these architectures are more efficient. Note that the output nodes do not require non-linear activation functions, and that it is not mandatory for all activation functions in the network to

be equal. Historically, the sigmoid function $F(x) = (1 + e^{-x})^{-1}$ is customarily taken as non-linear activation function of the hidden nodes.

Optimization is performed using a technique called (error) backpropagation [34]. It starts by defining a loss function and calculating its gradients to the variational parameters. Starting with the output nodes, the chain rule is repeatedly applied to Eqs. (2.4) in order to compute the contributions to the total loss. Typically, the resulting equations can not be solved in closed form, leading to the use of iterative methods such as gradient descent: samples are propagated through the network, the error is calculated, and the parameters are adjusted in the direction of decreasing error. This is repeated until a certain degree of convergence is reached.

In recent years, so-called convolutional neural networks (CNNs) have widely been applied, most often in the context of image recognition. CNNs are a restricted type of feedforward neural networks, where the weights $w_{ij}^p$ have specific properties. By restricting the values $w_{ij}^p$ to be non-zero only for certain nodes $i$, the weight vectors are said to act as feature extractors. For example, in case of an image classifier, we can have vectors $w_{ij}^\alpha$ and $w_{ij}^\beta$ which detect horizontal and vertical lines, respectively. Combining this information in a subsequent layer leads to the detection of angles, which in turn can be combined with other information to predict the content of the image.

**Restricted Boltzmann Machines** The restricted Boltzmann machine (RBM) [35] is a generative stochastic model that can learn a probability distribution over its set of inputs. The structure of an RBM is that of a bipartite graph, where the input variables $\mathbf{x}$ (also called visible units) are fully connected to the single layer of hidden variables $\boldsymbol{h}$ by a weight matrix $\mathbf{w}$. It is a subclass of the more general Boltzmann machines, where restricted means no intra-layer connections are allowed (see Fig. 2.4). With this architecture, an energy function can be defined as

$$E(\mathbf{x}, \boldsymbol{h}; \mathcal{W}) = \sum_{i=1}^{N_v} a_i x_i + \sum_{i=1}^{N_h} b_i h_i + \sum_{i=1}^{N_v} \sum_{j=1}^{N_h} w_{ij} x_i h_j, \tag{2.5}$$

which depends on the variational parameters $\mathcal{W} \equiv \{a_i, b_i, w_{ij}\}$. The numbers $N_v$ and $N_h$ denote the number of visible and hidden units, respectively. The factors $a_i$ ($b_i$) introduce a bias by independently acting on the corresponding visible (hidden) units. The hidden units capture correlations and together with the weight factors $w_{ij}$, the interaction terms of Eq. (2.5) can enhance or counteract the biases. Similar

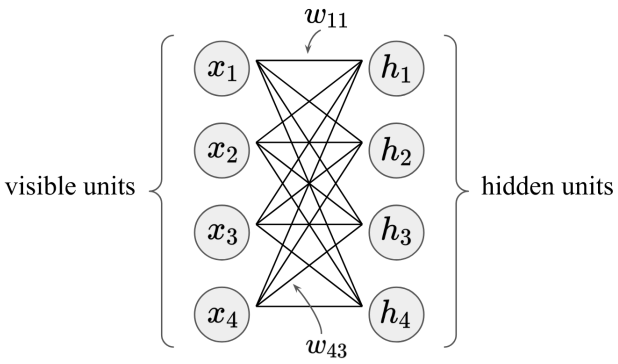



**Fig. 2.4.:** Graphical representation of a fully connected restricted Boltzmann machine. In this example, the number of inputs (or visible units) $N_v$ is equal to the number of hidden units $N_h$. The two layers are connected by weights $w_{ij}$. Biases are not explicitly shown.

to how we define Boltzmann distributions in statistical physics, we use the energy expression Eq. (2.5) to find a probability distribution at given $\mathcal{W}$

$$p(\mathbf{x}, \boldsymbol{h}; \mathcal{W}) = \frac{\exp(-E(\mathbf{x}, \boldsymbol{h}; \mathcal{W}))}{\sum_{\mathbf{x},\boldsymbol{h}} \exp(-E(\mathbf{x}, \boldsymbol{h}; \mathcal{W}))} = \frac{\exp(-E(\mathbf{x}, \boldsymbol{h}; \mathcal{W}))}{Z(\mathcal{W})}. \tag{2.6}$$

The partition function $Z(\mathcal{W})$ in the denominator runs over all possible configurations. We get the probability of the data $p(\mathbf{x}; \mathcal{W})$, i.e., the probability of a given configuration $\mathbf{x}$ of visible units, by summing over the hidden units

$$p(\mathbf{x}; \mathcal{W}) = \sum_{\{h^i\}} p(\mathbf{x}, \mathbf{h}; \mathcal{W}), \tag{2.7}$$

where $\{h^i\}$ is the set of all possible configurations of hidden units. With the RBM as a model for the probability Eq. (2.7), all sorts of interesting physics can be deduced. The particular case of using RBMs to represent quantum wave functions will be extensively discussed in section 3.1.

**Recurrent Neural Networks** Consider a discrete sample space with configurations denoted as $\boldsymbol{\sigma} \equiv (\sigma_1, \sigma_2, \ldots, \sigma_N)$. We have $N$ variables $\sigma_i$ which each can take values $\sigma_i \in \{0, 1, \ldots, D-1\}$, where $D$ is the input dimension representing the number of different obtainable values. We can cast the probability of a configuration $p(\boldsymbol{\sigma}) \equiv p(\sigma_1, \sigma_2, \ldots, \sigma_N)$ in the following form by using the product rule for probabilities

$$p(\boldsymbol{\sigma}) = p(\sigma_1)p(\sigma_2|\sigma_1)\ldots p(\sigma_N|\sigma_{N-1}, \cdots, \sigma_2, \sigma_1). \tag{2.8}$$

In this equation, $p(\sigma_i|\sigma_{i-1}, \ldots, \sigma_2, \sigma_1) \equiv p(\sigma_i|\sigma_{<i})$ is the conditional distribution of $\sigma_i$, given all $\sigma_j$ with $j < i$.

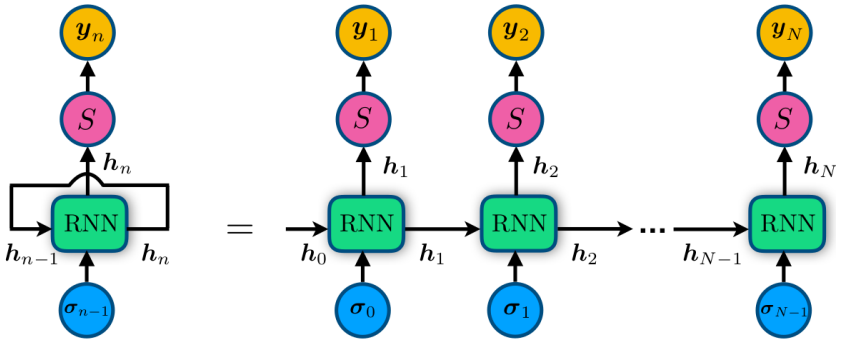



**Fig. 2.5.:** The recurrent neural network represented graphically. Left-hand side: compact version. Right-hand side: unrolled version. At each step $n$, a (one-hot encoded) input $\boldsymbol{\sigma}_{n-1}$ is fed to the recurrent cell, together with a hidden state vector $\boldsymbol{h}_{n-1}$. The cell computes a new hidden state vector $\boldsymbol{h}_n$, which passes a softmax layer $(S)$ to obtain conditional probabilities $\boldsymbol{y}_n$. Figure adapted from Ref. [36].

A recurrent neural network (RNN) [37] models the probability $p(\boldsymbol{\sigma})$ by determining the conditionals $p(\sigma_i|\sigma_{<i})$ of Eq. (2.8). The elementary building block of an RNN is called the recurrent cell. In its most basic form, the recurrent cell maps the direct sum (the concatenation) of an input hidden vector $\boldsymbol{h}_{i-1}$ of dimension $d_h$ (the number of memory or hidden units) and an input visible vector $\boldsymbol{\sigma}_{i-1}$ to an output hidden vector $\boldsymbol{h}_i$ of dimension $d_h$ by use of a non-linear activation function $F$, written as [36]

$$\boldsymbol{h}_i = F(W[\boldsymbol{h}_{i-1}; \boldsymbol{\sigma}_{i-1}] + \boldsymbol{b}). \tag{2.9}$$

We have as parameters the weight matrix $W \in \mathbb{R}^{d_h \times (d_h + D)}$, the bias vector $\boldsymbol{b} \in \mathbb{R}^{d_h}$, and initial states of the recursion $\boldsymbol{h}_0$ and $\boldsymbol{\sigma}_0$ which are fixed to constant values. The vector $\boldsymbol{\sigma}_i$ denotes the one-hot encoding of input $\sigma_i$. A graphical representation of the RNN is shown in Fig. 2.5. With this basic "vanilla" recurrent cell, long distance correlations are suppressed exponentially. Also, it is known to suffer from vanishing gradients, which leads to unstable optimizations. To accommodate for these shortcomings, an extension known as the gated recurrent unit (GRU) is often used [38], which processes the input configurations $\boldsymbol{\sigma}$ as

$$\boldsymbol{u}_i = \mathrm{sig}\,(W_u[\boldsymbol{h}_{i-1}; \boldsymbol{\sigma}_{i-1}] + \boldsymbol{b}_u), \tag{2.10}$$

$$\boldsymbol{r}_i = \mathrm{sig}\,(W_r[\boldsymbol{h}_{i-1}; \boldsymbol{\sigma}_{i-1}] + \boldsymbol{b}_r), \tag{2.11}$$

$$\boldsymbol{h}'_i = \tanh\,(W_c[\boldsymbol{r}_n \odot \boldsymbol{h}_{i-1}; \boldsymbol{\sigma}_{i-1}] + \boldsymbol{b}_c), \tag{2.12}$$

$$\boldsymbol{h}_i = (1 - \boldsymbol{u}_i) \odot \boldsymbol{h}_{i-1} + \boldsymbol{u}_i \odot \boldsymbol{h}'_i, \tag{2.13}$$

where $\mathrm{sig}$ and $\tanh$ are the sigmoid and hyperbolic tangent activation functions, respectively. The symbol $\odot$ denotes the point-wise (Hadamard) product [39]. The newly generated hidden state $\boldsymbol{h}_i$ is constructed as an interpolation between the

previous one $\boldsymbol{h}_{i-1}$ and a candidate hidden state $\boldsymbol{h}'_i$. The update gate $\boldsymbol{u}_i$ controls the extent of the update, and the reset gate $\boldsymbol{r}_i$ (potentially) cancels components of $\boldsymbol{h}_{i-1}$ (often referred to as the GRU 'forgetting' the information encoded in the previous hidden state). Regardless of which recurrent cell is used, a hidden state vector $\boldsymbol{h}$ is obtained. From this, the conditionals can be computed as

$$p(\sigma_i|\sigma_{i-1},\ldots,\sigma_1) = \boldsymbol{y}_i \cdot \boldsymbol{\sigma}_i, \quad \text{with} \quad \boldsymbol{y}_i \equiv \mathcal{S}(U\boldsymbol{h}_i + \boldsymbol{c}), \tag{2.14}$$

where $U \in \mathbb{R}^{D\times d_h}$ and $\boldsymbol{c} \in \mathbb{R}^D$ are weights and biases of a softmax layer, and $\mathcal{S}$ is the softmax activation function

$$\mathcal{S}(x_n) = \frac{\exp(x_n)}{\sum_i \exp(x_i)}. \tag{2.15}$$

The full probability $p(\boldsymbol{\sigma})$ is obtained by sequentially computing the conditionals of Eq. (2.14) and multiplying, this gives

$$p(\boldsymbol{\sigma}) = \prod_{i=1}^{N} \boldsymbol{y}_i \cdot \boldsymbol{\sigma}_i, \quad \text{where} \quad \|\boldsymbol{y}_i\|_1 = \|p(\boldsymbol{\sigma})\|_1 = 1. \tag{2.16}$$

The norms sum over positive, real values and thus represent probability distributions. For the vanilla RNN, the variational parameters are $\mathcal{W} = \{W, U, \boldsymbol{b}, \boldsymbol{c}\}$. The RNN is compact and independent of system size $N$, since the variational parameters are shared among the $N$ cells. Deep architectures are constructed by stacking RNN cells (with intermediate activation functions), the depth being denoted by the number of layers $n_l$.

**Initialization of variational parameters** Adequate initialization of the weights and biases of the network can be of great importance. Consider a deep neural network, where many consecutive layers perform matrix multiplications and have non-linear activation functions. If the variational parameters are not properly initialized, the (mean of the) outputs of the nodes of layers deep in the network are expected to vanish or to become very large. This means that the next layer gets inputs close to 0 or inputs that are very large, possibly interpreted by the computer as invalid (not a number, NaN). Consequently, the gradients needed in the optimization of the network become unstable. This leads to the network failing to converge, or possibly even errors to occur due to the appearance of NaNs.

In order to have a stable optimization and avoid extreme gradients, proper initialization of the parameters is required. When using activation functions that are symmetric around 0 and have outputs in $[-1, 1]$, we want each layer to have outputs with a mean of 0 and a standard deviation around 1, on average. In this way, information gets propagated throughout the entire network in a stable fashion, and gradients are expected to behave as desired.

For many years, the commonly used "heuristic" for initializing parameters was to generate random values from the uniform distribution $[-1, 1]$ and subsequently scaling them by $1/\sqrt{n}$, where $n$ is the size of the previous layer. However, it has been shown in [40] that this approach leads to diminishing gradients as we go deeper into the network. The authors proposed a new initialization scheme: the values are now picked from the uniform distribution ranging between $\left[-\sqrt{\frac{6}{n_i+n_{i+1}}}, +\sqrt{\frac{6}{n_i+n_{i+1}}}\right]$, where $n_i$ are the number of input connections to the layer (fan-in) and $n_{i+1}$ are the number of output connections (fan-out). Biases are initialized to 0. With this initialization, the authors have shown that the gradients remain approximately constant among different layers, and faster convergence is acquired. In close resemblance, we can draw parameter values from a Gaussian distribution with 0 mean and standard deviation $\sqrt{\frac{2}{n_i+n_{i+1}}}$. These initialization methods are commonly referred to as Xavier or Glorot initialization.

Xavier initialization is optimal when using symmetric activation functions with a limited range, e.g. the logistic sigmoid $F(x) = (1+e^{-x})^{-1}$ and the softmax functions Eq. (2.15). Inherently, these functions call for vanishing gradients. Take for example the sigmoid function, whose output does not change considerably by changing $x$ when $x$ is very large (or small), and thus the gradients are expected to vanish. To remedy this, activation functions with a broader range are customarily used, e.g. the rectified linear unit (ReLU) $G(s) = \max(0, s)$. For this activation function, the optimal standard deviation was found to be $2/n^L$ with $n^L$ the fan-in of layer $L$ [41]. As with Xavier initialization, bias vectors are initialized to 0. This is referred to as He or Kaiming initialization. The derivations can be found in the references.

# 3 Artificial neural networks in many-body physics

In this chapter, we discuss how restricted Boltzmann machines (RBM) and recurrent neural networks (RNN) can be used to represent many-body quantum wave functions. In sections 3.1.1−3.1.3 we introduce the idea of a variational ansatz and explain how we can use it to calculate expectation values. Next in section 3.1.4, we explain the optimization procedure in order to adequately represent the target quantum state, customarily the ground state. Some additional notes on using the RNN as ansatz are given in section 3.2. We subsequently show in section 3.3 how to adapt the neural networks such that they satisfy SU(2) symmetry. In chapter 4, we compare the performance of the two types of neural networks (RBMs and RNNs), with and without the implementation of SU(2) symmetry, by applying the strategy to established lattice systems.

## 3.1 Restricted Boltzmann machines for modeling quantum systems

### 3.1.1 The RBM as variational ansatz

The wave function $|\Psi\rangle$ of a quantum mechanical spin-$1/2$ system can be written as a superposition in some basis, e.g. the $\sigma^z$-basis, as follows

$$|\Psi\rangle \equiv \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle = \sum_{\{\sigma_i^z\}} \psi(\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z) \, |\sigma_1^z \sigma_2^z \ldots \sigma_{N_v}^z\rangle = \sum_{n=1}^{2^{N_v}} \psi(\boldsymbol{\sigma}_n) \, |\boldsymbol{\sigma}_n\rangle \, , \tag{3.1}$$

where $N_v$ is the number of lattice sites. A configuration $\boldsymbol{\sigma}$ is expressed in terms of the eigenvalues $\sigma_i^z = \pm 1$ of the single spin $\hat{\sigma}^z$-eigenfunctions at sites $i$. The wave function $|\Psi\rangle$ is fully determined by the complex amplitudes $\psi(\boldsymbol{\sigma}_n)$ associated to the $2^{N_v}$ possible configurations. Computing all amplitudes is typically intractable, since the number of terms in the expansion of Eq. (3.1) scales exponentially with $N_v$.

Therefore, a variational ansatz is often used, which is a parameterization of the wave function

$$|\Psi_{\mathcal{W}}\rangle = \sum_{\boldsymbol{\sigma}} \psi_{\mathcal{W}}(\boldsymbol{\sigma}) \, |\boldsymbol{\sigma}\rangle \, . \tag{3.2}$$

The complex amplitudes $\psi_{\mathcal{W}}(\boldsymbol{\sigma})$ and the wave function $|\Psi_{\mathcal{W}}\rangle$ are fully determined by the complex parameters $\mathcal{W} \in \mathbb{C}^{n_{\text{par}}}$. When the number of parameters $n_{\text{par}}$ is much smaller than the size of the Hilbert space $\sim 2^{N_v}$, we have an efficient representation. The goal is then to find those parameters $\mathcal{W}$ that best approximate the exact wave function Eq. (3.1).

For the spin states in Eq. (3.1), we can rewrite our expression of the energy given by an RBM Eq. (2.5): to suit our problem at hand, the visible units are written in terms of spin eigenvalues $\sigma_i^z = \pm 1$. The hidden units encode the correlations between the visible spins and adopt the same binary values. The expressivity of the model is determined by the number of hidden units $N_h$, usually characterized by the ratio $\alpha = N_h/N_v$. The energy of Eq. (2.5) is used to define the wave function by [10]

$$\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \langle \boldsymbol{\sigma} | \Psi_{\mathcal{W}} \rangle = \sum_{\mathbf{h}} e^{-E(\boldsymbol{\sigma}, \mathbf{h}; \mathcal{W})}, \tag{3.3}$$

where the sum extends over all possible configurations of the hidden units $\{h_i\}$. The partition function of Eq. (2.6) is absorbed in the normalization of the wave function.

Explicit summation over the hidden units results in a popular way of writing the wave function of an RBM, namely

$$\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = e^{\sum_i a_i \sigma_i^z} \times \prod_{i=1}^{N_h} F_i(\boldsymbol{\sigma}), \qquad \text{where} \quad F_i(\boldsymbol{\sigma}) = 2\cosh\left[ b_i + \sum_j^{N_v} w_{ij} \sigma_j^z \right]. \tag{3.4}$$

The advantage of this notation stems from the interpretation of the number of hidden units $N_h$, which plays a role analogous to the bond dimension of matrix product states [10]. A neural network that is used as variational ansatz for quantum wave functions as in Eq. (3.2) is termed a “neural network quantum state” (NQS).

### 3.1.2 Calculating expectation values

Even upon using an efficient variational wave function $|\Psi_{\mathcal{W}}\rangle$ in Eq. (3.2), exact calculation of expectation values is intractable for large systems, since it involves summing over the entire Hilbert space. However, numerical approximations can be obtained.

We start by inserting resolutions of the identity in terms of the basis elements $\hat{I} = \sum_{\boldsymbol{\sigma}} |\boldsymbol{\sigma}\rangle \langle\boldsymbol{\sigma}|$ in the definition of the expectation value of some observable $\hat{\mathrm{O}}$, and rewriting

$$\langle\hat{\mathrm{O}}\rangle \equiv \frac{\langle\Psi_{\mathcal{W}}|\hat{\mathrm{O}}|\Psi_{\mathcal{W}}\rangle}{\langle\Psi_{\mathcal{W}}|\Psi_{\mathcal{W}}\rangle} = \frac{\sum_{\boldsymbol{\sigma}} \langle\Psi_{\mathcal{W}}|\boldsymbol{\sigma}\rangle \langle\boldsymbol{\sigma}|\hat{O}|\Psi_{\mathcal{W}}\rangle}{\sum_{\boldsymbol{\sigma}} \langle\Psi_{\mathcal{W}}|\boldsymbol{\sigma}\rangle \langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle} = \frac{\sum_{\boldsymbol{\sigma}} O_L(\boldsymbol{\sigma})|\psi_{\mathcal{W}}(\boldsymbol{\sigma})|^2}{\sum_{\boldsymbol{\sigma}} |\psi_{\mathcal{W}}(\boldsymbol{\sigma})|^2}, \tag{3.5}$$

where we defined the local estimator $O_L(\boldsymbol{\sigma})$ as

$$O_L(\boldsymbol{\sigma}) = \frac{\langle\boldsymbol{\sigma}|\hat{O}|\Psi_{\mathcal{W}}\rangle}{\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle}. \tag{3.6}$$

In this way the expectation value $\langle\hat{\mathrm{O}}\rangle$ is recast as the average $\sum_{\boldsymbol{\sigma}} p(\boldsymbol{\sigma}) O_L(\boldsymbol{\sigma})$ of a random variable $O_L(\boldsymbol{\sigma})$ over the probability distribution $p(\boldsymbol{\sigma})$ given by

$$p(\boldsymbol{\sigma}) = \frac{|\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle|^2}{\sum_{\boldsymbol{\sigma}} |\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle|^2}. \tag{3.7}$$

In practice, $N_s$ configurations $\{\boldsymbol{\sigma}\}$ are sampled in order to approximate the probability distribution $p(\boldsymbol{\sigma})$, allowing us to take the mean of the corresponding local estimators

$$\langle\hat{\mathrm{O}}\rangle = \langle \mathrm{O}_L \rangle_{p(\boldsymbol{\sigma})} = \frac{1}{N_s} \sum_{n=1}^{N_s} \mathrm{O}_L(\boldsymbol{\sigma}_n) \pm \sqrt{\frac{\mathrm{Var}(\mathrm{O}_L)}{N_s}}. \tag{3.8}$$

Often we want to calculate the expectation value of the Hamiltonian $\hat{H}$, in which case the local estimator is called the local energy $e_L(\boldsymbol{\sigma})$ and we get

$$E(\mathcal{W}) \approx \frac{1}{N_s} \sum_{n}^{N_s} e_L(\boldsymbol{\sigma}_n). \tag{3.9}$$

This strategy requires efficient calculation of the local energies $e_L(\boldsymbol{\sigma})$, which is possible if $\hat{H}$ is sparse (which means that any given row of the matrix representation of $\hat{H}$ contains few non-zero terms). This is the case for Hamiltonians that only involve local few-body interactions. Furthermore, the variational representation must be able to efficiently generate samples according to $p(\boldsymbol{\sigma})$.

### 3.1.3 Sampling with Markov chains

For energy-based models such as the RBM, the output amplitudes $\psi(\boldsymbol{\sigma})$ are not properly normalized. This means that direct sampling according to Eq. (3.7) is unfeasible due to the sum over the entire Hilbert space in the denominator. In order to generate samples for estimating expectation values as in Eq. (3.8), we construct a Markov chain. We perform a Markov Chain Monte Carlo (MCMC) random walk in the configuration space of spin configurations $\boldsymbol{\sigma}$, and utilize the Metropolis-Hastings algorithm [42]. The algorithm looks as follows.

Step 0. Start with an initial configuration $\boldsymbol{\sigma}_0$.

Step 1. Propose a new spin configuration $\boldsymbol{\sigma'}$.

Step 2. Calculate the acceptance ratio $A(\boldsymbol{\sigma'}|\boldsymbol{\sigma}_{i-1}) = \min\left[1, \frac{p(\boldsymbol{\sigma'})}{p(\boldsymbol{\sigma}_{i-1})}\right]$.

Step 3. Draw a uniform random number $r$ between 0 and 1, and accept the proposed state, i.e., set $\boldsymbol{\sigma}_i = \boldsymbol{\sigma'}$, if $r < A(\boldsymbol{\sigma'}|\boldsymbol{\sigma}_{i-1})$. Otherwise, reject the state $\boldsymbol{\sigma'}$ if $r \geq A(\boldsymbol{\sigma'}|\boldsymbol{\sigma}_{i-1})$, meaning set $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_{i-1}$.

Repeat steps 1-3 until $N_s$ configurations are obtained.

We take the final configuration of a thermalization run (which starts with a random initial configuration) as the initial configuration $\boldsymbol{\sigma}_0$ of the production run. The new spin configuration $\boldsymbol{\sigma'}$ in step 1 is obtained by transforming the previous configuration $\boldsymbol{\sigma}_{i-1}$ according to some rule, e.g. by flipping a spin. The result of the algorithm is a chain of states $\boldsymbol{\sigma}_0 \to \boldsymbol{\sigma}_1 \to \boldsymbol{\sigma}_2 \to \cdots \to \boldsymbol{\sigma}_{N_s}$ that are distributed according to $p(\boldsymbol{\sigma})$. This chain can be used to evaluate expectation values as in Eq. (3.8). Note that the probability $p(\boldsymbol{\sigma})$ only enters as a fraction (step 2), which solves the denominator problem. At each iteration $i$, we only need the amplitudes of configurations $\boldsymbol{\sigma}_{i-1}$ and $\boldsymbol{\sigma}'$, and these are given by the RBM (Eq. (3.3)). We also note that the configurations in the Markov chain are correlated, hence statistical estimates need to be corrected for potential biases [43].

### 3.1.4 Optimizing the RBM

**Variational Monte Carlo** In this context, optimizing the RBM means finding those parameters $\mathcal{W}$ that correspond to the best approximation to the ground state $|\Psi_{\mathcal{W}}\rangle \approx |\Psi_{gs}\rangle$. With the tools introduced above, we can iteratively improve the representation using variational Monte Carlo (VMC) techniques [7, 10]. We start

with the variational principle, which says that the energy expectation value $E(\mathcal{W})$ of any trial wave function $|\Psi_{\mathcal{W}}\rangle$ will be higher than or equal to the exact ground-state energy $E_{gs}$. We can write this as

$$E(\mathcal{W}) = \langle \hat{H} \rangle = \frac{\langle \Psi_{\mathcal{W}} | \hat{H} | \Psi_{\mathcal{W}} \rangle}{\langle \Psi_{\mathcal{W}} | \Psi_{\mathcal{W}} \rangle} \geq E_{gs}, \tag{3.10}$$

which implies that if $E(\mathcal{W}) = E_{gs}$ then $|\Psi_{\mathcal{W}}\rangle = |\Psi_{gs}\rangle$. In order to find the best possible approximation to the exact ground state, we minimize the energy of the wave function ansatz by optimizing the variational parameters. This amounts to solving the objective function

$$\min_{\mathcal{W}} \left[ E(\mathcal{W}) \right]. \tag{3.11}$$

The variational principle Eq. (3.11) forms the basis for many methods that solve the quantum many-body problem, such as the Hartree-Fock (HF) [44] and density matrix renormalization group (DMRG) methods [45]. In our case of neural network quantum states, the minimum is found using iterative methods such as gradient descent or any of its variants. Gradient descent, in its most basic form, defines an update rule for the parameters

$$\mathcal{W}^{t+1} = \mathcal{W}^{t} - \eta \cdot \nabla_{\mathcal{W}} E(\mathcal{W}) \Big|_{\mathcal{W}=\mathcal{W}^{t}}, \tag{3.12}$$

with $\eta$ the learning rate and $t$ the optimization step. The gradients $\nabla_{\mathcal{W}} E(\mathcal{W})$ quantify the change in energy induced by varying the variational parameters. For a selected parameter $w$ (e.g. for the RBM $w \in \{a_i, b_i, w_{ij}\}$), the gradient of the energy to this parameter $g_w$ is calculated as

$$\begin{aligned} \nabla_w E(\mathcal{W}) = g_w &= \frac{\partial}{\partial w} \left( \frac{\langle \Psi_{\mathcal{W}} | \hat{H} | \Psi_{\mathcal{W}} \rangle}{\langle \Psi_{\mathcal{W}} | \Psi_{\mathcal{W}} \rangle} \right) \\ &= \sum_{\boldsymbol{\sigma}} \left[ \frac{\langle \boldsymbol{\sigma} | \frac{\partial \psi^*_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w} \hat{H} | \Psi_{\mathcal{W}} \rangle + \langle \Psi_{\mathcal{W}} | \frac{\partial \psi_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w} \hat{H} | \boldsymbol{\sigma} \rangle}{\langle \Psi_{\mathcal{W}} | \Psi_{\mathcal{W}} \rangle} \right] \\ &\quad - \sum_{\boldsymbol{\sigma}} \left[ \frac{\langle \Psi_{\mathcal{W}} | \hat{H} | \Psi_{\mathcal{W}} \rangle \left( \langle \boldsymbol{\sigma} | \frac{\partial \psi^*_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w} | \Psi_{\mathcal{W}} \rangle + \langle \Psi_{\mathcal{W}} | \frac{\partial \psi_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w} | \boldsymbol{\sigma} \rangle \right)}{\langle \Psi_{\mathcal{W}} | \Psi_{\mathcal{W}} \rangle^2} \right], \end{aligned} \tag{3.13}$$

where we again used the expansion of the wave function in the $\sigma^z$-basis. By introducing $\mathcal{O}_w(\boldsymbol{\sigma}) \equiv \frac{1}{\psi_{\mathcal{W}}(\boldsymbol{\sigma})} \frac{\partial \psi_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w}$, we can rewrite Eq. (3.13) as

$$g_w = 2 \, \mathrm{Re}(\langle \hat{H} \mathcal{O}^*_w \rangle - \langle \hat{H} \rangle \langle \mathcal{O}^*_w \rangle), \tag{3.14}$$

where expectation values are with respect to the wave function $|\Psi_{\mathcal{W}}\rangle$ ( Eq. (3.5)).

**Adaptations to gradient descent** The update rule of gradient descent Eq. (3.12) has known limitations. First, the rule does not necessarily find the global minimum of the energy landscape, and is bound to get stuck in local minima. This can partially be circumvented by introducing noise into the optimization, which enables the model to jump to new and potentially better local minima. A popular way of doing this is by using few samples to estimate the gradient of Eq. (3.12), thereby we also decrease the computational burden. In a supervised setting, this corresponds to only using part of the training set at each update step. This stochastic approximation is known as the stochastic gradient descent (SGD) method [46]. Secondly, the learning rate $\eta$ is fixed, which means that it has to be carefully chosen (see section 2.1.2). Learning rate schedules adjust the learning rate during training, by e.g. annealing, i.e., reducing the learning rate according to some schedule. This allows us to start with a large learning rate, such that the model can explore a broad region of parameter space. In order to properly converge to a minimum, the learning rate is decreased at each iteration. Finally, each parameter has the same learning rate. It is however sensible to stop adjusting parameters (or at least decrease the rate at which we adjust them) when they have already been extensively fine-tuned. Conversely, if a feature is rarely present in the samples that are used to estimate the gradients, we may want to adjust the corresponding parameters considerably whenever the feature does occur. Also, trenches or valleys in the energy landscape might cause the optimizer to oscillate. This hinders the model to descend into the valley along the dimension that actually leads to the minimum.

In order to facilitate the optimization and account for the shortcomings of standard gradient descent, the adaptive moment estimation (Adam) method can be used. Adam is said to implement momentum, which means that the updates of previous steps are taken into account when calculating new updates [47]. The Adam update rule for parameter $w$ at step $t$ is given by

$$w^{t+1} = w^t - \frac{\eta}{\sqrt{\hat{v}_w^t} + \epsilon} \hat{m}_w^t, \tag{3.15}$$

where $\epsilon$ is a small value ($\sim 10^{-8}$) to prevent division by zero, and $\eta$ determines the overall scale of the updates. The first moment estimate $\hat{m}_w^t$ and the second moment estimate $\hat{v}_w^t$ are bias-corrected versions of the unbiased estimators

$$\hat{m}_w^t = \frac{m_w^t}{1 - \beta_1^t}, \tag{3.16}$$

$$\hat{v}_w^t = \frac{v_w^t}{1 - \beta_2^t}. \tag{3.17}$$

The correction is done because $m_w$ and $v_w$ are initialised as zeros, making them biased towards zero in the initial steps. In this equation $\beta_1, \beta_2 \in [0, 1)$ are exponential decay rates that dictate how much the previous moments are taken into account in the new estimators. Note that $\beta^t$ denotes $\beta$ to the power $t$, whereas otherwise the superscripts $t$ indicate the step (and no mathematical operation). We will use Adam with the values proposed by the authors of Ref. [47], namely $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The estimators, which are the decaying averages of past (squared) gradients, are calculated as

$$m_w^t = \beta_1 m_w^{t-1} + (1 - \beta_1) g_w^t, \tag{3.18}$$

$$v_w^t = \beta_2 v_w^{t-1} + (1 - \beta_2) g_w^{t\,2}. \tag{3.19}$$

The values $m_w^t$ and $v_w^t$ are the estimates of the mean (first moment) and variance (second moment) of the gradients, which explains the name of the method.

Instead of using the $\ell_2$ norm in the $v_w^t$ factor, a general $\ell_p$ norm could be used. Norms for large values of p typically become numerically unstable, but $\ell_\infty$ generally exhibits stable behaviour. The authors parameterize $\beta_2$ as $\beta_2^p$, such that

$$u_w^t = \beta_2^\infty v_w^{t-1} + (1 - \beta_2^\infty) |g_w^t|^\infty, \tag{3.20}$$

$$= \max(\beta_2 \cdot v_w^{t-1}, |g_w^t|), \tag{3.21}$$

where we used $u_w^t$ to denote the infinity norm constrained $v_w^t$. Plugging this into the Adam update rule Eq. (3.15) and removing the now redundant $\epsilon$, we get

$$w^{t+1} = w^t - \frac{\eta}{u_w^t} \hat{m}_w^t. \tag{3.22}$$

The above equation is known as the Adamax update rule, and we will be using it with the same default values $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

**Stochastic reconfiguration** The stochastic reconfiguration (SR) scheme takes into account the change of the wave function when doing an update [48]. Therefore, it generally outperforms gradient descent algorithms that perform updates in the direction of steepest descent of the energy landscape [49]. The update rule for a variational parameter $w_k$ looks similar to that of gradient descent, namely

$$w_k' = w_k + \eta\, \delta w_k, \tag{3.23}$$

with learning rate $\eta > 0$ small enough to guarantee convergence. The difference with the gradient descent rule Eq. (3.12) is that instead of taking the gradient of the energy $\frac{\partial E}{\partial w}$ we now have another measure

$$\delta w_k = \sum_{k'} s^{-1}_{k,k'} f_{k'} \ . \tag{3.24}$$

Here, the generalized forces enter as $f_k = \frac{\partial E}{\partial w_k}$, meaning that if $s$ were the identity matrix we would recover the standard gradient descent rule. Instead, to account for the dependence between variational parameters and thereby accelerate the convergence, we use the positive definite matrix $s$ defined by

$$s_{k,k'} = \langle \mathcal{O}_k \mathcal{O}_{k'} \rangle - \langle \mathcal{O}_k \rangle \langle \mathcal{O}_{k'} \rangle \,, \qquad \text{with} \quad \mathcal{O}_w(\boldsymbol{\sigma}) \equiv \frac{1}{\psi_{\mathcal{W}}(\boldsymbol{\sigma})} \frac{\partial \psi_{\mathcal{W}}(\boldsymbol{\sigma})}{\partial w}. \tag{3.25}$$

Expectation values are again over the variational wave function and can be calculated according to Eq. (3.8). The matrix $s$ remains positive definite when using a finite number of samples, but the lowest eigenvalues and corresponding eigenvectors can be very sensitive to the statistical noise. Therefore, the matrix is regularized by the modification of its diagonal elements

$$s_{k,k} = s_{k,k}(1 + \epsilon), \tag{3.26}$$

where $\epsilon$ is a small $(\sim 0.1 - 0.001)$ regularizing term.

The SR scheme can actually be derived from linear approximations to the imaginary time evolution operator applied to normalized variational wave functions [50]. This shows that the fundamental difference between SR and steepest descent is the definition of the distance between parameters $w'$ and $w$ (which should be small to ensure stability). The Cartesian metric $\Delta_{\mathcal{W}} = \sum_k |w'_k - w_k|^2$ is namely replaced in SR with the physical Hilbert space metric of the wave function $\Delta_{\mathcal{W}} = \sum_{i,j} \bar{s}_{i,j}(w'_i - w_i)(w'_j - w_j)$, where $\bar{s}_{j,k} = s_{j,k} - s_{j,0}s_{0,k}$. The latter distance is just the square distance between the two normalized wave functions corresponding to the sets of parameters $\{w'\}$ and $\{w\}$ [51]. It can perfectly happen that a small change of the variational parameters leads to a large change of the wave function. The SR method is advantageous because it takes this effect into account.

**Zero variance property** If the variational state $|\Psi_{\mathcal{W}}\rangle$ coincides with an exact eigenstate of the Hamiltonian $\hat{H}$ such that $\hat{H}\,|\Psi_{\mathcal{W}}\rangle = E(\mathcal{W})\,|\Psi_{\mathcal{W}}\rangle$, then the local energy $e_L(\boldsymbol{\sigma})$ is constant [51]:

$$e_L(\boldsymbol{\sigma}) = \frac{\langle\boldsymbol{\sigma}|\hat{H}|\Psi_{\mathcal{W}}\rangle}{\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle} = E(\mathcal{W})\frac{\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle}{\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle} = E(\mathcal{W}). \tag{3.27}$$

This means that the local energy $e_L(\boldsymbol{\sigma})$, which enters as the random variable in Eq. (3.5), is independent of the configuration $|\boldsymbol{\sigma}\rangle$. We can therefore conclude that its variance is zero and that its mean coincides with the exact eigenvalue. In general, the closer the variational state $|\Psi_{\mathcal{W}}\rangle$ is to an exact eigenstate, the smaller the variance of $e_L(\boldsymbol{\sigma})$ becomes. Indeed, the average square of the local energy $\langle e_L^2(\boldsymbol{\sigma})\rangle$ corresponds to the exact quantum average of the Hamiltonian squared

$$\frac{\langle\Psi_{\mathcal{W}}|\hat{H}^2|\Psi_{\mathcal{W}}\rangle}{\langle\Psi_{\mathcal{W}}|\Psi_{\mathcal{W}}\rangle} = \frac{\sum_{\boldsymbol{\sigma}}\langle\Psi_{\mathcal{W}}|\hat{H}|\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}|\hat{H}|\Psi_{\mathcal{W}}\rangle}{\sum_{\boldsymbol{\sigma}}\langle\Psi_{\mathcal{W}}|\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle} = \frac{\sum_{\boldsymbol{\sigma}} e_L^2(\boldsymbol{\sigma})|\psi_{\mathcal{W}}(\boldsymbol{\sigma})|^2}{\sum_{\boldsymbol{\sigma}}|\psi_{\mathcal{W}}(\boldsymbol{\sigma})|^2} = \langle e_L^2\rangle\,. \tag{3.28}$$

Therefore we can write

$$\mathrm{Var}(\hat{H}) = \frac{\Psi_{\mathcal{W}}|(\hat{H}-E)^2|\Psi_{\mathcal{W}}}{\langle\Psi_{\mathcal{W}}|\Psi_{\mathcal{W}}\rangle} = \mathrm{Var}(e_L). \tag{3.29}$$

Thus, the closer the variational state is to the exact eigenstate, the smaller the variance. This reduces the statistical fluctuations, and allows us to use the variance as a convergence measure. Namely, instead of minimizing the energy, we can minimize the variance. This is helpful whenever the exact ground-state energy is unknown, since the variational principle on its own does not allow us to judge how accurate the representation is. On the contrary, the smallest possible value for the variance is known (it is zero), therefore the variance is a convenient accuracy measure. Note, however, that the variance goes to zero whenever the variational state accurately represents an arbitrary eigenstate of the Hamiltonian, and not only when it accurately represents the ground state.

## 3.2 Recurrent neural networks for modeling quantum systems

### 3.2.1 The RNN as variational ansatz

We have introduced the RNN in section 2.3, and showed how it can be used to approximate classical (i.e. having real and positive valued amplitudes) probability distributions $p(\boldsymbol{\sigma})$. A class of so-called stoquastic many-body Hamiltonians has ground states $|\Psi\rangle$ with real and positive amplitudes in the standard product spin basis. Those ground states can be readily modeled by the RNN we introduced earlier, as they have representations in terms of probability distributions. However, a general quantum wave function $|\Psi\rangle \equiv \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma}) |\boldsymbol{\sigma}\rangle$ has complex amplitudes $\psi(\boldsymbol{\sigma})$. In order to generalize to the complex case, we split the wave function in an amplitude $p(\boldsymbol{\sigma})$ and phase $\phi(\boldsymbol{\sigma})$, as [36]

$$|\Psi\rangle = \sum_{\boldsymbol{\sigma}} \exp(i\phi(\boldsymbol{\sigma}) \sqrt{p(\boldsymbol{\sigma})} |\boldsymbol{\sigma}\rangle . \tag{3.30}$$

Recall that the elementary building block of the RNN, the recurrent cell, generates a hidden vector state $\boldsymbol{h}_i$ for each spin site $\sigma_i$. From this, the conditional probabilities of the local configurations are (iteratively) calculated as

$$p(\sigma_i|\sigma_{i-1}, \ldots, \sigma_1) = \boldsymbol{y}_i^{(1)} \cdot \boldsymbol{\sigma}_i, \quad \text{with } \boldsymbol{y}_i^{(1)} = \mathcal{S}\left(U^{(1)}\boldsymbol{h}_i + \boldsymbol{c}^{(1)}\right), \tag{3.31}$$

where $\mathcal{S}$ is the softmax activation function Eq. (2.15). The probability of the total configuration is then obtained by multiplying the conditionals, $p(\boldsymbol{\sigma}) = \prod_{i=1}^{N_v} p(\sigma_i)$.[1] Similarly, we pass the hidden vector state $\boldsymbol{h}_i$ through a softsign layer

$$\boldsymbol{y}_i^{(2)} = \pi \,\text{softsign}\left(U^{(2)}\boldsymbol{h}_i + \boldsymbol{c}^{(2)}\right), \tag{3.32}$$

where the softsign activation function is defined as

$$\text{softsign}(x) = \frac{x}{1+|x|} \in (-1, 1). \tag{3.33}$$

The phases are then computed as $\phi(\sigma_i) = \boldsymbol{y}_i^{(2)} \cdot \boldsymbol{\sigma}_i$. Finally, the phase of the total configuration is given by the sum of the individual phases, $\phi(\boldsymbol{\sigma}) = \sum_{i=1}^{N_v} \phi(\sigma_i)$. By combining the amplitude part with the phase, the RNN can be used as variational

[1] We will from now on use $N_v$ to denote the number of input variables for the RNN, in line with the notation used for the RBM's number of visible units $N_v$.

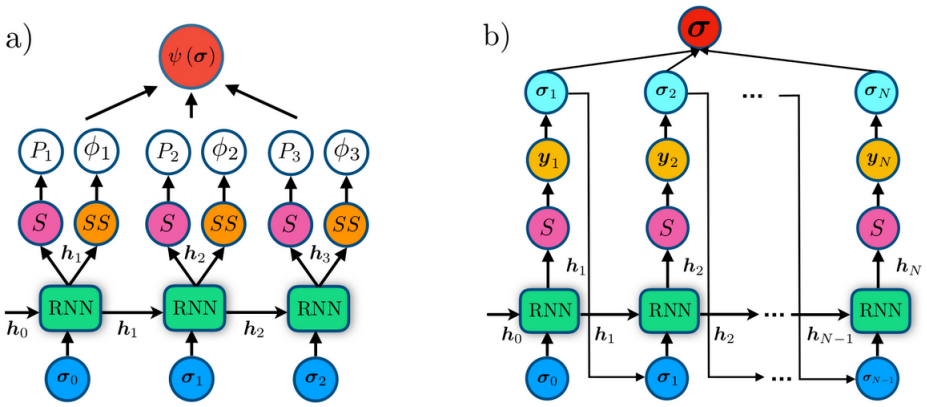



**Fig. 3.1.:** RNN wave function: a) For a given spin configuration $\boldsymbol{\sigma}$, the RNN computes the complex amplitude $\psi(\boldsymbol{\sigma})$. The softmax layer $(S)$ and softsign layer $(SS)$ are used to compute the amplitude and phase, respectively. b) A graphical representation of autoregressive sampling of spin configurations. Figure adapted from [36].

ansatz for quantum wave functions $\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \exp(i\phi(\boldsymbol{\sigma})\sqrt{p(\boldsymbol{\sigma})} \approx \psi(\boldsymbol{\sigma})$. This is illustrated in Fig. 3.1.

### 3.2.2 Autoregressive sampling and RNN optimization

Unlike energy-based methods such as the RBM (see section 3.1), RNN wave functions are normalized by construction. Furthermore, RNNs have the autoregressive property, which means that the conditional probability $p(\sigma_i|\sigma_{i-1}, \ldots, \sigma_1)$ depends only on local configurations $\sigma_j$ with $j < i$. This property allows us to directly sample successive independent samples in an autoregressive manner, thereby eliminating the need for a Markov chain and a corresponding sampling scheme.

An autoregressive sampling step consists of generating a total spin configuration $\boldsymbol{\sigma}$ by iteratively fixing the local spin states $\sigma_i$, as is illustrated in Fig. 3.1. Given a hidden vector state $\boldsymbol{h}_{i-1}$, the pass through the softmax layer provides us the vector $\boldsymbol{y}_i^{(1)}$. Using this vector, we calculate the probability $p(\sigma_i)$ that the spin at site $i$ is in state $\sigma_i$ (Eq. (3.31)). For example, if we were to use the standard basis we would obtain two probabilities, e.g. 40% probability of spin $i$ being $|\uparrow\rangle$ and 60% probability of the spin being $|\downarrow\rangle$. Using a (pseudo-)random number generator, a spin state is selected according to $p(\sigma_i)$. This new spin state together with the hidden state vector $\boldsymbol{h}_{i-1}$ are then processed by the recurrent cell, such that a new hidden state vector $\boldsymbol{h}_i$ is obtained. The process is repeated until $N_v$ local spin states are determined, which constitutes one sample configuration $\boldsymbol{\sigma}$. Note that the sampling time is linear in the size of the system, and that the sampling procedure can be parallelized.

After a chosen number of samples $N_s$ have been generated, expectation values such as the energy are computed using Eq. (3.8). As described in section 3.1.4, optimizing a neural network ansatz means minimizing the energy in a variational Monte Carlo setting. In practice, the gradients (Eq. (3.14)) are calculated using automatic differentiation [52]. The parameters are subsequently updated using gradient descent or any alternative update rule. In this work, the RNN is updated according to the Adam scheme Eq. (3.15) with a scale $\eta$ declining according to

$$\eta_t = \frac{\eta_{t-1}}{1 + 0.1\mathrm{t}}, \tag{3.34}$$

with $t$ the optimization step. Since the Adam optimizer keeps track of multiple learning rates (one for each variational parameter) we refer to the overall scale $\eta$ simply as "the learning rate".

## 3.3 Implementing SU(2) symmetry in artificial neural networks

### 3.3.1 Discrete lattice symmetries

Before we discuss how to construct SU(2) invariant ANNs, we first demonstrate how discrete lattice symmetries are usually implemented. To this end, suppose that we are dealing with a one-dimensional spin chain with periodic boundary conditions. Denoting the translation operator as $\hat{T}_n$, its action is to translate the spins on the chain by $n$ shifts to the right. We write $\hat{T} \in G = \{\hat{I}, \hat{T}_1, \ldots, \hat{T}_{N_v-1}\}$ and have that $\hat{T}_1^n = \hat{T}_n$, $\hat{T}_{N_v} = \hat{I}$ and $|G| = N_v$. Since an arbitrary spin state (expressed in the $\hat{\sigma}^z$-basis) transforms as $\hat{T}_n \left|\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z\right\rangle = \left|\sigma_{1-n}^z, \sigma_{2-n}^z, \ldots, \sigma_{N_v-n}^z\right\rangle$, an arbitrary wave function $|\Psi\rangle$ transforms as

$$\begin{aligned}
\hat{T}_n |\Psi\rangle &= \hat{T}_n \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma}) |\boldsymbol{\sigma}\rangle , && (3.35)\\
&= \sum_{\{\sigma_i^z\}} \psi(\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z)\, \hat{T}_n \left|\sigma_1^z \sigma_2^z \ldots \sigma_{N_v}^z\right\rangle , && (3.36)\\
&= \sum_{\{\sigma_i^z\}} \psi(\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z) \left|\sigma_{1-n}^z, \sigma_{2-n}^z, \ldots, \sigma_{N_v-n}^z\right\rangle , && (3.37)
\end{aligned}$$

where we have periodicity over the indices ($\sigma_{N_v+1}^z = \sigma_1^z$). If a Hamiltonian $\hat{H}$ has translational symmetry, i.e. if $[\hat{T}, \hat{H}] = 0$, its ground state is an eigenvector of the symmetry transformation $\hat{T}$. From the equations above, we see that this holds when

the complex amplitudes follow $\psi_{gs}(\boldsymbol{\sigma}) = \omega_{\hat{T}}\, \psi_{gs}(\hat{T}[\boldsymbol{\sigma}])$, where $\omega_{\hat{T}}$ is an eigenvalue with $\|\omega_{\hat{T}}\| = 1$. Since this expression is independent of $\boldsymbol{\sigma}$, the transformation changes the ground state with only a global phase term, and the probability $\|\psi_{gs}\|^2$ remains unchanged. This can be implemented in NQSs in either one of two ways.

**Approach I. Manipulation of the network structure** The first approach is to directly impose the symmetry by carefully manipulating the network structure. To illustrate this, consider the RBM, where the complex amplitudes are calculated as in Eq. (3.4), repeated here for convenience:

$$\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \exp\left(\sum_i a_i \sigma_i^z\right) \times \prod_{i=1}^{N_h} 2\cosh\left[b_i + \sum_j^{N_v} w_{ij}\, \sigma_j^z\right]. \tag{3.38}$$

Translational invariance can be implemented by placing constraints upon the variational parameters. For example, it is apparent that the elements of the bias vector $a_i$ all need to be equal — the vector $a_i$ is reduced to a scalar $a$. Now, take the number of hidden units $N_h$ to be a multiple of the number of visible units $N_v$, such that the ratio is given by $\alpha = \frac{N_h}{N_v}$. The weight matrix $w_{ij}$ then takes a rectangular form of dimension $\alpha N_v \times N_v$. We can identify a number $\alpha$ of different blocks of rows, where the rows in each block follow a particular pattern. Namely, each row is obtained by shifting the entries of the preceding row, meaning rows are translated copies of each other. Therefore, only one vector per block is needed to fully determine the weight matrix $w_{ij}$. We say that the weight matrix takes the form of feature filters $w_j^{(f)}$, with $f \in [0, \alpha]$, because of its resemblance to the filters in convolutional neural networks. The bias vector $b_i$ is made to only hold $\alpha$ different values, which can be seen as the biases of the filters. The expression for the translation invariant complex amplitude has now become

$$\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \exp\left(a \sum_i \sigma_i^z\right) \times \prod_{f=1}^{\alpha} 2\cosh\left[b_f + \sum_j^{N_v} w_{fN_v,j}\, \sigma_j^z\right]. \tag{3.39}$$

Note that this approach reduces the number of variational parameters. The bias vector $a_i \in \mathbb{C}^{N_v}$ has been reduced to a scalar, the bias vector $b_i \in \mathbb{C}^{N_h}$ ends up with $\alpha$ entries left, and the dimension of the weight matrix went from $\alpha N_v^2$ entries to only $\alpha N_v$. This strategy is, however, not guaranteed to work for any given neural network. Furthermore, the loss of variational freedom might result in the network having difficulties finding the optimal ground state representation.

**Approach II. Linear combinations of amplitudes** The second approach leaves the network structure intact, but includes an extra step in the calculation of complex amplitudes. Continuing with the case of translational symmetry, define the amplitude $\psi_{\mathcal{W}}(\boldsymbol{\sigma})$ as a linear combination of the amplitudes corresponding to transformed configurations $\hat{T}(\boldsymbol{\sigma})$, such that

$$\psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \frac{1}{|G|} \sum_{\hat{T} \in G} \psi_{\mathcal{W}}(\hat{T}[\boldsymbol{\sigma}]). \tag{3.40}$$

Note that by using this definition, all amplitudes $\psi_{\mathcal{W}}(\boldsymbol{\sigma}')$ with $\boldsymbol{\sigma}' = \hat{T}[\boldsymbol{\sigma}]$, are equal. The complex amplitudes are therefore invariant under translations, and the quantum state $|\Psi_{\mathcal{W}}\rangle = \sum_{\boldsymbol{\sigma}} \psi_{\mathcal{W}}(\boldsymbol{\sigma}) |\boldsymbol{\sigma}\rangle$ has translational symmetry. This approach works for any discrete lattice symmetry and is independent of the network structure.

### 3.3.2 SU(2) symmetry

In contrast to the lattice symmetries discussed above, the SU(2) symmetry group is continuous (section 1.4.2). There is no straightforward way of imposing this symmetry by manipulating the network structure of RBMs or RNNs. Recently, specific networks that directly encode SU(2) symmetry into their structure have been designed [53]. There is, however, not much freedom left in the form of these networks, since they are specifically designed for implementing SU(2) symmetry. Taking linear combinations of transformed amplitudes as in Eq. (3.40) is not a valid strategy, because SU(2) symmetry is a continuous symmetry. The sum would have to be generalized to an integral, and an exact treatment (of the symmetry) would require the calculation of an infinite number of amplitudes.

**NQSs in the coupled basis** The eigenstates of a Hamiltonian with SU(2) symmetry are also eigenstates of the SU(2) symmetry operation with well-defined total angular momentum. The idea is to move from the standard basis which expresses the spins in terms of the eigenvalues of the $\hat{\sigma}^z$-operator ($\sigma_i^z \in \{+1, -1\}$), to the coupled basis in which the states are expressed using intermediate angular momentum eigenvalues ($j_i \in \{0, \frac{1}{2}, 1, ...\}$) and where we have well-defined total angular momentum $J$ [54]

$$\begin{aligned} |\boldsymbol{\sigma}\rangle &= |\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z\rangle && (\text{e.g. } |+1, +1, \ldots, -1\rangle)\,, \\ &\Downarrow \quad \textit{change to coupled basis} \\ |\boldsymbol{\sigma}\rangle &= |j_1, j_2, \ldots, j_{N_v-1}; j_{N_v} \equiv J\rangle && \left(\text{e.g. } |\tfrac{1}{2}, 1, \ldots, 0\rangle\right). \end{aligned} \tag{3.41}$$

Unlike the original formulation of the ansatz as in Eq. (3.2), we use the intermediate degrees of freedom $j_i$ as inputs for the NQS

$$\begin{aligned} |\Psi\rangle &= \sum_{\{\sigma_i^z\}} \psi(\sigma_1^z, \sigma_2^z, \ldots, \sigma_{N_v}^z) \, |\sigma_1^z \sigma_2^z \ldots \sigma_{N_v}^z\rangle \,, \\ &\Downarrow \quad \textit{change to coupled basis} \\ |\Psi\rangle &= \sum_{\{j_i\}} \psi(j_1, j_2, \ldots, j_{N_v-1}) \, |j_1, j_2, \ldots, j_{N_v-1}; J\, M_J\rangle \,, \end{aligned} \tag{3.42}$$

where $\sum_{\{j_i\}}$ denotes the summation over all physically allowed configurations. The representation in terms of basis states $|j_1, j_2, \ldots, j_{N_v-1}; J\, M_J\rangle$ with a total angular momentum $J$ forms an irreducible representation (irrep) with respect to SU(2) symmetry. The irrep is labeled by angular momentum $J$ and has dimension $2J+1$, since $M_J \in \{-J, -J+1, \ldots, 0, \ldots, J\}$. The action of the SU(2) symmetry operator Eq. (1.39) leaves the total angular momentum invariant, as it transforms only the angular momentum projection degrees of freedom $M_J$. In particular, states with $J = 0$ are irreps with dimension 1, meaning these states are manifestly invariant under SU(2) transformations. We will now discuss how we move to the coupled basis and show how matrix elements are calculated.

**Spin coupling** We want to represent wave functions with well-defined total angular momentum $|J\, M_J\rangle$, which belong to the subspace spanned by states with quantum numbers $J$ and $M_J$ of the full Hilbert space. We do this by expanding the wave function in terms of basis states with intermediate angular momentum degrees of freedom $j_i$, such that

$$|\Psi\rangle = \sum_{\{j_i\}} \psi(j_1, j_2, \ldots, j_{N_v-1}) \, |j_1, j_2, \ldots, j_{N_v-1}; J\, M_J\rangle \,. \tag{3.43}$$

The intermediate angular momenta $j_i$ can be obtained using a coupling scheme, wherein two degrees of freedom are sequentially coupled to a new degree of freedom. Two angular momenta $\hat{\boldsymbol{j}}_A$ and $\hat{\boldsymbol{j}}_B$ are coupled to a single angular momentum degree of freedom $\hat{\boldsymbol{j}}_{AB}$ using Clebsch-Gordan coefficients

$$|j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle = \sum_{m_{j_A}, m_{j_B}} \langle j_A\, m_{j_A}, j_B\, m_{j_B} | j_{AB}\, m_{j_{AB}}\rangle \, |j_A\, m_{j_A}, j_B\, m_{j_B}\rangle \,, \tag{3.44}$$

where $C^{j_{AB}\, m_{j_{AB}}}_{j_A m_{j_A}, j_B m_{j_B}} = \langle j_A\, m_{j_A}, j_B\, m_{j_B} | j_{AB}\, m_{j_{AB}}\rangle$ is the Clebsch-Gordan (CG) coefficient. The CG coefficients are related to Wigner $3j$-symbols

$$C^{j_{AB}\, m_{j_{AB}}}_{j_A m_{j_A}, j_B m_{j_B}} = (-1)^{-j_A + j_B - m_{j_{AB}}} \sqrt{2j_{AB}+1} \begin{pmatrix} j_A & j_B & j_{AB} \\ m_{j_A} & m_{j_B} & -m_{j_{AB}} \end{pmatrix} . \tag{3.45}$$

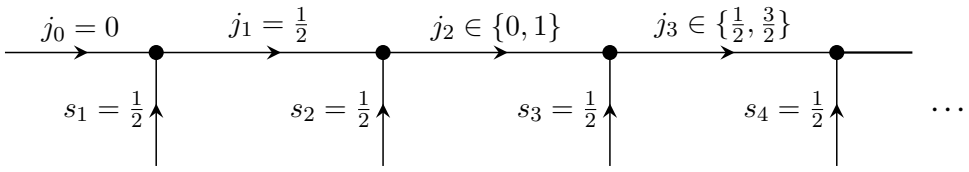



**Fig. 3.2.:** An illustration of the coupling of spins along a chain. The spin-1/2 degrees of freedom $s_i$ are coupled to intermediate angular momenta $j_i$ in a linear fashion.

Equation (3.44) shows the relation between the product states of $|j_A\, m_{j_A}\rangle$ and $|j_B\, m_{j_B}\rangle$, which are basis states of Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, and the basis states $|j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle$ of the tensor product space $\mathcal{H}_1 \otimes \mathcal{H}_2$. The states $|j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle$ are angular momentum eigenstates

$$\hat{\boldsymbol{j}}^2_{AB}\, |j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle = j_{AB}(j_{AB}+1)\, |j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle\,, \tag{3.46}$$

$$\hat{j}_{z_{AB}}\, |j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle = m_{j_{AB}}\, |j_A\, j_B; j_{AB}\, m_{j_{AB}}\rangle\,. \tag{3.47}$$

The basis states of the coupled basis are orthonormal, as follows from their definition Eq. (3.44) and $\hat{I} = \sum_{m_{j_A}, m_{j_B}} |j_A\, m_{j_A}, j_B\, m_{j_B}\rangle\, \langle j_A\, m_{j_A}, j_B\, m_{j_B}|$. The completeness of the coupled basis follows from the completeness of the product basis, and the fact that the coupled basis has the same number of orthonormal elements.

The $N_v$ spin degrees of freedom of a composite system can be coupled to a total angular momentum $\hat{\boldsymbol{J}}$ by sequentially using the coupling rule defined by Eq. (3.44). However, we are free to choose which degrees of freedom to couple at each coupling step, which leads to various coupling schemes. For one-dimensional systems, it is natural to define the linear coupling along a chain, as illustrated in Fig. 3.2.

Starting with an ancillary angular momentum state $|j_0\, m_{j_0}\rangle = |0\, 0\rangle$ that is coupled with the first spin $|s_1\, m_{s_1}\rangle$, we obtain $|j_1 = s_1\, m_{j_1} = m_{s_1}\rangle$. Subsequently, $|j_1 = s_1\, m_{j_1} = m_{s_1}\rangle$ is coupled with $|s_2\, m_{s_2}\rangle$, which gives $|j_2, m_{j_2}\rangle$ where $j_2 \in \{0, 1\}$. Repeating this process until the end of the chain is reached leads to a state $|j_{N_v} \equiv J\, m_{j_{N_v}} \equiv M_J\rangle$ of total angular momentum $J$. The intermediate angular momenta $\{j_1, j_2, \ldots, J_{N_v-1}\}$ are used as input to the neural network quantum state, such that wave functions are represented as in Eq. (3.43). According to the addition rules of angular momentum, the values of the angular momentum quantum numbers satisfy the triangle inequalities

$$|j_{i-1} - s_i| \leq j_i \leq |j_{i-1} + s_i|. \tag{3.48}$$

Therefore, the number of possible configurations with given $J$ is limited. For concreteness, consider a spin-1/2 chain of length $N_v = 6$, and suppose we restrict

ourselves to a total angular momentum of $J = 0$. The possible intermediate angular momentum states are $|\frac{1}{2}\,0\,\frac{1}{2}\,0\,\frac{1}{2}\,0\rangle$, $|\frac{1}{2}\,0\,\frac{1}{2}\,1\,\frac{1}{2}\,0\rangle$, $|\frac{1}{2}\,1\,\frac{1}{2}\,0\,\frac{1}{2}\,0\rangle$, $|\frac{1}{2}\,1\,\frac{1}{2}\,1\,\frac{1}{2}\,0\rangle$ and $|\frac{1}{2}\,1\,\frac{3}{2}\,1\,\frac{1}{2}\,0\rangle$. The number of possible configurations for a chain of length $N_v$ is given by the $n$-th Catalan number $\frac{1}{1+n}\binom{2n}{n}$, where $n = \frac{N_v}{2}$. This number grows exponentially with system size.

The full basis transformation is obtained by repeatedly applying Eq. (3.45) and can be written as

$$\begin{aligned}|s_1 \dots s_{N_v}\, j_1 \dots j_{N_v-1}; J\, M_J\rangle = \\ \sum_{\{m_{s_i}\}} \sum_{\{m_{j_i}\}} \left( \prod_{i=1}^{N_v} (-1)^{-j_{i-1}+s_i-m_{j_i}} \sqrt{2j_i+1} \begin{pmatrix} j_{i-1} & s_i & j_i \\ m_{j_{i-1}} & m_{s_i} & -m_{j_i} \end{pmatrix} \right) \\ \delta_{j_{N_v},J} \delta_{m_{j_{N_v}},M_J} \, |s_1\, m_{s_1}, s_2\, m_{s_2}, \dots, s_N\, m_{s_N}\rangle \,.\end{aligned} \tag{3.49}$$

We use the shorthand notation $|s_1 \dots s_{N_v}\, j_1\, \dots j_{N_v-1}; J\, M_J\rangle \equiv |j_1 \dots j_{N_v-1}; J\, M_J\rangle$. The orthogonality relation of $3j$-symbols is given by

$$\sum_{m_{j_{i-1}}} \sum_{m_{s_i}} (2j_i+1) \begin{pmatrix} j_{i-1} & s_i & j_i \\ m_{j_{i-1}} & m_{s_i} & m_{j_i} \end{pmatrix} \begin{pmatrix} j_{i-1} & s_i & j'_i \\ m_{j_{i-1}} & m_{s_i} & m_{j'_i} \end{pmatrix} = \delta_{j'_i,j_i} \delta_{m_{j'_i},m_{j_i}}. \tag{3.50}$$

Using Eq. (3.50) from left to right, the orthonormality of the coupled basis $|j_1 \dots j_{N_v-1}; J\, M_J\rangle$ can be shown as

$$\begin{aligned}\langle j'_1 \dots j'_{N_v-1}; J\, M_J | j_1 \dots j_{N_v-1}; J\, M_J\rangle = \\ \sum_{\{m_{s_i}\}} \sum_{\{m_{j_i}\}} \sum_{\{m_{j'_i}\}} \left( \prod_{i=1}^{N_v} (-1)^{-j_{i-1}+s_i-m_{j_i}} \sqrt{2j_i+2} \begin{pmatrix} j_{i-1} & s_i & j_i \\ m_{j_{i-1}} & m_{s_i} & -m_{j_i} \end{pmatrix} \right) \\ \left( \prod_{i=1}^{N_v} (-1)^{-j'_{i-1}+s_i-m_{j'_i}} \sqrt{2j'_i+2} \begin{pmatrix} j'_{i-1} & s_i & j'_i \\ m_{j'_{i-1}} & m_{s_i} & -m_{j'_i} \end{pmatrix} \right), \\ = \delta_{j_1,j'_1} \delta_{j_2,j'_2} \dots \delta_{j_{N_v-1},j'_{N_v-1}} \,.\end{aligned} \tag{3.51}$$

Moreover, the coupled basis is complete, since the Clebsch-Gordan coefficients relate the product states of complete bases to a complete basis of the coupled system. Consecutive coupling using the Clebsch-Gordan coefficients therefore yields a complete basis. In conclusion, the coupled basis is complete and orthonormal.

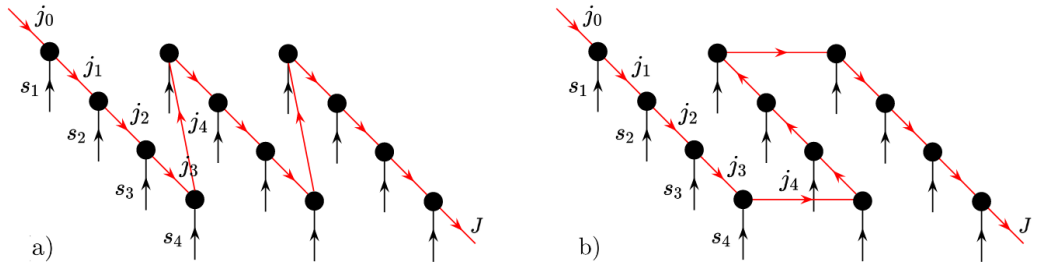



**Fig. 3.3.:** An illustration of two coupling schemes to couple the spins on a two-dimensional lattice. The spin-$^1/_2$ degrees of freedom $s_i$ are coupled to intermediate angular momenta $j_i$ in a linear fashion. a) The "ZigZag" scheme. b) The "Snake" scheme. The spin-$^1/_2$ degrees of freedom $s_i$ are coupled to intermediate angular momenta $j_i$ in a linear fashion.

For two-dimensional systems, we define the spin couplings along a one-dimensional chain. For example, a "Snake" pattern can be used to traverse the two-dimensional lattice (see Fig. 3.3). This is similar in spirit to how matrix product states (see section 3.4.2) are used to represent wave functions in two dimensions. Alternatively, the lattice can be traversed in a "ZigZag" pattern, which is also illustrated in Fig. 3.3. At first glance, the choice of pattern might look arbitrary. However, as will become clear in the forthcoming paragraph and in chapter 4, the ZigZag pattern is more advantageous for certain well-established lattice models. It is worth mentioning that coupling in a tree-like fashion is also a valid strategy, graphically depicted in Fig. 3.4. However, this scheme will not be investigated in this work (for reasons discussed in the forthcoming paragraph).

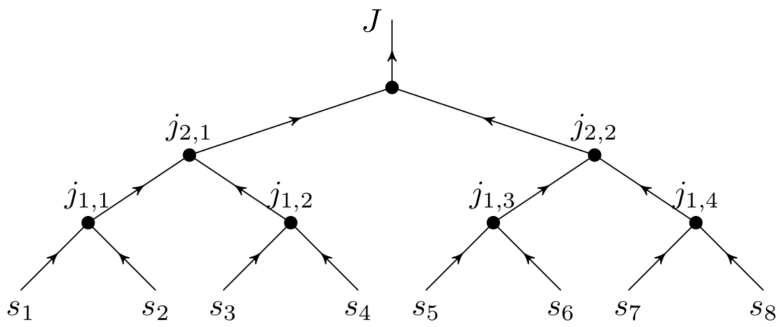



**Fig. 3.4.:** An illustration of the coupling in a tree-like fashion. The original spin-$^1/_2$ degrees of freedom $s_i$ are pairwise coupled, which results in intermediate angular momentum degrees of freedom $j_{1,i}$ which are subsequently coupled together into degrees of freedom of a second layer $j_{2,i}$. This is repeated up until the total angular momentum $J$ is obtained.

**Matrix elements in the coupled basis** We have seen in section 3.1.2 that the expectation value $\langle \hat{O} \rangle$ of some observable $\hat{O}$ is approximated by the mean of local estimators $O_L(\boldsymbol{\sigma}) = \frac{\langle \boldsymbol{\sigma}|\hat{O}|\Psi_{\mathcal{W}}\rangle}{\langle \boldsymbol{\sigma}|\Psi_{\mathcal{W}}\rangle}$, where the configurations $\boldsymbol{\sigma}$ are sampled according to $p(\boldsymbol{\sigma})$ given by Eq. (3.7). In most cases, the matrix representation of the observable is specified in the $\sigma^z$-basis. When the wave function $|\Psi_{\mathcal{W}}\rangle$ is expanded in the coupled basis, we need the transformed matrix elements to calculate $\langle \boldsymbol{\sigma}|\hat{O}|\Psi_{\mathcal{W}}\rangle$.

For example, consider the Hamiltonian $\hat{H} = \sum_i \hat{h}_i$, which we have written as a direct sum of terms $\hat{h}_i$. When dealing with systems that have local interactions, the sum in the calculation of the local energy $e_L = \sum_{\boldsymbol{\sigma}'} \psi_{\mathcal{W}}(\boldsymbol{\sigma}') \langle \boldsymbol{\sigma}|\hat{h}|\boldsymbol{\sigma}'\rangle = \langle \boldsymbol{\sigma}|\hat{h}|\Psi_{\mathcal{W}}\rangle$ contains only few non-zero terms (the matrix is sparse). This allows for efficient calculation of the local energies, and therefore an efficient optimization. We call the states $\boldsymbol{\sigma}'$ that lead to a non-zero contribution $\langle \boldsymbol{\sigma}|\hat{h}|\boldsymbol{\sigma}'\rangle$ the "connected states" of $\boldsymbol{\sigma}$ (and *vice versa*). To preserve the efficiency of the optimization scheme, we require the states of the coupled basis to have few connected states, such that the matrix elements can be computed efficiently.

The average number of connected states in the coupled basis depends on the coupling scheme. Some coupling schemes can lead to non-local interactions, such that the sparse Hamiltonian $\hat{H}$ might no longer be sparse in the coupled basis. For this reason, the coupling along a chain is suboptimal for lattice systems in more than one dimension — due to the Snake (or ZigZag) pattern that traverses the lattice, physically neighbouring spins might be well separated on the chain, and their interaction is therefore non-local in the coupled basis. Periodic boundary conditions are likewise ill-advised when using the linear chain. Coupling in a tree-like fashion is more suitable for these kind of systems. A disadvantage of the tree is that local interactions become less local, even in one-dimensional systems.

For a given state $|\boldsymbol{\sigma}\rangle = |j_1, j_2, \ldots, j_{N_v-1}; J\, M_J\rangle$ and a term of the Hamiltonian $\hat{h}$, the connected states $|\boldsymbol{\sigma}'\rangle = |j_1', j_2', \ldots, j_{N_v-1}'; J\, M_J\rangle$ and the corresponding non-zero matrix element $\langle \boldsymbol{\sigma}'|\hat{h}|\boldsymbol{\sigma}\rangle$ can be found by writing the operator in terms of a contraction of Clebsch-Gordan coefficients. That the operator can be written this way is guaranteed by the Wigner-Eckart theorem [55]. For a detailed and rigorous discussion of how the matrix elements are found, see Ref. [56]. Here, we will restrict ourselves to an introduction to the topic. As we will see in the next chapter, we are interested in spin-spin interactions of the form $\hat{h} = \hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2$. This interaction can be rewritten as

$$\hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2 = \frac{1}{2}\left[(\hat{\boldsymbol{s}}_1 + \hat{\boldsymbol{s}}_2)^2 - \hat{\boldsymbol{s}}_1^2 - \hat{\boldsymbol{s}}_2^2\right]. \tag{3.52}$$

A state $|s_1\, m_{s_1}, s_2\, m_{s_2}\rangle$ can be related to states of the coupled basis using the Clebsch-Gordan coefficients (inverse transformation of Eq. (3.44))

$$|s_1\, m_{s_1}, s_2\, m_{s_2}\rangle = \sum_{j,m_j} \langle j\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, |j\, m_j\rangle\,, \tag{3.53}$$

where $j \in \{0, 1\}$. We can explicitly write out the sum $\sum_j$ and apply Eq. (3.52)

$$\begin{aligned}\hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2\, |s_1\, m_{s_1}, s_2\, m_{s_2}\rangle = \frac{1}{2}\Bigg( &\left[0 - \frac{3}{4} - \frac{3}{4}\right] \sum_{m_j} \langle 0\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, |0\, m_j\rangle \\ &+ \left[2 - \frac{3}{4} - \frac{3}{4}\right] \sum_{m_j} \langle 1\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, |1\, m_j\rangle \Bigg)\,.\end{aligned} \tag{3.54}$$

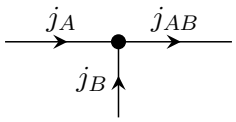
To calculate the matrix element $\langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | \hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2 | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle$, we rewrite the $\langle s_1\, m'_{s_1}, s_2\, m'_{s_2}|$ using Eq. (3.53) and take the overlap with Eq. (3.54)

$$\begin{aligned}&\langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | \hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2 | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle = \\ &\quad -\frac{3}{4} \sum_{j',m_{j'}} \sum_{m_j} \langle 0\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, \langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | j'\, m_{j'}\rangle\, \langle j'\, m_{j'} | 0\, m_j\rangle \\ &\quad +\frac{1}{4} \sum_{j',m_{j'}} \sum_{m_j} \langle 1\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, \langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | j'\, m_{j'}\rangle\, \langle j'\, m_{j'} | 1\, m_j\rangle\,.\end{aligned} \tag{3.55}$$

We have $\langle j'\, m_{j'} | 0\, m_j\rangle = \delta_{j',0}\delta_{m_{j'},m_j}$ and $\langle j'\, m_{j'} | 1\, m_j\rangle = \delta_{j',1}\delta_{m_{j'},m_j}$. Therefore, we set $j' = 0$ in the first term of Eq. (3.55) and $j' = 1$ in the second term. The sums $\sum_{j',m_{j'}}$ disappear ($m_{j'} = m_j$) to give

$$\begin{aligned}&\langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | \hat{\boldsymbol{s}}_1 \cdot \hat{\boldsymbol{s}}_2 | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle = \\ &\quad -\frac{3}{4} \sum_{m_j} \langle 0\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, \langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | 0\, m_j\rangle \\ &\quad +\frac{1}{4} \sum_{m_j} \langle 1\, m_j | s_1\, m_{s_1}, s_2\, m_{s_2}\rangle\, \langle s_1\, m'_{s_1}, s_2\, m'_{s_2} | 1\, m_j\rangle\,.\end{aligned} \tag{3.56}$$

We now use a diagrammatic notation of CG coefficients, defined by

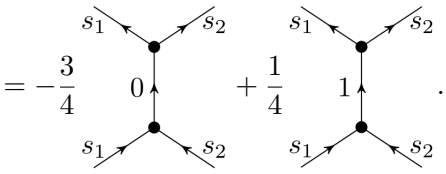
$\langle j_A\, m_{j_A}, j_B\, m_{j_B} | j_{AB}\, m_{j_{AB}}\rangle =$ 

. (3.57)

The two incoming arrows are coupled to the outgoing arrow. We can build networks using these diagrams. An edge between two nodes is interpreted as a sum over the

projection of the corresponding angular momentum. The products of CG coefficients in Eq. (3.56) can be written using the diagrammatic notation of Eq. (3.57)

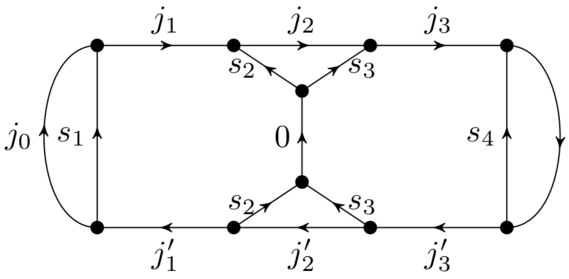
$$\langle s_1\, m'_{s_1}, s_2\, m'_{s_2}|\hat{h}|s_1\, m_{s_1}, s_2\, m_{s_2}\rangle = \langle s_1\, m'_{s_1}, s_2\, m'_{s_2}| -\frac{3}{4}\hat{h}_0 + \frac{1}{4}\hat{h}_1|s_1\, m_{s_1}, s_2\, m_{s_2}\rangle$$




(3.58)

This expression is for a system consisting of two spins. Of course, the systems we will be considering involve more spins. To ease the notation of the diagrams, we perform the calculation for a system of $4$ spins, i.e. $|\boldsymbol{\sigma}\rangle = |j_1\, j_2\, j_3; J\, M_J\rangle$, and observe at the end of the calculation that this has no influence on the matrix elements. We focus on the first term of the matrix element for $\hat{h} = \hat{\boldsymbol{s}}_2 \cdot \hat{\boldsymbol{s}}_3$, and write the overlap as

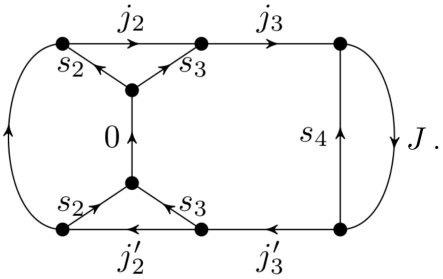
$$\langle \boldsymbol{\sigma}'|\hat{h}_0|\boldsymbol{\sigma}\rangle =$$




(3.59)

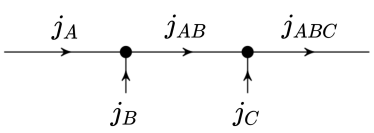
Using the orthogonality relations of CG coefficients, we obtain

$$\langle \boldsymbol{\sigma}'|\hat{h}_0|\boldsymbol{\sigma}\rangle =$$

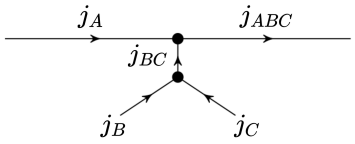



(3.60)

In the next step, we recouple the coefficients using $6j$-symbols. This is called an F−move, and the factors that enter are the F−factors. The recoupling is given by




$$= \sum_{j_{BC}} F^{j_{BC},j_{AB}}_{j_C,j_B,j_A,j_{ABC}}$$



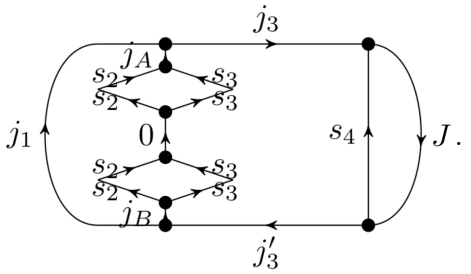

(3.61)

with the F−factor defined as

$$F^{j_{BC},j_{AB}}_{j_C,j_B,j_A,j_{ABC}} = (-1)^{j_C+j_B+j_A+j_{ABC}}\sqrt{(2j_{BC}+1)(2j_{AB}+1)}\begin{Bmatrix} j_C & j_B & j_{BC} \\ j_A & j_{ABC} & j_{AB}\end{Bmatrix}, \tag{3.62}$$

where the curly brackets indicate the $6j$-symbols, which are closely related to the $3j$-symbols. By applying the recoupling, Eq. (3.60) is rewritten as

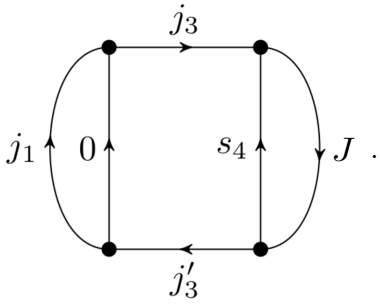
$$\langle \boldsymbol{\sigma}'|\hat{h}_0|\boldsymbol{\sigma}\rangle = \sum_{j_A}\sum_{j_B} F^{j_A,j_2}_{s_3,s_2,j_1,j_3} F^{j_B,j_2'}_{s_3,s_2,j_1,j_3}$$




(3.63)

The sum vanishes due to the orthogonality relations, giving $j_A = j_B = 0$. This yields the diagram

$$\langle \boldsymbol{\sigma}'|\hat{h}_0|\boldsymbol{\sigma}\rangle = F^{0,j_2}_{s_3,s_2,j_1,j_3} F^{0,j_2'}_{s_3,s_2,j_1,j_3}$$




(3.64)

Again using the orthogonality relations of Clebsch-Gordan coefficients, the diagram can be fully contracted to unity. The result is an overlap given by

$$\langle j_1\, j_2'\, j_3; J\, M_J|\hat{h}_0|j_1\, j_2\, j_3; J\, M_J\rangle = F^{0,j_2}_{s_3,s_2,j_1,j_3} F^{0,j_2'}_{s_3,s_2,j_1,j_3}. \tag{3.65}$$

The second part of the matrix element in Eq. (3.58) can be calculated in a similar fashion. We observe that the matrix elements of an interaction in the coupled basis depend only on the angular momentum which couples the physical spins, and the neighbours of this angular momentum. Due to the symmetries of $6j$-symbols, the precalculated F−factors can be efficiently stored [57]. Furthermore, the momenta entering in the $6j$-symbols have to fulfill four triangle relations.

## 3.4 Other methods

There are many methods to approximate quantum many-body wave functions, especially for the ground state and the low-lying excited states. Most of these methods rely on finding an efficient ansatz for the wave function (or rather its complex coefficients), similar to the NQS approach introduced in sections 3.1 and 3.2. We focus here on two important methods, the first being exact diagonalization (ED), which lies closest to the original formulation of the problem (solving the Schrödinger equation Eq. (1.22)). This technique will be used as a reference, i.e., to verify the accuracy of the NQS approach, as it is exact. It is, however, only tractable for small systems, which is why new and more efficient strategies are constantly being developed. Afterwards, we will briefly discuss the formalism of tensor networks, which have been central in most of the recent literature about quantum many-body systems. Finally, we discuss the density matrix renormalization group (DMRG) method, the state-of-the-art for approximating ground-state wave functions for one-dimensional lattices. In the forthcoming chapter, we will simulate systems that are too large for ED, and as a consequence, we compare our results to what is obtained using DMRG.

### 3.4.1 Exact diagonalization

The spectrum of a time-independent Hamiltonian $\hat{H}$ can be found by solving the time-independent Schrödinger equation Eq. (1.22). This is an eigenvalue problem, i.e., the eigenvectors $|\Psi_n\rangle$ and eigenvalues $E_n$ can be found by diagonalization of the matrix $\hat{H}$. However, standard diagonalization techniques (such as the LAPACK routine [58]) require a disk space $\sim \mathcal{D}^2$ and their CPU-time increases as $\mathcal{D}^3$, where $\mathcal{D}$ is the size of the Hilbert space. In general, the Hilbert space increases exponentially in the size of the system $N_v$ (e.g. $\mathcal{D} = D^{N_v}$ with $D = 2$ for spin-$1/2$). Accordingly, full exact diagonalization is unfeasible for systems that are big or have a large local Hilbert space $D$.

**Lanczos algorithm** Often, one is interested in the low-lying states of a sparse Hamiltonian $\hat{H}$. When resorting to diagonalization of $\hat{H}$, the Lanczos algorithm [59] is especially useful in these cases, as it provides a considerable speed-up when compared to standard methods. It is a general procedure that transforms a symmetric $\mathcal{D} \times \mathcal{D}$ matrix into a symmetric $M \times M$ tridiagonal matrix, thereby reducing the dimension $M < \mathcal{D}$. The starting point is a random normalized state $|\phi_0\rangle$, which we assume not to be orthogonal to the ground state. The matrix operator $\hat{H}$ is applied

to $|\phi_0\rangle$, and the resulting vector is split into a component parallel to $|\phi_0\rangle$ and an orthogonal component $|\phi_1\rangle$

$$\hat{H}\,|\phi_0\rangle = a_0\,|\phi_0\rangle + b_1\,|\phi_1\rangle\,, \tag{3.66}$$

where $a_0 = \langle\phi_0|\hat{H}|\phi_0\rangle$ and $b_1 = \langle\phi_1|\hat{H}|\phi_0\rangle$. From Eq. (3.66) we also find

$$b_1^2 = \langle\phi_0|(\hat{H}-a_0)(\hat{H}-a_0)|\phi_0\rangle = \langle\phi_0|\hat{H}^2|\phi_0\rangle - a_0^2, \tag{3.67}$$

and thus $b_1$ is the mean square energy deviation of $|\phi_0\rangle$. In the next step $\hat{H}$ is applied to $|\phi_1\rangle$, which gives

$$\hat{H}\,|\phi_1\rangle = b_1'\,|\phi_0\rangle + a_1\,|\phi_1\rangle + b_2\,|\phi_2\rangle\,, \tag{3.68}$$

where $|\phi_2\rangle$ is orthogonal to both $|\phi_0\rangle$ and $|\phi_1\rangle$, which implies

$$\langle\phi_1|\phi_2\rangle = 0 \iff a_1 = \langle\phi_1|\hat{H}|\phi_1\rangle\,, \tag{3.69}$$

$$\langle\phi_1|\phi_2\rangle = 0 \iff \langle\phi_0|\hat{H}|\phi_1\rangle - b_1' = 0 \iff b_1' = b_1. \tag{3.70}$$

Also, $|\phi_2\rangle$ being normalized leads to $b_2 = \langle\phi_3|\hat{H}|\phi_1\rangle$. Reiterating the above, we get in $i$ steps

$$\hat{H}\,|\phi_i\rangle = b_i\,|\phi_{i-1}\rangle + a_i\,|\phi_i\rangle + b_{i+1}\,|\phi_{i+1}\rangle\,, \quad 1 \le i \le M, \tag{3.71}$$

where we note that there are, by construction, no terms involving $|\phi_{i-2}\rangle$ etc. This is an important point: at each Lanczos step, we need to orthogonalize the vector only to the previous two vectors, the orthogonality to the rest of the vectors is automatic (up to numerical rounding error). If we stop the algorithm at iteration $i = M$ and set $b_{M+1} = 0$, the Hamiltonian can be represented in the basis of orthogonal Lanczos functions $|\phi_i\rangle$ as

$$\hat{H}_M = \begin{pmatrix} a_0 & b_1 & 0 & \dots & 0 \\ b_1 & a_1 & b_2 & & 0 \\ 0 & b_2 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & b_{M-1} \\ 0 & 0 & \dots & b_{M-1} & a_M \end{pmatrix}, \tag{3.72}$$

which is a tridiagonal symmetric matrix. These kinds of matrices can be efficiently diagonalized using standard techniques, mostly based around the so-called QR algorithm [60] or the divide-and-conquer method [61]. Specialized algorithms

allow the determination of the eigenvalues in $\mathcal{O}(M \log M)$ operations [62]. Thus, we finally obtain approximations to the eigenvectors

$$|\Psi_j\rangle = \sum_{i=0}^{M} c_{ji} |\phi_i\rangle , \quad j = 0, 1, \ldots, M , \tag{3.73}$$

with the corresponding eigenvalues $E_j$. The Lanczos algorithm is known to converge fast for lower and upper eigenvalues [63]. The number of steps needed to obtain a given accuracy depends on the Hamiltonian (typically $M \sim 10^2$). The Lanczos approach is efficient for sparse Hamiltonians, since the number of operations is proportional to $n_{\text{CS}} M\mathcal{D}$, where $n_{\text{CS}} << \mathcal{D}$ is the number of connected states for each basis state. The memory requirements scale as $M\mathcal{D}$, but for the evaluation of the eigenvalues alone, only three $|\phi_i\rangle$ vectors are successively required, meaning the scaling reduces to $3\mathcal{D}$. An additional memory $\sim n_{\text{CS}}\mathcal{D}$ is needed if the matrix elements are precalculated and stored instead of being calculated on the fly.

### 3.4.2 Tensor networks

As discussed in section 1.2, a quantum state can be expanded in a basis (Eq. (1.20)). The number of expansion coefficients $\mathcal{D}$ scales exponentially with the system size, and becomes cumbersome for large systems. This is exactly why we introduced the NQS ansatz in the previous sections: the NQS is a parameterization of the wave function, and, since the number of parameters of the NQS is much lower than $\mathcal{D}$ (and information can be efficiently extracted from the representation), the NQS representation is efficient. This is the major motivation for using an ansatz in many-body quantum physics, and this methodology is certainly not restricted to NQSs.

**General idea behind tensor networks** The complex coefficients

$$\psi(\boldsymbol{\sigma}) = \psi(\sigma_1 \sigma_2 \ldots \sigma_{N_v}) = C_{\sigma_1, \sigma_2, \ldots, \sigma_{N_v}} , \tag{3.74}$$

can be interpreted as the entries of a tensor $C$ of rank $N_v$, where each index $\sigma_i$ can take up to $D$ different values. The idea behind Tensor Networks (TN) is to reduce the complexity (the number of parameters) by replacing the tensor $C$ by a network of tensors of smaller rank [64]. The total number of parameters $n_{\text{par, tot}}$ is given by $n_{\text{par, tot}} = \sum_t n_{\text{par}}(t)$, where $n_{\text{par}}(t)$ is the number of parameters of tensor $t$ in the TN and the sum runs over all tensors in the network. A practical TN has that the number of tensors $N_t = \mathcal{O}(\text{poly}(N_v))$.

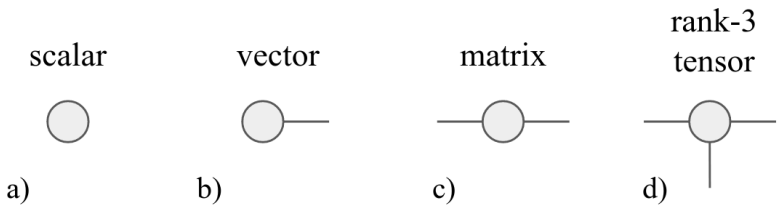



**Fig. 3.5.:** Diagrammatic notation of tensors: a) scalar; b) vector; c) matrix; d) rank-3 tensor.

For a given tensor $t$, the number of parameters is

$$n_{\text{par}}(t) = \mathcal{O}\left(\prod_{\alpha_t=1}^{\text{rank}(t)} B(\alpha_t)\right), \tag{3.75}$$

where $\alpha_t$ denotes the different indices of the tensor and $B(\alpha_t)$ is the different possible values of index $\alpha_t$. Taking $B_t = \max(B(\alpha_t))$, we have $n_{\text{par}}(t) = \mathcal{O}(B_t^{\text{rank}(t)})$. We thus find the total number of parameters to be

$$n_{\text{par, tot}} = \sum_{t=1}^{N_t} \mathcal{O}\left(B_t^{\text{rank}(t)}\right) = \mathcal{O}(\text{poly}(N_v)\text{poly}(\mathcal{B})), \tag{3.76}$$

where the bond dimension $\mathcal{B} = \max(B_t)$ is the maximum of $B_t$ over all tensors.

To sketch how this works in practice, we will consider the class of Matrix Product States (MPS). At this point, it is convenient to introduce a diagrammatic notation in which a tensor is represented by a circle with a number of outgoing lines, as illustrated in Fig. 3.5. The lines correspond to the indices of the tensor. If a line connects two tensors, this means that the corresponding indices are contracted. The diagrammatic notation of a general MPS with periodic boundary conditions is shown in Fig. 3.6. The corresponding formula is given by [65]

$$|\Psi_{\text{MPS}}\rangle = \sum_{\{\sigma_n\}} \sum_{\{\alpha_n\}} A^{\sigma_1}_{\alpha_0\alpha_1}(1) A^{\sigma_2}_{\alpha_1\alpha_2}(2) \dots A^{\sigma_{N_v}}_{\alpha_{N_v-1}\alpha_{N_v}}(N_v) \, |\sigma_1\sigma_2\dots\sigma_{N_v}\rangle \,, \tag{3.77}$$

$$= \sum_{\{\sigma_n\}} \text{Tr}[A^{\sigma_1}(1) A^{\sigma_2}(2) \dots A^{\sigma_{N_v}}(N_v)] \, |\sigma_1\sigma_2\dots\sigma_{N_v}\rangle \,, \tag{3.78}$$

where the trace reflects the periodic boundary conditions $\alpha_0 = \alpha_{N_v}$, and the range of summation $\alpha_i = 1, \dots, B_i$ can be different for the various indices. The result of the contraction can be seen as a tensor of rank $N_v$, where each index $\sigma_i$ runs over $D$ different values. The tensor represents the $\mathcal{D} = D^{N_v}$ coefficients, but they are (necessarily) not independent — structure is introduced by the contraction of a given TN. For a practical discussion on tensor networks which includes the implementation of symmetries and the calculation of expectation values, we refer to Ref. [66].

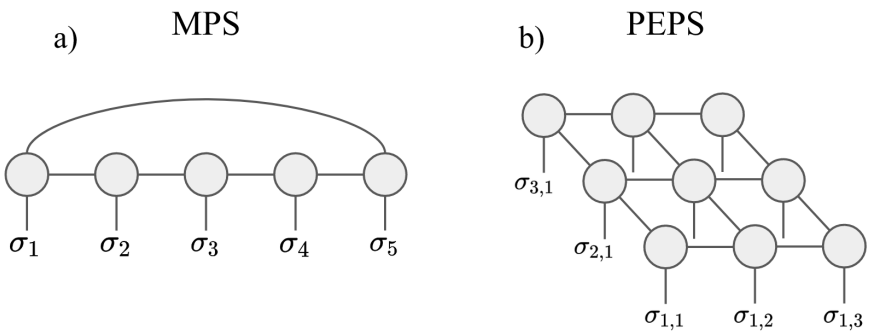



**Fig. 3.6.:** Examples of tensor networks. Lines connected on both ends indicate the contraction of the corresponding indices. Open indices typically coincide with the physical degrees of freedom. a) matrix product state (MPS) of 5 spins with periodic boundary conditions b) projected entangled pair state (PEPS) of a $3 \times 3$ spin system with open boundary conditions.

**Entanglement entropy and relation to RBMs** The interconnected indices have a physical meaning: they represent the structure of the entanglement in the quantum state, and the number of different values they can adopt is a measure of the amount of quantum correlations in the wave function [64]. Increasing these amounts, the so-called bond dimensions, increases the complexity of the TN. The relation between the connected indices and the entanglement properties can be made explicit by considering the two-dimensional generalization of an MPS, namely the projected entangled pair state (PEPS), see Fig 3.6. For simplicity, assume that all indices have bond dimension $D$. We can split the system in two complementary regions, say, the inner region "$in$" of size $L \times L$ and the outer region "$out$". Define the combined index $\bar{\alpha} = \{\alpha_1 \alpha_2 \dots \alpha_{4L}\}$ of all indices across the boundary of the inner region. We have that $\bar{\alpha}$ can take up to $D^{4L}$ different values. Writing the state in terms of unnormalized kets we get

$$|\Psi_{\text{PEPS}}\rangle = \sum_{\bar{\alpha}=1}^{D^{4L}} |in(\bar{\alpha}\rangle \otimes |out(\bar{\alpha})\rangle \,. \tag{3.79}$$

The reduced density matrix of the inner part is

$$\hat{\rho}_{in} = \sum_{\bar{\alpha},\bar{\alpha}'} X_{\bar{\alpha}\bar{\alpha}'} \, |in(\bar{\alpha})\rangle \, \langle in(\bar{\alpha}')| \,, \tag{3.80}$$

with $X_{\bar{\alpha}\bar{\alpha}'} \equiv \langle out(\bar{\alpha}')|out(\bar{\alpha})\rangle$. The entanglement entropy of the inner region is

$$S_e(L) = -\operatorname{Tr}(\hat{\rho}_{in} \log \hat{\rho}_{in}), \tag{3.81}$$

and it is upper bounded by the logarithm of the rank of $\hat{\rho}_{in}$, which is at most $D^{4L}$.

We therefore obtain

$$S_e(L) \leq 4L \log D, \tag{3.82}$$

which is an upper-bound of the *area law* for the entanglement entropy. We would have obtained the same result if we had considered the outer region "$out$". Note that if $D = 1$ we have $S_e(L) = 0$, i.e., there is no entanglement and the TN is a product state. For $D > 1$ the ansatz has area law entanglement entropy, and the entropy scales as the size of the boundary between the two regions (in this example the size of the boundary is $4L$). Increasing $D$ does not change this scaling, this would require a change in geometry, i.e., the way the indices are connected. We emphasize that this does not imply that tensor networks, e.g. MPS, can only approximate states that follow an area law. To the contrary, MPS are universal approximators, i.e. an MPS can represent any state — but it does so at the cost of a bond dimension which scales exponentially with the system size.

At first sight, the area-law entanglement entropy (3.82) might seem as a restriction of the TN ansatz. If we were to pick a random state out of the total Hilbert space, it would have volume law entanglement entropy, meaning the amount of entanglement between two regions scales according to their volumes [67]. Furthermore, classical thermal entropy is an extensive property. However, it has been proven that an important class of local gapped Hamiltonians (gapped meaning there is an energy difference between the ground state and the first excited state) follows area-law entanglement entropy [68]. The area-law relation can thus be seen as a feature of TNs, since by construction they represent states in the relevant corner of the Hilbert space, and they do so very efficiently.

One might ask if neural networks, such as the RBM (section 3.1), are also restricted to some part of the total Hilbert space, or are in some way related to TNs. It has been shown in Ref. [69] that general (long-range) RBM states exhibit volume-law entanglement, yet live in a restricted subspace of the Hilbert space. Short-range RBMs, meaning the hidden units are connected only to visible units within a certain range, show area-law entanglement. The area-law scaling with respect to the number of parameters is however not identical, and it was found that RBMs can represent highly entangled systems more efficiently than MPS. Another study [70] made the relation between RBMs and tensor networks more explicit. It showed that short-range RBMs are entangled plaquette states (EPS) [71] and that fully-connected RBMs are string-bond states (SBS) [72].

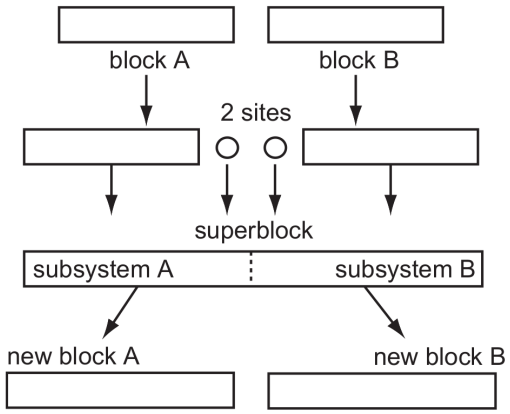



**Fig. 3.7.:** An illustration of the steps in infinite-size DMRG. New spins are added to the two blocks $A$ and $B$. The blocks are combined into a superblock, and its ground state is calculated. The reduced density matrices of the subsystems are diagonalized, their eigenvectors are ordered, and the $K$ most important ones are kept. The Hamiltonian is transformed to the basis defined by the $K$ states per block. The process is repeated until the desired size is reached. Figure from Ref. [67].

**Density matrix renormalization group (DMRG)** Tensor networks are optimized using a variety of techniques, mostly based on either a variational (Monte Carlo) optimization procedure or imaginary time evolution [64]. Interestingly, Steve White's paper on *density matrix renormalization group* [45], a very powerful method to find the ground state of one-dimensional quantum chains, appeared in 1992. The same year, a paper on "finitely correlated states on quantum spin chains" [73] was published, regarding states that are nowadays known as translation invariant matrix product states. It was later realised that DMRG implicitly represents the state of the system as a matrix product state [74]. Now, it is customary to formulate DMRG as a variational optimization over the set of matrix product states.

The DMRG algorithm has been continually refined and reformulated, as such it has been described in various forms [67]. The general idea is to start with a small system and iteratively increase its size. At each step, we construct a transformed basis and order the states according to their importance. Only the $K$ most important states are kept, the rest is discarded.[2] The system size is increased by adding two new spins, and the procedure is repeated until the desired system size is obtained. This is the essence of infinite-size DMRG, graphically depicted in Fig. 3.7.

[2]These steps have a certain resemblance to what is done in PCA, see section 2.3.1.

The infinite-size DMRG algorithm is outlined as follows [75]:

1. Start with a small system and perform exact diagonalization (e.g. the Lanczos procedure of section 3.4.1).

2. Divide the system equally in two blocks $A$ and $B$, and add a new spin to each block. Denoting the basis states of block $A$ ($B$) by $|\Lambda_A\rangle$ ($|\Lambda_B\rangle$) and the basis states of the spin added to block $A$ ($B$) by $|\lambda_A\rangle$ ($|\lambda_B\rangle$), states of the full system (the superblock (SB)) can be written as

$$|\Psi^{\text{SB}}\rangle = \sum_{\Lambda_A,\Lambda_B,\lambda_A,\lambda_B} \psi(\Lambda_A,\Lambda_B,\lambda_A,\lambda_B)\,|\Lambda_A\rangle\,|\Lambda_B\rangle\,|\lambda_A\rangle\,|\lambda_B\rangle\,. \tag{3.83}$$

   Diagonalize the full Hamiltonian, which gives the ground state $|\Psi^{\text{SB}}_{gs}\rangle$.

3. Calculate the reduced density matrices of each block and diagonalize them. For example, the reduced density matrix of block $A$ is $\hat{\rho}_A = \text{Tr}_B(\hat{\rho}_{\Psi^{SP}_{gs}})$. After diagonalization, keep only those $K$ eigenvectors that have the largest eigenvalues.

4. Transform the Hamiltonian into the new bases of $K$ states per block by using two $K \times KD$ rotation matrices.

5. Repeat steps 2-4 until the desired system size is reached.

In step 3, only the $K$ most important eigenstates are kept. Therefore, the number of basis states remains constant instead of increasing exponentially with block size. The infinite-size DMRG algorithm is usually followed by finite-size DMRG, for which we refer to Refs. [67, 75] as it involves almost the same operations as infinite-size DMRG. A lot can be said about the reason why renormalization group methods work for quantum many-body problems, or why DMRG leads to the class of MPSs. For this, we refer to Refs. [64, 65, 67, 74].

# 4 Model systems and results

In this chapter, we introduce two prototypical spin systems for quantum magnetism: i) the antiferromagnetic Heisenberg model (AFH) and ii) the $J_1 - J_2$ model. The interacting particles live on a lattice, i.e., fermionic motion is frozen. The corresponding Hamiltonians adopt a simple form and only nearest (AFH) and next-to-nearest neighbour $(J_1 - J_2)$ interactions are present. However, these "toy-models" exhibit many of the (quantum) phenomena we are interested in: quantum phase transitions, frustration effects, entanglement, etc. Because of these features, the models provide insight into exotic quantum behaviour. Their (apparent) simplicity makes them ideal for testing the limits of new methods. We examine the RBM and RNN as neural network quantum states (introduced in sections 3.1 and 3.2), and investigate our implementation of SU(2) symmetry (section 3.3).

## 4.1 Antiferromagnetic Heisenberg model

The antiferromagnetic Heisenberg (AFH) model is defined by the Hamiltonian

$$\hat{H}_{AFH} = \sum_{\langle i,j \rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j, \tag{4.1}$$

where pairs $\langle i, j \rangle$ in the sum denote nearest neighbours, and $\hat{\boldsymbol{\sigma}}_i = (\sigma_x, \sigma_y, \sigma_z)_i$ acts on site $i$. The operators $\sigma_{x,y,z}$ are the Pauli matrices. The Hamiltonian is sometimes stated in terms of spin operators $\hat{\boldsymbol{s}}_i = \frac{\hbar}{2}\hat{\boldsymbol{\sigma}}_i$, where one usually takes $\hbar = 1$. This is a rescaling of the Hamiltonian by a factor 4 and does not change the physics. The interaction $\hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j$ is called an exchange interaction due to its origin [11, 76].

We also need to specify the geometry of the system. For one-dimensional systems, the lattice reduces to a chain. We restrict ourselves to bipartite lattices when considering higher dimensions. In a bipartite lattice, the sites can be divided in two sets $A$ and $B$, such that the sites of $A$ only interact with sites from set $B$ and *vice versa*, as illustrated in Fig. 4.1. There is no *geometric frustration* when dealing with bipartite lattices. Geometric frustration means that the lattice geometry gives rise to competing interactions that try to minimize the energy in such a way that it is

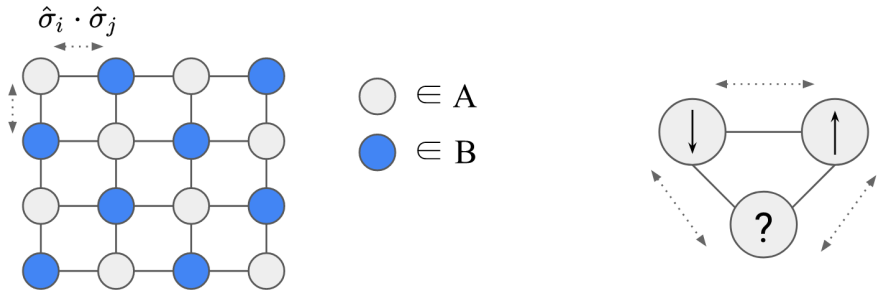



**Fig. 4.1.:** Examples of lattice geometries. a) bipartite lattice: the lattice sites can be divided in two sets $A$ and $B$, such that each site interacts only with sites from the other set. b) geometric frustration: in a triangular lattice geometry, it is impossible to minimize all antiferromagnetic interaction terms simultaneously. As a consequence, the ground state is highly degenerate.

impossible to find a unique ground state. In other words, sets of interaction terms cannot be minimized simultaneously. This type of frustration is entirely classical, and can be found in e.g. the triangular lattice, also shown in Fig. 4.1.

### 4.1.1 Background theory

For bipartite lattices, the classical ground state of antiferromagnetic systems corresponds to the Néel state, i.e. a state where interacting spins point in opposite directions (also called staggered magnetization). Expressing the local spin configurations in the $s_z$-basis and denoting the states with eigenvalue $+1$ as spin-up ($\uparrow$) and those with eigenvalue $-1$ as spin-down ($\downarrow$), the Néel state is conveniently written as $|\uparrow\downarrow\uparrow \ldots \uparrow\downarrow\rangle$. The interaction terms can be rewritten as

$$\hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_j = \frac{1}{2}[(\hat{\boldsymbol{s}}_i + \hat{\boldsymbol{s}}_j)^2 - \hat{\boldsymbol{s}}_i^2 - \hat{\boldsymbol{s}}_j^2] = \frac{1}{2}[(\hat{\boldsymbol{s}}_i + \hat{\boldsymbol{s}}_j)^2 - 2s(s+1)]. \tag{4.2}$$

If two interacting spins are combined into a singlet ($\hat{\boldsymbol{s}}_i + \hat{\boldsymbol{s}}_j = 0$), the corresponding energy term is minimized and contributes $-s(s+1)$ to the total energy. However, it is impossible to minimize every term individually: if two spins are combined into a singlet, they are maximally entangled, and can no longer be entangled with a third spin. It is said that entanglement is monogamous [11]. This leads to frustration at the quantum level.

The Hamiltonian of Eq. (4.1) commutes with the total spin operator $\hat{\boldsymbol{s}}_{\text{tot}} = \sum_i \hat{\boldsymbol{s}}_i$

$$[\hat{H}_{AFH}, \hat{\boldsymbol{s}}_{\text{tot}}] = 0. \tag{4.3}$$

Because the orbital degrees of freedom are frozen, the total spin operator $\hat{\boldsymbol{s}}_{\text{tot}}$ is equivalent to the total angular momentum operator $\hat{\boldsymbol{J}}$. It follows that the Hamiltonian also commutes with the SU(2) symmetry operator defined in Eq. (1.39). The system thus has global SU(2) spin-rotation symmetry. This means that the eigenstates of $\hat{H}_{AFM}$ can be labeled by the total angular momentum eigenvalue $J$ (of the operator $\hat{\boldsymbol{J}}^2$) and its projection $M_J$ (of the operator $\hat{J}_z$).

There is an exact result for the ground state of the AFH model that is interesting from a theoretical perspective but also used in numerical simulations. A statement known as *Marshall's sign rule*[1] states that the ground state can be written as [77]

$$|\Psi_{gs}\rangle = \sum_{\boldsymbol{\sigma}} (-1)^{M_A}\, \psi(\boldsymbol{\sigma})\, |\boldsymbol{\sigma}\rangle\,, \qquad \text{with} \quad \psi(\boldsymbol{\sigma}) \geq 0, \tag{4.4}$$

where $M_A$ is the total number of up spins in set $A$ of the bipartite lattice. That is, when expanding the wave function in the standard $s_z$-basis, the signs of the complex coefficients follow a strict rule. Combining this information with the full spin rotation symmetry leads to *Marshall's theorem*: the ground state of the AFH model is a singlet $J = 0$ and is unique.

An important notion in the classification of quantum phases is that of *gapped phases of matter*, see Fig. 4.2. A system is in a gapped phase if one can define the ground-state subspace of energy eigenstates in a certain window of size $\epsilon$, separated from the lowest lying excitations by a gap lower bounded by some constant $\Delta$ independently of the system size $N_v$ [11]. The number of states in the ground-state subspace must remain constant in the thermodynamic limit $N_v \to \infty$. Two Hamiltonians $\hat{H}_1$ and $\hat{H}_2$ are said to be in the same phase if we can continuously transform one into the other along a path of Hamiltonians which are gapped. When moving between Hamiltonians in a different phase, one passes a quantum phase transition, which is a point along the path where the Hamiltonian becomes gapless.

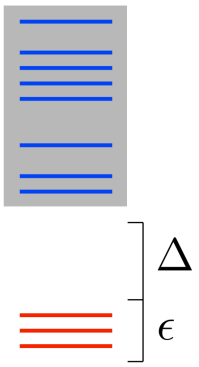



**Fig. 4.2.:** Illustration of the energy spectrum of gapped phases of matter (see main text). Taken from Ref. [11].

The Lieb-Schultz-Mattis theorem [78] states that the spin-$1/2$ AFH chain cannot have a unique ground state with an energy gap in the thermodynamic limit, and in combination with Marshall's

[1] This is also called the Marshall-Peierls sign rule.

theorem, it follows that the spin-$1/2$ AFH chain is gapless. The same is true for the two-dimensional spin-$1/2$ AFH model with an odd number of sites in the cross section [79, 80]. As stated by the Mermin-Wagner-Hohenberg-Coleman theorem [81–83], continuous symmetry breaking is not possible for classical systems in one or two dimensions. It follows from the quantum-classical correspondence (section 1.3.2) that there is no continuous symmetry breaking in the ground state of the one-dimensional AFH chain.[2] As a result, there can be no long-range order. However, the system realizes a state of quasi long-range order, i.e., the spin-spin correlation function $\langle \hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_{i+n} \rangle$ goes to zero as a power law[3]

$$\lim_{n\to\infty} \langle \hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_{i+n} \rangle = \frac{(-1)^n}{|n|^\gamma}, \tag{4.5}$$

for some exponent $\gamma$. The two-dimensional AFH model on a square lattice has long-range Néel order, although it is less pronounced than in the classical case due to quantum fluctuations [84]. Finally, we note that the one-dimensional AFH model is exactly solvable using the Bethe ansatz [85].

### 4.1.2 Results

**Sign structure initialization and translational symmetry** As discussed in section 4.1.1, the AFH ground state on a bipartite lattice follows Marshall's sign rule Eq. (4.4). Both the RBM and the RNN have complex valued expansion coefficients, such that in principle, the models can learn the sign rule during the variational optimization. However, recent literature has shown that optimization is challenging if the sign structure is not imposed [86]. This means that, in order to have an appropriate optimization, the variational ansatz takes the form of Eq. (4.4) with positive expansion coefficients $\psi(\boldsymbol{\sigma})$.

We now illustrate the need for an appropriate sing structure initialization. To this end, we use the RBM ansatz in the standard $s_z$-basis to find the ground state of the 1D AFH model with periodic boundary conditions and $N_v = 22$. Two optimizations are performed, one with and one without the sign structure imposed. The RBM has a hidden unit density $\alpha = 1$. Additionally, we train an RBM with translational symmetry (TRBM), where the symmetry is implemented by placing constraints on the network weights (section 3.3). For the TRBM, several hidden unit densities

[2] For the one-dimensional AFH model, we can derive this from the previous theorems. Symmetry breaking necessarily requires a ground state degeneracy. However, Marshall's theorem states that the ground state is unique.

[3] If there is no long-range order at all, the spin-spin correlation function decays exponentially.

$\alpha \in \{1, 10, 20\}$ are tested. Optimal values for the remaining hyperparameters are found by performing a hyperparameter sweep, see Appendix A.1. The selected values are summarized in table A.1 (Appendix A.2). In this table, one can find all the hyperparameters used in this section.

To assess the performance of the models, we use the relative energy error of the ground state

$$\Delta E_0 = \left| \frac{E_0 - E_{0,\text{exact}}}{E_{0,\text{exact}}} \right|, \tag{4.6}$$

where $E_{0,\text{exact}}$ is the exact ground-state energy obtained using Lanczos diagonalization (section 3.4.1) and $E_0$ is the energy of the NQS after convergence. At the end of each simulation, we use $10^6$ samples to evaluate $E_0$ and other expectation values according to Eq. (3.8). However, the number of samples used to estimate gradients during optimization $N_s \in [200, 2000]$ depends on the experiment.

In figure 4.3, we show the relative energy error of the ground state $\Delta E_0$ obtained by the RBM and the TRBM. Note that if the sign structure is not imposed, the RBM failes to approximate the ground state accurately, with a relative energy error $\Delta E_0$ greater than $10\%$. However, an accurate representation with $\Delta E_0 \approx 10^{-4}$ is obtained if the sign structure is implemented. The TRBM with $\alpha \in \{1, 10\}$ has a relative energy error $\Delta E_0 \in [10^{-1}, 10^{-2}]$ even if the sign structure is properly initialized. This reflects that the symmetrization procedure (section 3.3) lowers the variational freedom of the RBM, and that this needs to be compensated by a higher number of hidden units. The number of variational parameters of our models reveals that this is indeed the case: the RBM with $\alpha = 1$ has a number of parameters $n_{\text{par}} = 505$, and the TRBM with $\alpha = (1, 10, 20)$ has $n_{\text{par}} = (24, 231, 461)$. The results illustrate that it is challenging to find a good representation of the ground-state wave function in terms of a TRBM with only $[24, 231]$ variational parameters. On the other hand, a relatively high accuracy is obtained using the TRBM with $\alpha = 20$, which has a total of $461$ variational parameters. Even though this is approximately $10\%$ less parameters than the RBM with $\alpha = 1$, a similar relative energy error is reached $\Delta E_0 \sim \mathcal{O}(10^{-4})$. Note that also for the TRBM, the accuracy is extremely dependent on the initialization of the sign structure.

In the hyperparameter runs of Appendix A.1 and in all the forthcoming experiments, the sign structure Eq. (4.4) is imposed when using the standard $s_z$-basis, unless stated otherwise. No such initialization is performed for the coupled basis.

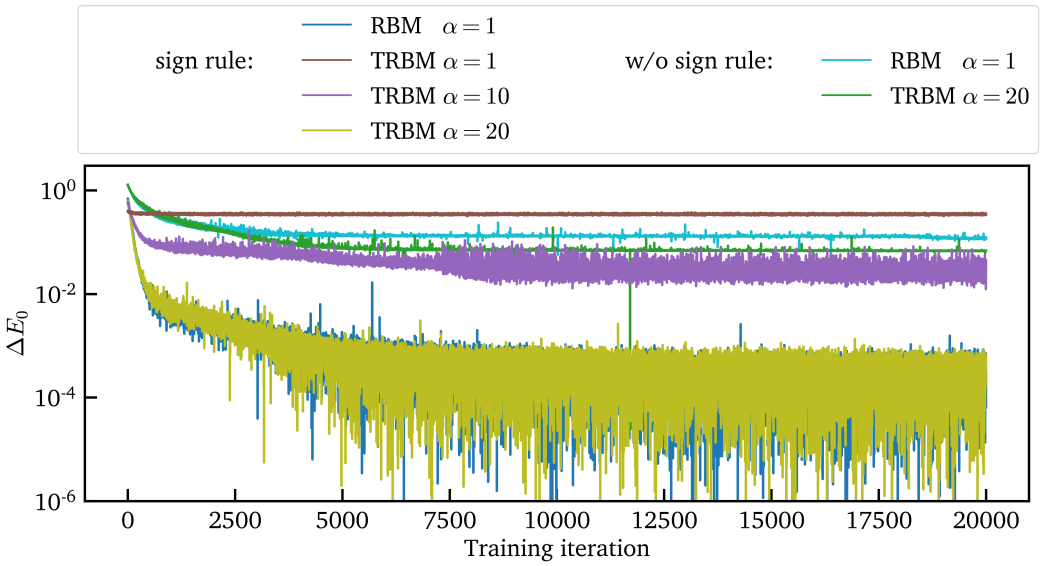



**Fig. 4.3.:** The relative energy error $\Delta E_0$ of the 1D AFH ground state of size $N_v = 22$ with respect to the training iteration for i) the RBM with hidden unit density $\alpha = 1$ and ii) the TRBM that has translational symmetry with $\alpha \in \{1, 10, 20\}$. We investigate the importance of initializing the networks according to Marshall's sign rule. The TRBM $\alpha = 20$ has approximately $10\%$ less parameters than the RBM $\alpha = 1$.

**Expressivity and the coupled basis** The number of variational parameters of the RBM is controlled by the hidden unit density $\alpha = N_h/N_v$ (section 3.1). For the RNN, this number depends on the hidden vector size $d_h$ (also called the number of memory units) and the number of layers $n_l$ (section 3.2). We investigate the relative energy error of the ground state $\Delta E_0$ (Eq. (4.6)) and the Hamiltonian variance $\mathrm{Var}(\hat{H})$ (Eq. (3.29)) when increasing the number of variational parameters. In this context, we demonstrate the advantage of using the coupled basis.

We treat a system of size $N_v = 22$ with open boundary conditions and compare our results with those of Ref. [54]. The results of the RBM are shown in figure 4.4. We see that both the energy error and the variance systematically decrease with increasing network complexity. In the $s_z$-basis, the energy error decreases from $\Delta E_0 \sim \mathcal{O}(10^{-2})$ $(\alpha = 0.5)$ down to $\Delta E_0 \sim \mathcal{O}(10^{-5})$ $(\alpha = 4)$ and then starts to settle. Moreover, the RBM in the coupled basis consistently outperforms the RBM in the standard $s_z$-basis, especially for low network complexities. The smallest network in the coupled basis has a relative energy error $\Delta E_0 \sim \mathcal{O}(10^{-4})$ $(\alpha = 0.5)$, and the obtained energies are more accurate throughout the entire range of $\alpha \in [0.5, 1, 2, 4]$. Similar trends are observed for the Hamiltonian variance $\mathrm{Var}(\hat{H})$. The obtained energies and variances are comparable to those obtained in the reference study [54]. These results indicate that the implementation of SU(2) symmetry is advantageous for obtaining accurate ground states, especially if the network complexity is low.

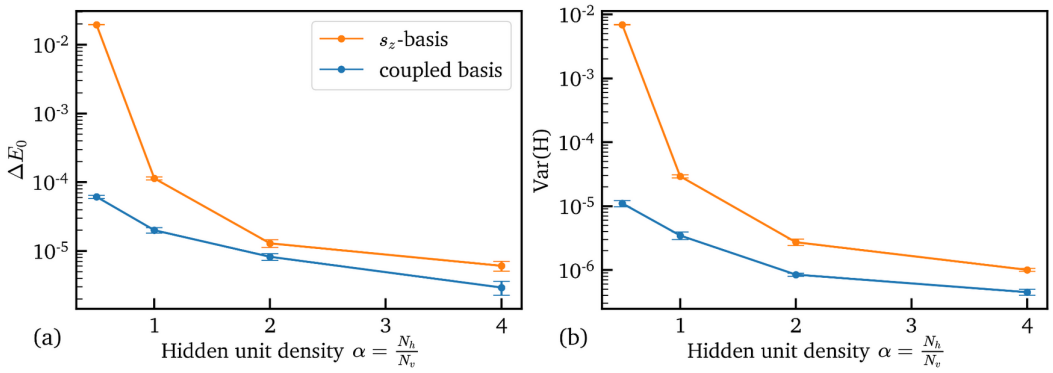



**Fig. 4.4.:** Relative energy errors $\Delta E_0$ and Hamiltonian variances $\mathrm{Var}(\hat{H})$ of the 1D AFH ground state with $N_v = 22$ lattice sites and open boundary conditions. The expressivity of the RBM can be systematically increased by increasing the hidden unit density $\alpha$. The accuracy measures are consistently better when using the coupled basis. Results are comparable to those in Ref. [54].

The RNN has previously been used to study quantum many-body problems [36], and it was shown that the RNN represents ground states faithfully. Here, we combine the RNN with the coupled basis and investigate the expressivity. We again focus on the AFH chain of length $N_v = 22$ with open boundary conditions.

The results of the RNN are shown in Fig. 4.5. We start by fixing the number of layers $n_l = 1$ (Fig. 4.5(a)), and observe that increasing the number of memory units from $d_h = 16$ to $d_h = 32$ leads to an increase in accuracy from $\Delta E_0 \sim \mathcal{O}(10^{-5})$ to $\Delta E_0 \sim \mathcal{O}(10^{-6})$. Further increasing $d_h$ has a minor effect on the ground-state energy and the Hamiltonian variance. This indicates that relatively small RNN networks (concerning $n_{\text{par}}$) are sufficiently complex to represent the AFH ground state accurately. This result is desirable, since complex networks are harder to optimize and require more computational resources. Next, we fix the number of memory units $d_h = 32$ and vary the number of layers $n_l \in [1, 4]$ (Fig. 4.5(b)). Surprisingly, increasing the number of layers does not increase the ground-state accuracy. The measures even become worse for the network with $n_l = 4$, which shows that an increase in network complexity does not necessarily lead to a better ground-state accuracy. This indicates that relatively simple (or rather shallow) networks are sufficiently complex to capture (most of) the correlations in the wave function. A further increase in network complexity unnecessarily complicates the parameter space, making it harder for the optimizer to find the optimal solution. Another explanation is that the additional parameters do not capture new relevant information, and therefore contribute mainly in the form of noise.

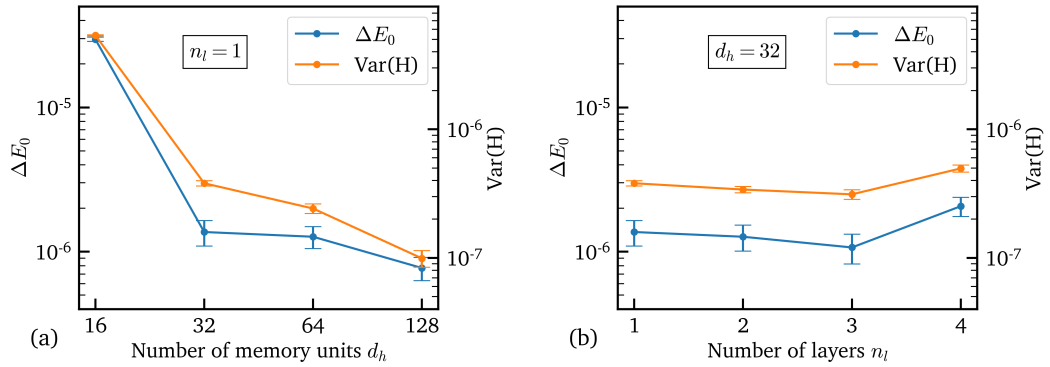



**Fig. 4.5.:** The relative energy error $\Delta E_0$ and Hamiltonian variance $\mathrm{Var}(\hat{H})$ of the 1D AFH ground state using the RNN in the coupled basis. (a) Increasing the number of memory units $d_h$ systematically increases the expressivity of the RNN. (b) Increasing the number of layers $n_l$ does not increase the accuracy measures.

It should be noted that in Ref. [36], a graph similar to Fig. 4.5(b) was obtained for the Hamiltonian variance with respect to number of layers. However, the authors of [36] fixed the total number of variational parameters $n_{\text{par}}$ by decreasing $d_h$ when increasing $n_l$. From their results, one could conclude that the way in which additional parameters are added to the RNN (increasing $d_h$ versus increasing $n_l$) is irrelevant, one finds the same gain in expressivity. This fixing of $n_{\text{par}}$ was not done in our analysis, so we reach very different conclusions and should be skeptical of our results. To ensure that our roughly constant accuracy is not due to numerical precision, the runs were redone with the float64 format. The obtained energies were almost identical, with differences of order $10^{-6}$ or less.

Note that the relative energy errors obtained by the RNN $\Delta E_0 \sim \mathcal{O}(10^{-6})$ are substantially lower than those obtained by the RBM $\Delta E_0 \sim \mathcal{O}(10^{-5})$. The Hamiltonian variances are likewise roughly one order of magnitude lower. This illustrates the expressive power of RNNs, especially when combined with the coupled basis. It has been shown [86] that the total spin is not conserved when using the standard $s_z$-basis, whereas the coupled basis ensures this by construction. As will be discussed in section 4.2.2, spin-spin correlation functions are intrinsically unbiased in the coupled basis [54].

The number of variational parameters of the RNN is independent of system size, which is not the case for the RBM. Another advantage of the RNN is its autoregressive property, which allows for autoregressive sampling (section 3.2.2). Therefore, the sampling time of the RNN scales linearly with system size, hence obtaining samples is no bottleneck when considering large systems $(N_v \approx 100)$. This is in contrast to the Markov chain approach used for energy-based methods such as the RBM

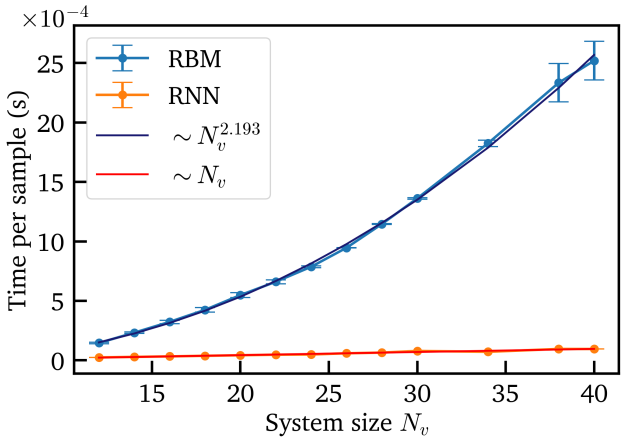



**Fig. 4.6.:** Average sampling time per sample for the RNN ($d_h = 32$ and $n_l = 1$) and the RBM ($\alpha = 1$) approximating the 1D AFH ground state in the coupled basis. The RNN sampling time increases linearly (red fit: $0.0261 N_v - 0.0894$) because of its autoregressive sampling property and the fact that its number of variational parameters is independent of system size. The RBM does not have these properties. Here, the average sampling time is fitted to a power law (blue fit: $0.008 N_v^{2.193} - 0.351$).

(section 3.1.3). In figure 4.6 we show the average sampling time per sample of the RNN ($d_h = 32$ and $n_l = 1$) and RBM ($\alpha = 1$) for system sizes $N_v \in [12, 40]$ in the coupled basis. For the RNN, the sampling time per sample is of order $10^{-5}$ seconds for the smallest system $N_v = 12$ and increases to $\mathcal{O}(10^{-4})$ seconds for the largest system $N_v = 40$. We see that the sampling time scales linearly with system size. The RBM sampling times are non-linear and become prohibitively large systems. Here, the average sampling time per sample is of order $10^{-4}$ seconds for $N_v = 12$ and increases to $2.5 \times 10^{-3}$ seconds for $N_v = 40$. However, it should be noted that the majority of the CPU-time is used for the calculation of energy gradients in the variational Monte Carlo setting (Eqs. 3.14 and 3.25).

**Basis cut-off and big systems** In the coupled basis, the intermediate angular momenta $j_{i \in \{1,2,\ldots,N_v-1\}}$ take values $j_i \in \{0, 1/2, 1, \ldots\}$ that satisfy the triangle relations Eq. (3.48). Since we restrict ourselves to the singlet sector $J = 0$, the triangle relations impose a maximal value $j_{max}$ on the intermediate angular momenta. For example, for a system of $N_v = 4$ spins and $J = 0$, the highest intermediate momentum is obtained by the state $|1/2\, 1\, 1/2\, 0\rangle$ with the highest value $j_{max} = 1$. This value $j_{max}$ is proportional with system size $N_v$. Both for the RBM and the RNN, the number of variational parameters increases linearly with $j_{max}$. In section 4.1.1, we showed that the lowest energy eigenstate of an individual term of $\hat{H}_{AFH}$ is the state where the two spins couple to a singlet. Therefore, it is expected that the low-energy eigenstates of the AFH model involve coupling of spins to low angular momenta.

This motivates the use of a cut-off $j_{cut}$, which is the (self-imposed) maximal value of the intermediate angular momenta $j_i$. In essence, the introduction of $j_{cut}$ is a Hilbert space truncation, which simultaneously influences the model's complexity. Using exact diagonalization, Ref. [54] showed that $j_{cut} = 3$ settles the relative energy error to machine precision. Moreover, the result was largely independent of the investigated system sizes $N_v \in [10, 22]$. We do a similar analysis for the $J_1 - J_2$ model in section 4.2.

In the next experiment, we approximate the AFH ground state of a relatively large system $N_v = 100$ using the RNN in the coupled basis. The RNN has a hidden vector size $d_h = 32$ and a single layer $n_l = 1$, and we use $N_s = 1000$ samples per iteration. We investigate the dependence of the relative energy error $\Delta E_0$ and the Hamiltonian variance $\mathrm{Var}(\hat{H})$ on the cut-off $j_{cut}$. We see in figure 4.7(b) that the RNN can adequately approximate the AFH ground state with $N_v = 100$ for $j_{cut} = 2$. This is shown by the relative energy error $\Delta E_0 \sim \mathcal{O}(10^{-4})$ and the Hamiltonian variance $\mathrm{Var}(\hat{H}) \sim \mathcal{O}(10^{-6})$. The differences in accuracy are relatively small throughout the entire range $j_{cut} \in [2, 10]$, with an outlier at $j_{cut} = 6$. This outlier is an excellent example of the model getting stuck in a local minimum during optimization. This minimum is expected to correspond to an excited state. Namely, the Hamiltonian variance is lower than those of the other data points, yet the relative energy error is significantly higher. This behaviour can also be seen in the training history (Fig. 4.7(a)): the energy error decreases up to a given point ($\approx$ iteration $1500$), suddenly $\mathrm{Var}(\hat{H})$ drops from $3 \times 10^{-4}$ to $10^{-6}$ and the energy error fluctuates around $\Delta E_0 \approx 5 \times 10^{-3}$. A training history with smooth convergence looks like the one in Fig. 4.7(c). The fluctuations are due to the stochastic nature of the variational Monte Carlo approach Eq. (3.8).

The result that the accuracy measures are largely independent of $j_{cut}$ is highly desirable. This indicates that: i) the states with high intermediate angular momentum ($j_i \geq 2$) contribute little to the AFH ground state, and the Hilbert space can thus be truncated without losing too much information and; ii) the network complexity of the RNN can be controlled when probing different system sizes $N_v$, since we fix $j_{max} = j_{cut}$. The relative energy error $\Delta E_0 \approx 2 \times 10^{-4}$ is acceptable, also for the lowest value $j_{cut} = 2$. Since a relatively small network was chosen for these experiments, we find that the RNN can represent the ground state efficiently, using $4074$ variational parameters for $j_{cut} = 2$. The results are in line with previous conclusions that the performance of the network does not necessarily increase with increasing network complexity.

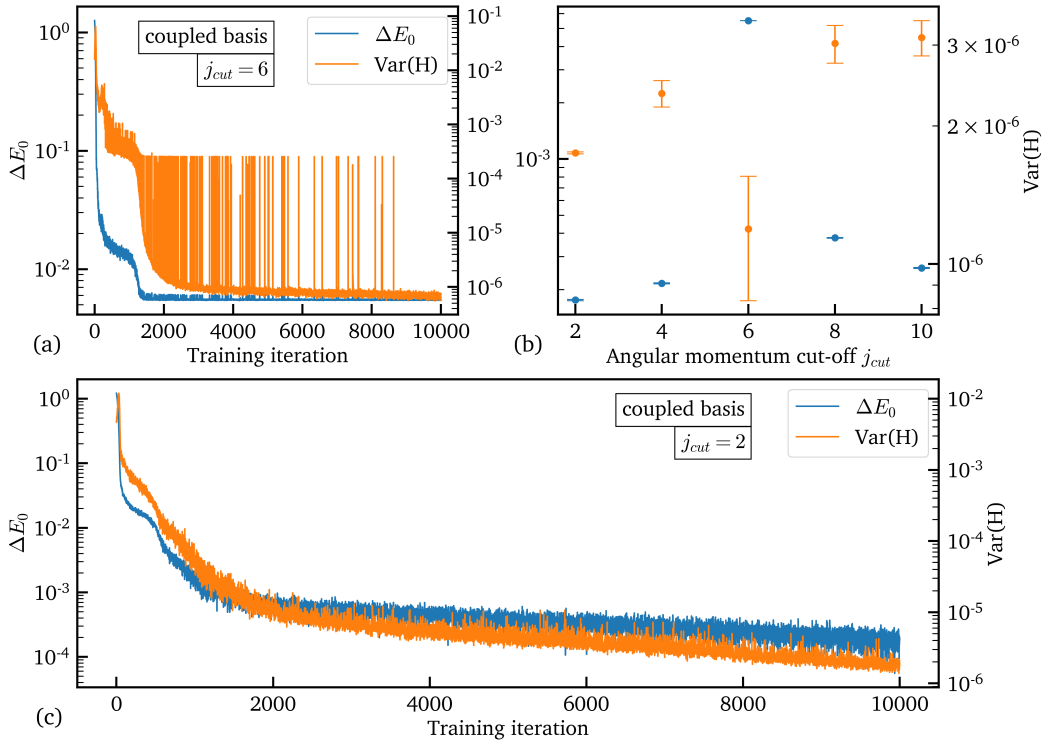



**Fig. 4.7.:** The RNN with a number of memory units $d_h = 32$ and a single layer $n_l = 1$ approximating the 1D AFH ground state with system size $N_v = 100$ in the coupled basis. We investigate the relative energy error $\Delta E_0$ and Hamiltonian variance $\mathrm{Var}(\hat{H})$ and their dependence on the angular momentum cut-off $j_{cut}$. (a) The RNN gets stuck in a local minimum for $j_{cut} = 6$. (b) Increasing the cut-off $j_{cut}$ does not increase the accuracy measures: the relative energy errors remain around $\Delta E_0 \approx 2 \times 10^{-4}$. (c) A smooth convergence in the training for $j_{cut} = 2$.

**Excited states and energy gap** Up until now, we have used the coupled basis to approximate the ground state of the 1D AFH model by optimizing the variational ansätze in the subspace defined by $|J = 0\, M_J = 0\rangle$. The first excited state of the AFH model is in the subspace $|J = 1\, M_J \in \{-1, 0, 1\}\rangle$. This state can thus be targeted in the coupled basis, which enables us to compute the energy gap $E_{\text{gap}} = E_1 - E_0$, where $E_1$ is the energy of the first excited state. We note that the first excited state can be approximated by NQSs without making use of SU(2) symmetry [87]. Since the 1D AFH model is gapless (section 4.1.1), the gap vanishes as $E_{\text{gap}} \propto N_v^{-1}$ for $N_v \to \infty$. We construct the ground state and first excited state for system sizes $N_v \in [12, 40]$, compute $E_{\text{gap}}$ at each size, and compare our results to Refs. [54, 87].

In figure 4.8, we show the relative energy errors of the ground state $\Delta E_0$ and the first excited state $\Delta E_1$ for system sizes $N_v \in [12, 40]$. The Lanczos diagonalization algorithm (section 3.4.1) becomes intractable for $N_v \geq 30$. We compare our energies with those obtained by the density matrix renormalization group method (DMRG, section 3.4.2). We find that both the RBM ($\alpha = 1$) and the RNN ($d_h = 32$ and $n_l = 1$) can adequately reproduce the gap $E_{\text{gap}}$ in the entire range, apart from a few outliers. Considering the RBM, the relative energy error of the ground state is $\Delta E_0 \sim \mathcal{O}(10^{-5})$, and the error becomes larger when increasing the system size $N_v$. Different behaviour is observed for the RNN: the error $\Delta E_0 \sim \mathcal{O}(10^{-5})$ for $N_v \in [14, 20]$, and it then decreases $\Delta E_0 \in [10^{-7}, 10^{-6}]$ for sizes $N_v \geq 26$. The relative energy error of the first excited state $\Delta E_1 \sim \mathcal{O}(10^{-5}) - \mathcal{O}(10^{-4})$ is generally higher than $\Delta E_0$, and increases with increasing system size for both models. The trend of $\Delta E_1$ is seemingly more stable than that of $\Delta E_0$. We conclude that both models can adequately reproduce the ground state and first excited states for small systems. The RNN is has smaller errors when considering systems of size $N_v \geq 20$.

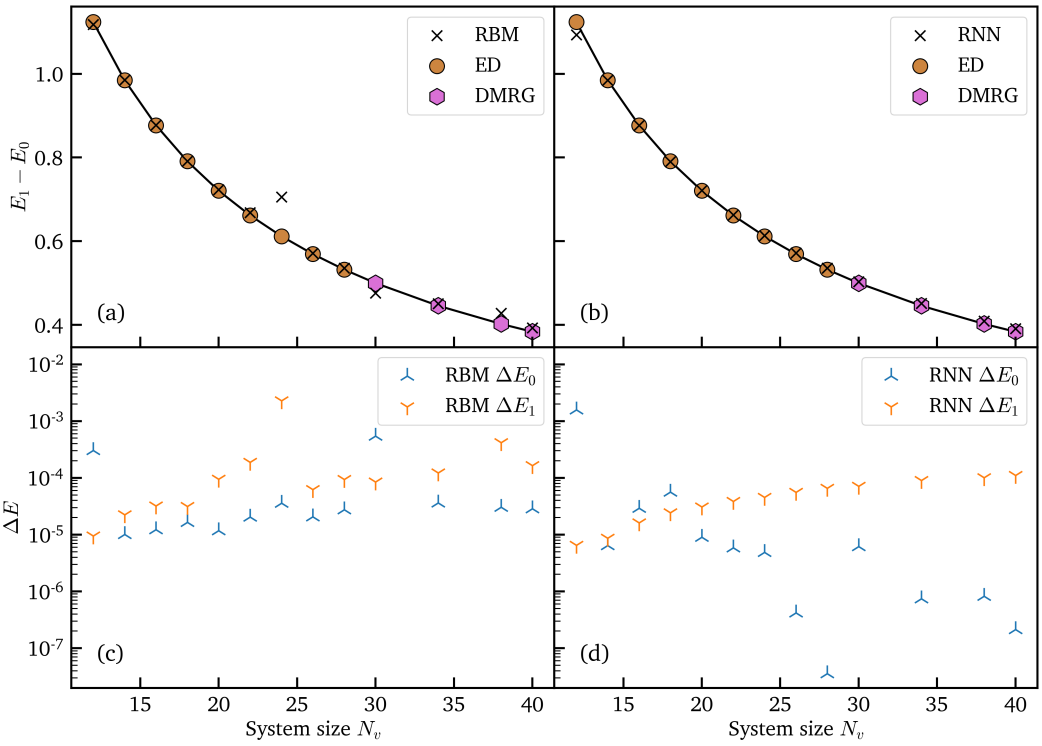



**Fig. 4.8.:** The energy gap $E_{\text{gap}} = E_1 - E_0$ between the first excited state and the ground state of the 1D AFH model for various system sizes $N_v$ and open boundary conditions (a-b), together with the relative energy errors $\Delta E_0$ and $\Delta E_1$ (c-d): (a) and (c) RBM ($\alpha = 1$) results; (b) and (d) RNN ($d_h = 32, n_l = 1$) results. We compare the RBM and RNN energies with those obtained by exact diagonalization (ED) and density matrix renormalization group methods (DMRG).

The results obtained here are almost identical to those of Ref. [54], where the gap was likewise calculated using the coupled basis. Although the authors of Ref. [87] also used an RBM, they obtained the gap using a different strategy. There, a superposition of two networks (in the standard $s_z$-basis) is used to represent a state orthogonal to the ground state. The authors obtained relative errors $\Delta E_1 \lesssim 3 \times 10^{-4}$ for the 1D AFH of size $N_v \in [8, 36]$ with periodic boundary conditions. As can be seen in Fig. 4.8, we obtain similar results using the RBM. The relative error $\Delta E_1$ of the RNN does not exceed $10^{-4}$ in the entire range $N_v \in [12, 40]$.

It is worth mentioning the outliers in Fig. 4.8 and their nature. Generally speaking, optimizing neural networks can be tricky. Some aspects of the optimization can be controlled by carefully selecting hyperparameters, or by properly initializing the weights (section 2.3.3). Even with these precautions, the model might get stuck in a local minimum of the parameter space (for example an excited state, as was the case in Fig. 4.7(a)). Often, repeating the simulation with a different seed for the random number generators makes the simulation find the proper minimum. However, this can be time-consuming and requires extra computational resources. A way to partially circumvent this is by intermittently storing the model. After completion of the simulation, one finds at which iteration of the optimization the model got stuck. Retraining one of the earlier stored models normally does the trick.

In general, the RNN required few interventions, whereas the RBM was prone to difficulties. Also, the RNN is more likely to get stuck in local minima if the system size is small. These findings are reflected by the results of Fig. 4.8, where each of the data points was obtained by doing a single simulation. By retraining the models we expect the outliers to disappear, i.e., there is nothing physically distinct about these points (e.g. the RBM point $N_v = 24$ in Fig. 4.8). This is not always the case: the accuracy at a critical point is expected to be relatively low compared to non-critical points (for NQSs this is illustrated in [86]).

**Wave function structure** An accurate variational energy is an indicator for a good wave function representation. However, it is highly desirable that the expectation values of other observables (e.g. spin-spin correlation functions) can also be predicted accurately. For this reason, we are interested in the structure of the wave function. The structure is investigated by looking at the importance of the basis states. The importance of a state with index $j$ is measured by the squared modulus of its expansion coefficient $|\psi_j|^2$, relative to the largest squared modulus $|\psi_0|^2$. The importance of state $j$ is thus $|\psi_j|^2 \,/\, |\psi_0|^2$.

We inspect the ability of NQSs to reproduce the wave function structure as follows. An RBM and RNN are optimized to represent the ground state of the 1D AFH chain of length $N_v = 22$ with open boundary conditions. This is done in the coupled basis with $j_{cut} = 2$. The exact ground state is calculated using the Lanczos algorithm (section 3.4.1). We thus have to our disposal the (exact) complex coefficients $\psi_j$ and the corresponding basis states. Then, we calculate the squared modulus $|\psi_j|^2$ and sort them from largest to smallest, such that $|\psi_0|^2 \geq |\psi_1|^2 \geq \dots$. Subsequently, we normalize the squared modulus by dividing each of them by the largest one $|\psi_0|^2$. This is repeated for the squared modulus of the RBM and the RNN.

The $11$ most important configurations $|j_1, j_2, \dots, j_{N_v-1}; J = 0\, M_J = 0\rangle$ of the 1D AFH ground state are shown in figure 4.9. The importance measure $|\psi_j|^2 / |\psi_0|^2$ is also given. By recoupling the spins, we see that the most important configuration $|\sigma\rangle_0 = |j_0 = 0, j_1 = 1/2, j_3 = 0, \dots\rangle$ is a resonating valence bond state [88], where every two neighbouring spins are coupled to a singlet. The $10$ following states have a background of singlets as before, but with two neighbouring singlets excited to triplets, which couple together to form a singlet. The most important excitations are located near the middle of the chain due to the open boundary conditions. The exact diagonalization result shows that we have reflection symmetry about the center of the chain. This symmetry is not implemented in the neural networks, which gives rise to small discrepancies in the squared modulus. Apart from these discrepancies, the results match well, and the states are correctly ordered according to their importance.

We can also order the states according to the squared modulus of exact diagonalization and subsequently compute the corresponding squared modulus of the NQSs (without ordering them independently). Thus, for a given state, we calculate the squared modulus of the RBM and the RNN, and compare to the exact result. The importance measure $|\psi_j|^2 / |\psi_0|^2$ with respect to ordered index $j$ is shown in figure 4.10. Both the RBM and the RNN approximate the most important $\approx 2000$ expansion coefficients accurately, as there is an excellent agreement with exact diagonalization (note the logarithmic scale). The relative deviations become more pronounced for the less important ($|\psi_j|^2 / |\psi_0|^2 \lesssim 10^{-6}$) squared moduli. In general, the squared moduli of the RBM are less accurate than those of the RNN.

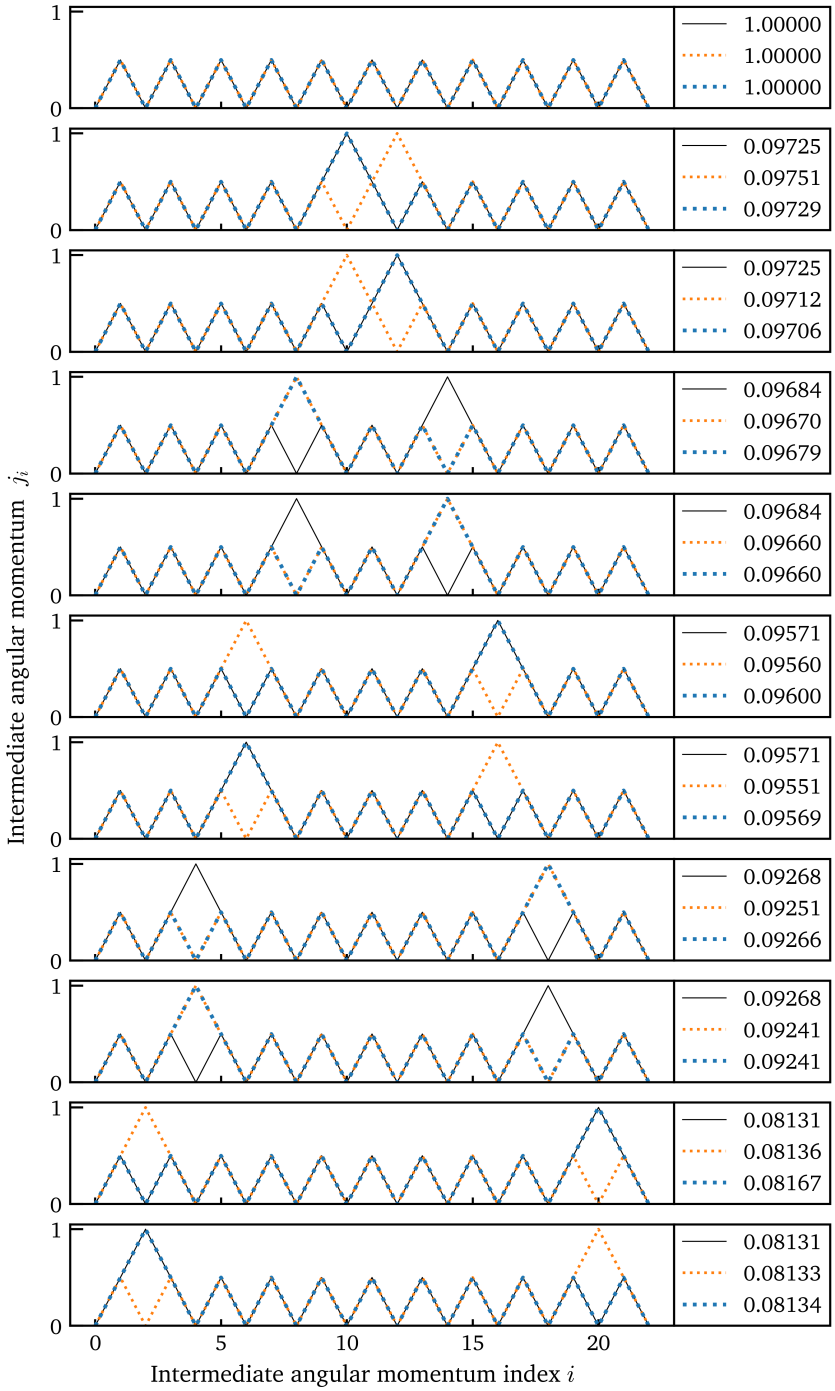



**Fig. 4.9.:** The 11 most important configurations of the ground state of the 1D AFH chain of length $N_v = 22$ with open boundary conditions. We compare the relative squared modulus $|\psi_j|^2 / |\psi_0|^2$ obtained by exact diagonalization (black), the RBM (orange), and the RNN (blue). The squared modulus of the wave functions are ordered according to their importance, and the corresponding states with intermediate angular momenta $j_i$ are shown.

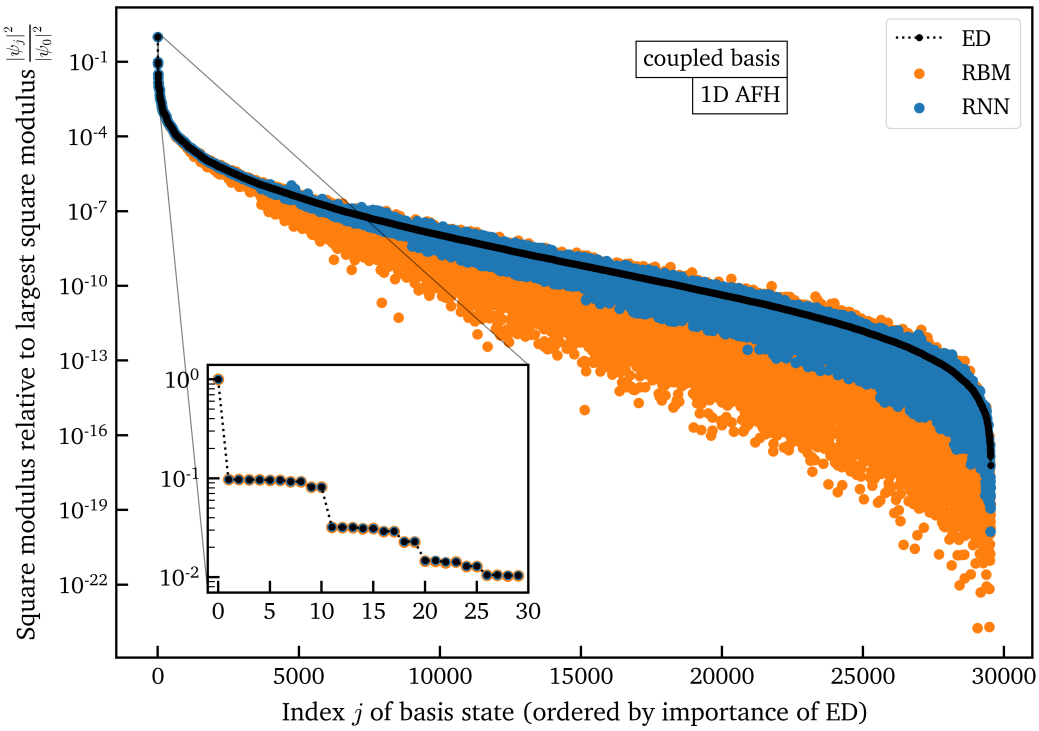



**Fig. 4.10.:** The squared modulus relative to the largest squared modulus $|\psi_j|^2 / |\psi_0|^2$ of the 1D AFH ground state with $N_v = 22$ (coupled basis). The basis states are ordered in descending fashion according to the squared modulus obtained by exact diagonalization (ED). The inset shows the 30 most important states.

## 4.2 The $J_1 - J_2$ model

The $J_1 - J_2$ model adds to the AFH model (section 4.1.1) next-to-nearest neighbour interactions. The model is defined by the Hamiltonian

$$\hat{H}_{J_1-J_2} = J_1 \sum_{\langle i,j \rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j, \tag{4.7}$$

where nearest neighbours $\langle i, j \rangle$ interact with interaction strength $J_1$ and next-to-nearest neighbours $\langle\langle i, j \rangle\rangle$ interact with strength $J_2$. We restrict ourselves to antiferromagnetic interactions, i.e. $J_1 > 0$ and $J_2 > 0$. It is customary to put $J_1 = 1$, such that the value of $J_2$ reflects the ratio $J_2/J_1$. We consider the case of spin-1/2.

### 4.2.1 Background theory

The $J_1 - J_2$ model reduces to the AFH model (section 4.1.1) for $J_2 = 0$. The model has SU(2) symmetry and the ground state is known to have $J = 0$. The introduction of the next-to-nearest neighbour interaction leads to frustration at the quantum level (on top of the frustration due to the non-commutativity of operators $\hat{\boldsymbol{\sigma}}_i$). Frustrated systems are widely studied in modern literature, as the exact phase diagram of the 2D $J_1 - J_2$ model is unknown (although numerical results have been found [84, 86, 89–95]). Also of interest is the case of lattice geometries that additionally involve geometric frustration, e.g. the triangular lattice [93].

For the square lattice, a tensor network approach using the entangled-plaquette state (EPS) ansatz indicates Néel order for $J_2 \lesssim 0.5$ and stripe order for $J_2 \gtrsim 0.6$ [90]. These findings have already been reproduced by NQS ansätze [86, 91]. For $0.5 \lesssim J_2 \lesssim 0.6$, numerical simulations point towards a spin liquid phase [90], a phase which does not break any symmetries (neither spin rotation nor lattice symmetries). The neural network approach of [91] showed that the system realizes a valence bond solid in this region, which does break lattice symmetry. The model has also been studied with a cylindrical geometry [89]. Due to this geometry, so-called topological sectors arise, which come with a splitting of the energy levels. The splitting vanishes exponentially with increasing perimeter of the cylinder.

The 1D $J_1 - J_2$ chain is gapless up to a critical point $J_{2,c} \approx 0.241$, and is gapped for $J_2 > J_{2,c}$ [92, 94]. At the particular point $J_2 = 0.5$ the chain is known as the Majumdar-Ghosh model (1970), which is exactly solvable [96]. The ground state takes the form of valence bond states (also called dimer states), a superposition of the two possible ways to completely cover the chain by nearest neighbour singlets (this requires periodic boundary conditions and an even number of sites $N_v$). The dimer phase remains present over the whole gapped region $J_2 \gtrsim 0.241$. Beyond the Majumdar-Ghosh point $J_2 > 0.5$, the (short-range) Néel order seizes to exist,[4] and the system shows short-ranged incommensurate spin ("spiral") correlations, also called the frustrated regime [92]. Further increasing the next-to-nearest neighbour interaction strength such that $J_2 \gg J_1$ leads to two decoupled spin chains [95]. The Marshall's sign rule (section 4.1.1, Eq. (4.4)) is preserved for $J_2 < 0 < J_1$ (requires a bipartite lattice). However, for positive $J_1$ and $J_2$ it is not generally true. For 1D, studies suggest that the breakdown occurs near the Heisenberg point $J_2 \approx 0.032$ (when $J_1 = 1$) [84]. Finite-size lattice calculations have shown that the two-dimensional $J_1 - J_2$ model may preserve the sign rule up to $0.2 \lesssim J_2 \lesssim 0.3$ [84].

[4] Recall that long-range order in the limit $N_v \to \infty$ cannot exist (even at $T = 0$) in 1D (sec. 4.1.1).

### 4.2.2 Results

**The angular momentum cut-off revisited** We start with the $J_1 - J_2$ chain, i.e. the one-dimensional model. Before we do any optimizations with the RBM or RNN ansätze, we investigate the basis cut-off by exact diagonalization. Since the next-to-nearest neighbour interactions introduce frustration, we expect that the wave function structure of the ground state strongly depends on the interaction strength $J_2$. Therefore, we investigate $j_{cut}$ as follows. For a given range of interaction strengths $J_2 \in [0, 2.0]$ and $N_v = 22$, we calculate the exact ground-state energy by exact diagonalization (section 3.4.1) using i) the complete basis which gives $E_{exact}$ and ii) the truncated basis with a selected $j_{cut} \in [1, 5]$, which gives an energy $E_{cut}$. The accuracy measure is defined by

$$\Delta E_{cut}(J_2, j_{cut}) = \left| \frac{E_{\text{exact}}(J_2) - E_{cut}(J_2, j_{cut})}{E_{\text{exact}}(J_2)} \right|, \tag{4.8}$$

where all energies are of the ground state. In this way, one can find the dependencies between the energy $E_{cut}$, the interaction strength $J_2$, and the cut-off angular momentum $j_{cut}$. Note that the relative errors $\Delta E_{cut}(J_2, j_{cut})$ are entirely due to a Hilbert space truncation.

The results of this analysis are visualized in figure 4.11. First, we observe that for $J_2 = 0$ (the AFH point) a cut-off $j_{cut} = 3$ is indeed sufficient to reach numerical precision, as was shown in [54]. The cut-off we used in our experiments of the AFH chain (section 4.1.2), namely $j_{cut} = 2$, leads to an exact diagonalization error of order $10^{-8}$. This is more accurate than what we expect to obtain by the NQS ansätze for any $j_{cut}$. In other words, the Hilbert space truncation error $\Delta E_{cut} \sim \mathcal{O}(10^{-8})$ is negligible compared to the variational error $\Delta E_0 \sim \mathcal{O}(10^{-6})$. Thus, it was indeed justified to use $j_{cut} = 2$ in our experiments of the AFH chain.

Interestingly, the accuracy $\Delta E_{\text{cut}}(J_2|j_{cut})$ at any given $j_{cut}$ increases with increasing $J_2$ in the range $J_2 \in [0, 0.5]$. At the Majumdar-Ghosh point $J_2 = 0.5$, the exact diagonalization ground-state energy at any cut-off $j_{cut} \in [1, 5]$ reaches numerical precision. This reflects the nature of the valence bond ground state at this particular point. When neighbouring spins couple to singlets, the largest intermediate angular momentum in the coupled basis is $j_{max,MG} = 1/2$. There is thus no error due to Hilbert space truncation for $j_{cut} \geq 1/2$. Beyond the Majumdar-Ghosh point, i.e. if $J_2 > 0.5$, the accuracy $\Delta E_{cut}$ decreases with increasing $J_2$. This can also be interpreted physically: because of the frustration, spins are less inclined to form singlets with their neighbours, leading to an increased importance of states with higher intermediate angular momentum $j_i$. This effect becomes more pronounced

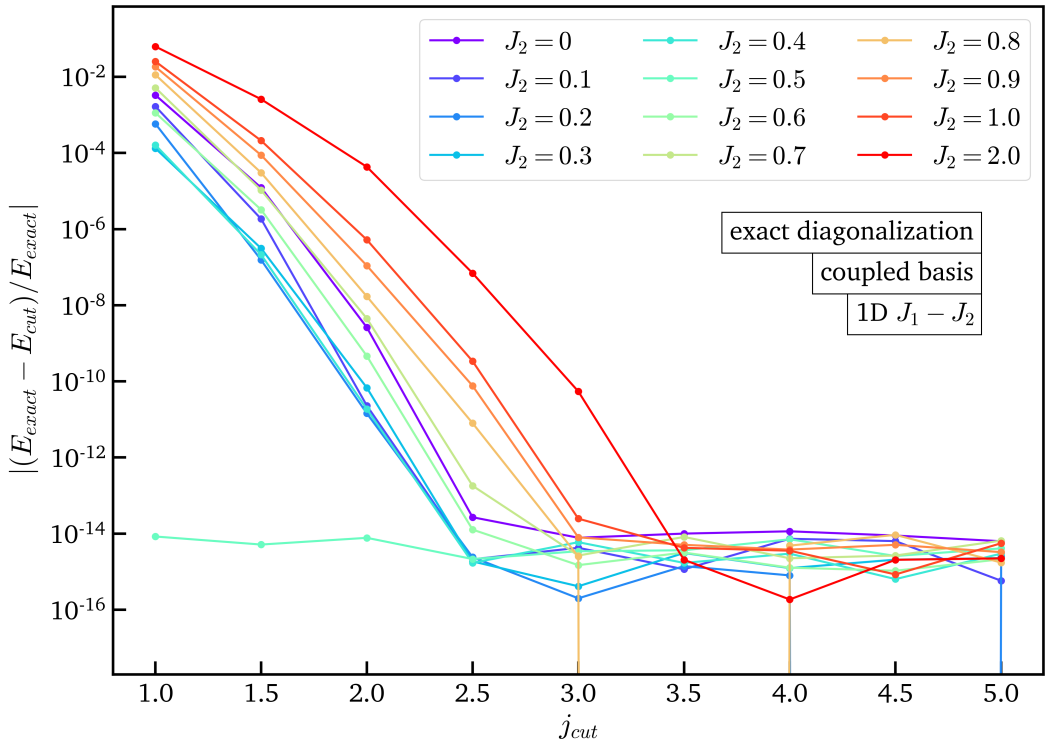



**Fig. 4.11.:** The truncation error $\Delta E_{\text{cut}}(J_2 \,|\, j_{cut}) = |(E_{\text{exact}}(J_2) - E_{cut}(J_2, j_{cut}))/E_{\text{exact}}(J_2)|$ versus angular momentum cut-off $j_{cut}$ for various next-to-nearest neighbour interaction strengths $J_2$ of the $J_1 - J_2$ chain of length $N_v = 22$. Numerical precision is obtained for $j_{cut} = 3$ in the range $J_2 \in [0, 1]$.

for larger $J_2$. Note that the most interesting region to explore is $J_2 \in [0, 1]$. An adequate trade-off between Hilbert space truncation and network complexity is provided for $j_{cut} = 3$.

**Ground-state energy** We now investigate the ability of the NQSs to represent the ground state of the $J_1 - J_2$ model for $J_2/J_1 \in [0, 1]$. Two important questions are posed: i) for which values of $J_2$ is the sign structure initialization Eq. (4.4) advantageous, knowing that the sign rule breaks down near the Heisenberg point $J_2 \approx 0.032$; ii) does the higher accuracy in the coupled basis persist throughout the range $J_2 \in [0, 1]$. We answer these questions by optimizing the RBM in the $s_z$-basis with and without the sign rule initialization. The ground-state energies of these optimizations are then compared to the those obtained by the RBM and RNN in the coupled basis with $j_{cut} = 3$.

Figure 4.12 shows the ground-state energy $E_0$ and the relative energy error $\Delta E_0$ for the $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions. The ground-state energy becomes higher with increasing $J_2$ for $J_2 \lesssim 0.5$, after which it decreases. To answer our first question i): the results suggest that the sign rule initialization

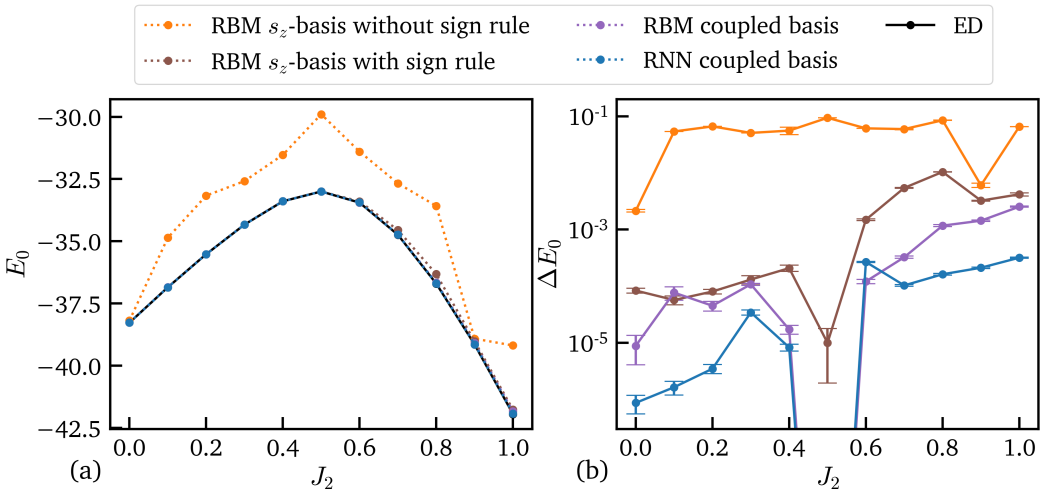



**Fig. 4.12.:** Comparison of different NQS ansätze for the ground state representation for various $J_2$ of the $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions. The RBM in the standard $s_z$-basis is tested with and without the sign rule initialization. The RBM and RNN are used as ansatz in the coupled basis. We use the exact diagonalization (ED) result as reference. (a) The ground-state energy $E_0$ with respect to $J_2$. (b) The relative energy error $\Delta E_0$ with respect to $J_2$.

is required for adequate optimization throughout the entire range $J_2 \in [0, 1]$. For all of the selected values of $J_2$, the relative energy error without using the sign rule initialization is $\Delta E_0 \approx 10^{-1}$ $(10\%)$. Notable exceptions are the Heisenberg point $J_2 = 0$ at which $\Delta E_0 \approx 2 \times 10^{-3}$, and the point $J_2 = 0.9$ with $\Delta E_0 \approx 8 \times 10^{-3}$. The energies we obtain with a sign rule initialization are remarkably better: in the range $J_2 \in [0, 0.5[$ the relative energy errors are of order $\Delta E_0 \approx 10^{-4}$. However, the variational optimization is clearly more challenging in the frustrated regime, where we have errors $\Delta E_0 \in [10^{-3}, 10^{-2}]$. Considering that the sign rule breaks down close to the Heisenberg point, these findings are surprising. One may argue that the sign rule applies approximately for $J_2 \approx 0$, and that for $J_2 \approx 1$ the model is able to learn the correct signs more easily if the sign structure of Eq. (4.4) is imposed, even though this initial structure is not entirely correct.

We can learn more by looking at how the relative energy error $\Delta E_0$ progresses during training. In figure 4.13, we show the training histories of the points $J_2 = 0.4$ and $J_2 = 0.9$, both with and without the sign rule initialization Eq. (4.4). The first observation is that the convergence takes more iterations when the system is in the frustrated phase, i.e. the RBM needs more training steps to converge for $J_2 = 0.9$ ($\approx$ 7500 steps) than for $J_2 = 0.4$ ($\approx$ 1500 steps). The appearance of a steady variational energy with high relative error $(\Delta E_0 \approx 10^{-1})$ shows that it is indeed challenging for the RBM ansatz to learn the ground state if the sign structure is not manually imposed. However, something unexpected happens for

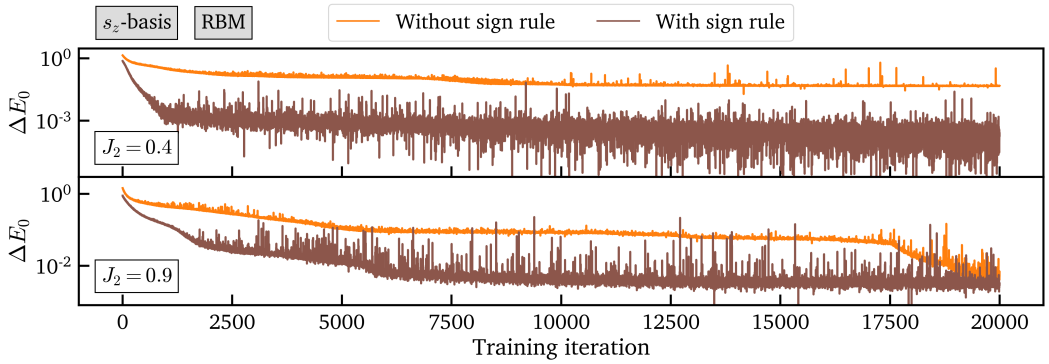



**Fig. 4.13.:** The relative energy error of the ground state $\Delta E_0$ with respect to training iteration for the RBM in the standard $s_z$-basis with and without sign rule initialization. The system under consideration is a $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions. The training histories are plotted for the two points $J_2 = 0.4$ and $J_2 = 0.9$.

$J_2 = 0.9$: the energy remains steady during steps $5000 - 17500$, but then suddenly decreases. This implies that the RBM without a proper sign initialization can in practice learn the correct sign structure (which in principle is made possible by the complex amplitudes in the RBM ansatz Eq. (2.6)). Since $J_2 = 0.9$ is the only $J_2$-point at which this transition occurs, we speculate that this transition is highly unlikely.

Perhaps the most important information to extract from figure 4.12 is that the relative energy error $\Delta E_0$ can be multiple orders of magnitude lower in the coupled basis. Focusing on the RBMs, the ground-state energies in the ordered phase $J_2 < 0.5$ are comparable, although lower in the coupled basis. The difference becomes more significant in the frustrated regime $J_2 > 0.5$, with the $s_z$-basis having relative energy errors $\Delta E_0 \sim \mathcal{O}(10^{-2}) - \mathcal{O}(10^{-3})$, compared to the coupled basis with $\Delta E_0 \sim \mathcal{O}(10^{-3}) - \mathcal{O}(10^{-4})$. This indicates the importance of SU(2) invariance in the frustrated phase of the $J_1 - J_2$ chain. Moreover, the Majumdar-Ghosh point $J_2 = 0.5$ is represented exactly in the coupled basis ($\Delta E_0 \approx 10^{-16}$ for RBM and $\Delta E_0 = 0$ for RNN). The RNN performs even better than the RBM, with relative energy errors of up to almost two orders of magnitude lower $\Delta E_0 \sim \mathcal{O}(10^{-4}) - \mathcal{O}(10^{-6})$ in the coupled basis.

One might now ask whether similar accuracy gains can be realized by imposing lattice symmetries. We illustrate that, by considering an RBM with translational symmetry (TRBM), this is not necessarily the case. In figure 4.14, we show the relative energy error $\Delta E_0$ with respect to $J_2$ for $N_v = 22$ with periodic boundary conditions. The variational optimization is done using an RBM ($\alpha = 1$, $n_{\text{par}} = 505$) and a translation invariant TRBM ($\alpha = 20$, $n_{\text{par}} = 461$), both in the $s_z$-basis. The TRBM does not necessarily obtain lower ground-state energies than the standard RBM. A possible

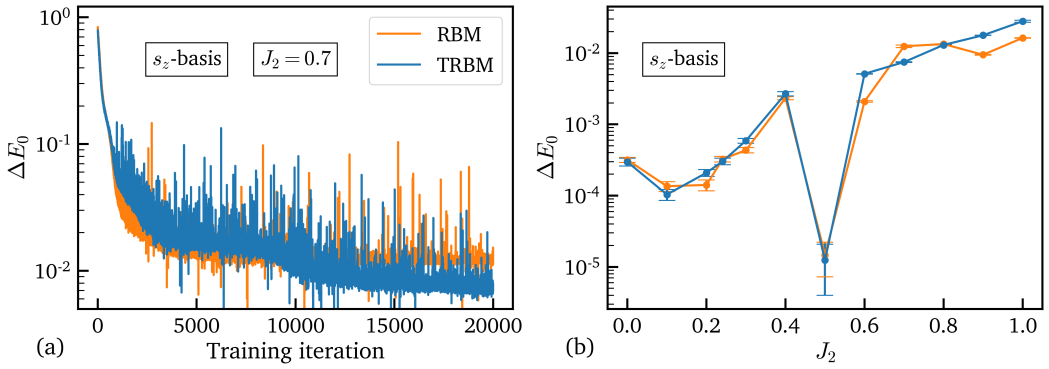



**Fig. 4.14.:** Results obtained by a standard RBM ($\alpha = 1$) and an RBM with translational symmetry (TRBM) ($\alpha = 20$) for the $J_1 - J_2$ chain of length $N_v = 22$ with periodic boundary conditions. Both ansätze are in the standard $s_z$-basis. (a) The relative energy error $\Delta E_0$ with respect to the training iteration for $J_2 = 0.7$. (b) $E_0$ with respect to $J_2 \in [0, 1]$.

explanation is that the RBM learns the translation invariance accurately, such that the symmetry does not need to be implemented in an exact fashion. This is in line with the relatively small deviations of reflection symmetry found in Fig. 4.9. However, symmetries are often implemented to accelerate the convergence. We find that such a speed-up is not guaranteed: at the point $J_2 = 0.7$, the TRBM seems to be stuck in a local minimum between training iterations $4000 - 9000$. After $\approx 10000$ training steps the TRBM overtakes the RBM, but this did not happen for other $J_2$ (e.g. $J_2 = 0.6$). Even if the TRBM converges to a lower energy than the RBM, the error is relatively high $\Delta E_0 \approx 10^{-2}$ in the frustrated regime $J_2 > 0.5$. These results indicate that the implementation of SU(2) symmetry is more relevant than the implementation of translational lattice symmetry, at least in the range $J_2 \in [0, 1]$. To verify that this is the case, a logical next step is to repeat the simulations with the RNN in the $s_z$-basis.

**Spin-spin correlation function** Perhaps the most relevant observable for studying many-body quantum wave functions is the spin-spin correlation function $C(i, j) = \langle \hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_j \rangle$, whose components are defined as

$$C_{\xi,\zeta}(i, j) = \langle \hat{s}_i^{\xi} \hat{s}_i^{\zeta} \rangle \,, \tag{4.9}$$

with $\xi, \zeta \in \{x, y, z\}$ the directions of the spin operator $\hat{\boldsymbol{s}}$. For systems that satisfy SU(2) symmetry, the components of the correlation function with $\xi = \zeta$ are equal. Thus, in a system with SU(2) symmetry we have that

$$C(i, j) = C_{x,x}(i, j) = C_{y,y}(i, j) = C_{z,z}(i, j), \tag{4.10}$$

where each component is equal to a third of the scalar product $\hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_j$. In the coupled basis, this principle Eq. (4.10) holds by construction, which can be proven using the Wigner-Eckart theorem [54, 55]. For antiferromagnetic systems, it is customary to define the correlation function with an additional prefactor $(-1)^{i-j}$.

We first consider the correlation function at the Heisenberg point $J_2 = 0$. We use the optimized models of Fig. 4.12, and are dealing with a chain of length $N_v = 22$ and open boundary conditions. For the RBM ($\alpha = 1$) and RNN ($d_h = 50, n_l = 1$) in the coupled basis, the correlation function $C(i,j)$ of Eq. (4.10) is computed. For the RBM ($\alpha = 1$) in the standard $s_z$-basis, we compute separately the longitudinal $C^{s_z}_{\parallel}(i,j) = C^{s_z}_{z,z}(i,j)$ and the transverse $C^{s_z}_{\perp}(i,j) = (C^{s_z}_{x,x}(i,j) + C^{s_z}_{x,x}(i,j))/2$ correlation functions. There is no translational symmetry because of the open boundary conditions. Therefore, the correlation function for a given distance $|i-j|$ is obtained by choosing the $(i-j)$-pair such that both sites $i$ and $j$ are equally distant from the center of the chain.

The correlation functions of the NQSs are compared to exact diagonalization in figure 4.15. The power law fit in Fig. 4.15(a) matches the expected quasi long-range order of Eq. (4.5) with an exponent $\gamma = 1.165$, although relatively large deviations appear due to the fact that we consider small distances $|i-j|$, the open boundary conditions, and finite-size effects. The absolute errors of the correlation functions (Fig. 4.15(b)) indicate that the RBM in the $s_z$-basis is biased towards the longitudinal component $C^{s_z}_{\parallel}$. It has been shown that this bias is more significant when the network complexity is lower ($\alpha = 0.5$) [54]. The RBM in the coupled basis leads to a correlation function $C(i,j)$ that is inherently unbiased, and generally it shows smaller errors $\sim \mathcal{O}(10^{-4})$. Surprisingly, apart from the distances $|i-j| \in \{8, 12\}$, the RNN in the coupled basis performed worse than the RBM, with typical errors of order $10^{-3}$ and higher.

Since the RNN has lower variational energies than the RBM (Fig. 4.12, both NQSs in the coupled basis), the results of Fig. 4.15 are unexpected. To explain the inferior correlation functions of the RNN, we calculate the weight $\epsilon$ of excited states in the wave function. The variational wave function can be written as

$$|\Psi_{\mathcal{W}}\rangle = \sqrt{1-\epsilon^2}\,|\Psi_{gs}\rangle + \epsilon\,|\Psi_{\perp}\rangle\,, \tag{4.11}$$

where $|\Psi_{gs}\rangle$ is the exact ground-state wave function and $|\Psi_{\perp}\rangle$ is a normalized state orthogonal to $|\Psi_{gs}\rangle$. The weight $\epsilon$ is a measure for the accuracy of the variational ansatz. Even if there is spurious content in the variational ground state, the corresponding energy may be close to the exact one. This is expected for small energy gaps. When dealing with systems of varying energy gaps, the weight $\epsilon$ is the more

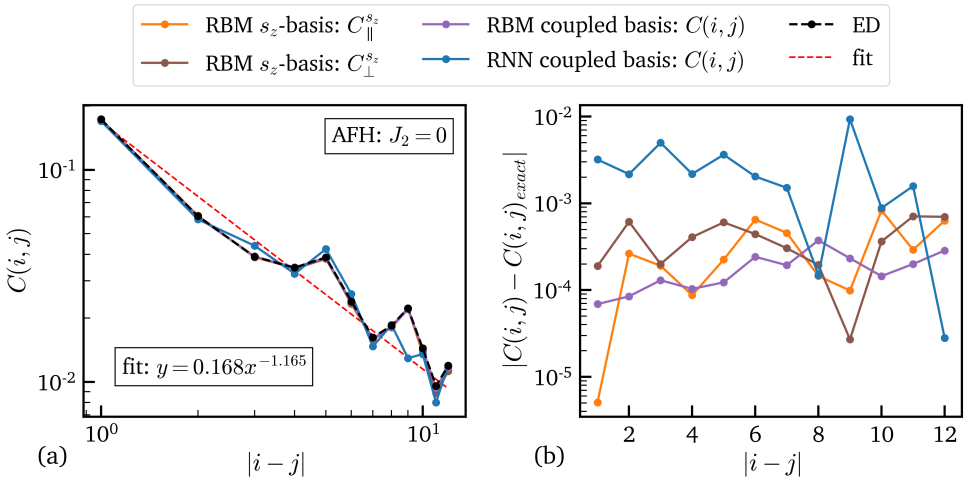



**Fig. 4.15.:** The antiferromagnetic spin-spin correlation function $C(i,j)$ versus $|i-j|$ at the Heisenberg point $J_2 = 0$ for $N_v = 22$ with open boundary conditions. We compare the longitudinal $C^{s_z}_{\parallel}$ and transverse $C^{s_z}_{\parallel}$ correlation functions of the RBM in the $s_z$-basis with the symmetric correlation function $C^{\mathrm{SU(2)}}(i,j)$ obtained by the RBM and RNN in the coupled basis. The exact diagonalization (ED) result $C(i,j)_{\mathrm{exact}}$ is taken as reference. (a) The correlation functions $C(i,j)$ versus distance between spins $|i-j|$. The fit is a power law $y = 0.168x^{-1.165}$. (b) The absolute error of the correlation functions for several distances $|i-j|$.

appropriate accuracy measure. This is also true when one is unsure about the size of the energy gap, since $\Delta E_0$ might give false impressions for small energy gaps. Since we have calculated the exact diagonalization, the weight $\epsilon$ can be computed exactly by the overlap $\langle \Psi_{gs} | \Psi_{\mathcal{W}} \rangle = \sqrt{1-\epsilon^2}$.

The weight of excited states $\epsilon$ with respect to the next-to-nearest interaction strength $J_2$ are shown in Fig. 4.16. We see that the RNN has more spurious content in its wave function than does the RBM (at $J_2 = 0$), even though the RNN has a lower variational energy ($\Delta E_0 \approx 10^{-6}$) than the RBM ($\Delta E_0 \approx 10^{-5}$). A possible explanation for these findings is that the RNN does not use stochastic reconfiguration for its variational optimization (section 3.1.4), whereas the RBM does use it. Although the difference in $\epsilon$ is relatively small, this might in part explain the observations of Fig. 4.15.

The weight of excited states $\epsilon$ is close to $1\%$ in the region $J_2 < 0.5$, whereas it is higher $\epsilon \approx 3\%$ in the frustrated phase $J_2 > 0.5$. In general, the RNN has less spurious content in its wave function than the RBM, except for $J_2 = 0$ and $j_2 = 0.6$ (Fig. 4.16). Because of the RNNs better overlap at $J_2 = 1.0$, we expect the correlation function of the RNN to be more accurate than the RBM one. The correlation functions at $J_2 = 1.0$ are plotted in figure 4.17. We indeed find that the RNN performs equally well or better than the RBM, but this is only true for small distances $|i-j| < 6$. For $J_2 = 1.0$, the system is in the frustrated phase, and the antiferromagnetic (quasi)

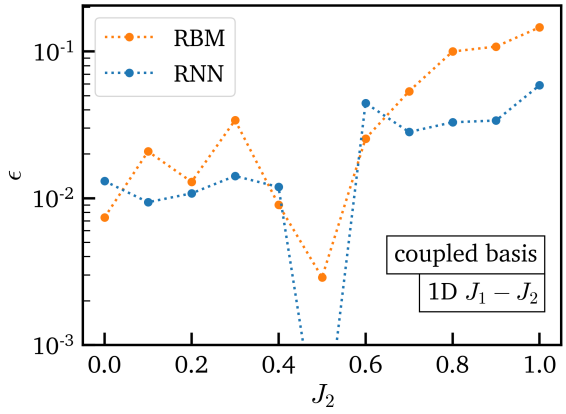



**Fig. 4.16.:** The weight of excited states $\epsilon$ in the variational ground-state wave function $|\Psi_{\mathcal{W}}\rangle$ versus $J_2$ of the $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions. The weights $\epsilon$ are shown for the optimized RBM and RNN in the coupled basis.

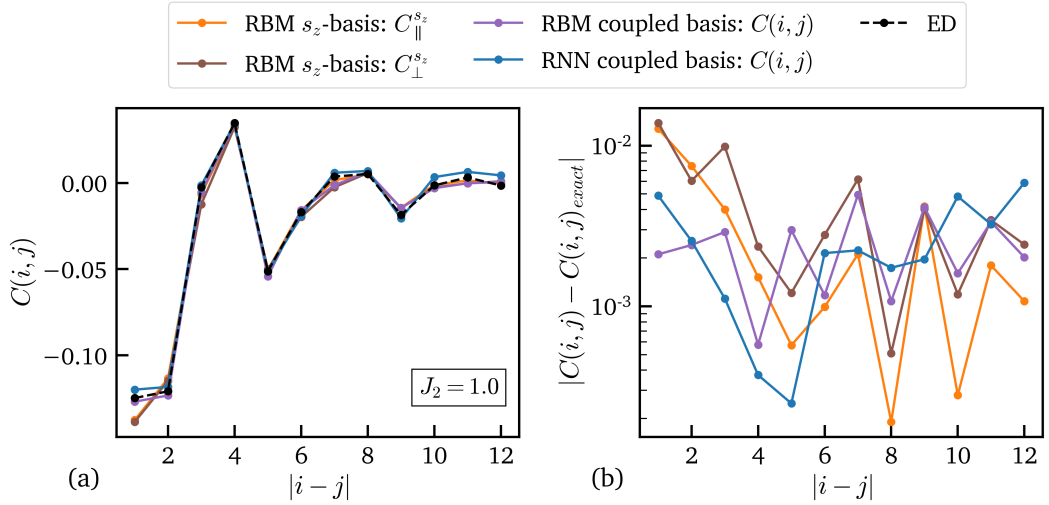



**Fig. 4.17.:** The spin-spin correlation function $C(i,j)$ for various distances $|i-j|$ for a $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions at $J_2 = 1.0$. We compare the longitudinal $C_{\parallel}^{s_z}$ and transverse $C_{\parallel}^{s_z}$ correlation functions of the RBM in the $s_z$-basis with the symmetric correlation function $C(i,j)$ obtained by the RBM and RNN in the coupled basis. The exact diagonalization (ED) result $C(i,j)_{\text{exact}}$ is taken as reference. (a) The correlation functions $C(i,j)$ versus distance between spins $|i-j|$. (b) The absolute error of the correlation functions at several distances $|i-j|$.

long-range order is no longer expected. Therefore, we did not use the prefactor $(1)^{i-j}$, since the sign of the correlation function for subsequent distances $|i-j|$ is no longer guaranteed to alternate. Indeed, the structure is different from what was obtained at the Heisenberg point $J_2 = 0$ (Fig. 4.15). Again, the RBM in the $s_z$-basis is biased towards the longitudinal component $C_{\parallel}^{s_z}$.

It is actually more common to identify the magnetic structure by measuring the spin-spin correlation structure factor defined by

$$S^2(\vec{q}) = \frac{1}{N_v(N_v+2)} \sum_{i,j} \langle \hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_j \rangle \, e^{i\vec{q}\cdot(\vec{r}_i - \vec{r}_j)}, \tag{4.12}$$

where $\vec{q}$ is the pitch vector and $\vec{r}_i$ is the direction vector of lattice site $i$. Magnetic structure is identified by a peak of $S^2(\vec{q})$ at a certain pitch vector $\vec{q}$. For example, it has been found that on a square lattice the $J_1 - J_2$ model exhibits Néel order for $J_2 \lesssim 0.5$ (section 4.2.1), which corresponds to a peak at $\vec{q} = (\pi, \pi)$. The stripe order for $J_2 \gtrsim 0.6$ corresponds to a peak at pitch vector $\vec{q} = (\pi, 0)$ or $(0, \pi)$. For one-dimensional systems, the vectors reduce to scalars. Néel order is then identified by a peak of $S^2(q)$ at pitch $q = \pi$, corresponding to the prefactor $e^{iq(i-j)} = (-1)^{i-j}$.

The implementation of SU(2) symmetry by using the coupled basis has a downside. Note the summation over all pairs of sites in Eq. (4.12), which includes large inter-site distances. Due to the way the matrix elements for an operator of type $\hat{\boldsymbol{s}}_i \cdot \hat{\boldsymbol{s}}_j$ are calculated in the coupled basis (section 3.3), the number of connected states in Eq. (3.6) increases exponentially with distance $|i-j|$. This leaves the calculation of the structure factor Eq. (4.12) computationally intractable, even for modest system sizes such as $N_v = 22$. Also, these calculations need to be repeated over a range of interaction strengths $J_2$ and for varying pitch $q$. Experiments with exact diagonalization showed that the truncation of $(i-j)$-pairs with a large distance $|i-j|$ is unjustified. Therefore, the calculation of structure factors could not be done.

**Energy gap with respect to $J_2$** In section 4.1.2, we discussed how the coupled basis allows us to determine the first excited state of the AFH model. We computed the energy gap $E_{\text{gap}} = E_1 - E_0$ between the first excited state with energy $E_1$ and the ground-state energy $E_0$ for different system sizes $N_v$. Now, we investigate the energy gap $E_{\text{gap}}(J_2)$ for different interaction strengths $J_2 \in [0, 1]$ of the $J_1 - J_2$ chain of fixed length $N_v = 22$ with open boundary conditions.

The energy gap $E_{\text{gap}}$ of an RNN $(d_h = 50, n_l = 1)$ in the coupled basis is compared to the exact diagonalization result in figure 4.18. The RNN accurately reproduces the gap in the region $J_2 < 0.5$, with relative energy errors $\Delta E_0 \sim \mathcal{O}(10^{-6}) - \mathcal{O}(10^{-5})$ and $\Delta E_1 \sim \mathcal{O}(10^{-5}) - \mathcal{O}(10^{-4})$. At the Majumdar-Ghosh point $J_2 = 0.5$, the ground-state energy is essentially exact, whereas the first excited state has an error of $\Delta E_1 \approx 10^{-3}$. The representation is less accurate in the frustrated regime $J_2 > 0.5$, where we find $\Delta E_0 \approx 10^{-4}$ and $\Delta E_1 \in [10^{-3}, 10^{-2}]$. The analysis allows us to identify $J_2$-points of interest: the energy gap $E_{\text{gap}}$ decreases up to the critical point $J_{2,c} \approx 0.241$ (see section 4.2.1) and then becomes larger for increasing interaction

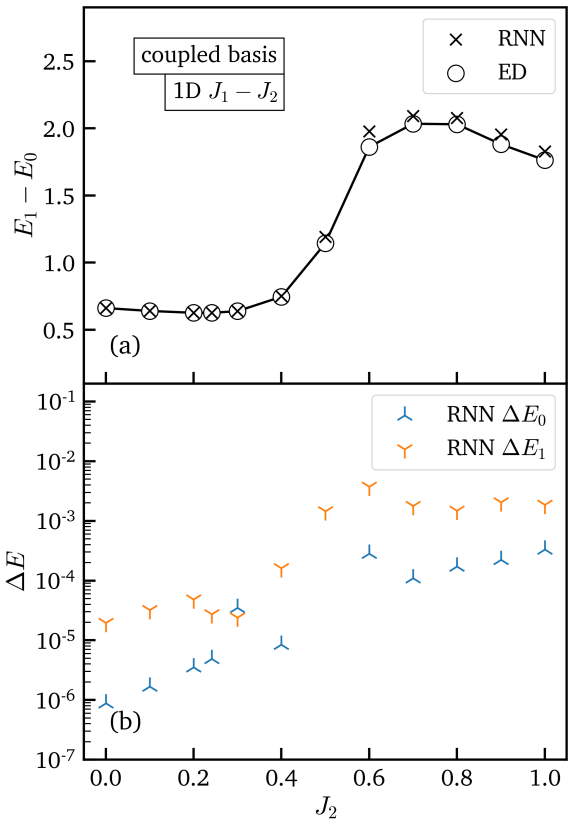



**Fig. 4.18.:** The energy gap $E_{\text{gap}} = E_1 - E_0$ between the first exited and the ground state of the $J_1 - J_2$ chain of length $N_v = 22$ with open boundary conditions. The results of an RNN $(d_h = 50, n_l = 1)$ in the coupled basis are compared to exact diagonalization (ED). (a) The energy gap $E_{\text{gap}}$ versus $J_2$. (b) The relative energy error of the ground state $\Delta E_0$ and the first excited state $\Delta E_1$ versus $J_2$.

strength $J_2$. There is a notable increase in the energy gap when going to the frustrated phase $J_2 > 0.5$. For $J_2 \gtrsim 0.7$ the energy gap shrinks.

**Cylindrical systems** Consider the two-dimensional $J_1 - J_2$ model on a rectangular lattice of size $N_{v,x} \times N_{v,y}$, where $N_{v,x(y)}$ denotes the number of sites in the $x$-direction ($y$-direction). By having periodic boundary conditions only in the $y$-direction, the geometry is that of a cylinder with length $N_{v,x}$ and circumference $N_{v,y}$. An analysis of $j_{cut}$ (similar to Fig. 4.11) for relatively small systems $N_v < 30$ shows that the truncation error $\Delta E_{cut}$ is largely independent of $N_{v,x}$, and that the error increases slowly for increasing $N_{v,y}$. We also find that using periodic boundary conditions in the $y$-direction decreases the error at given $j_{cut}$. Moreover, the truncation error $\Delta E_{cut}$ is independent of the coupling scheme (Snake or ZigZag) which dictates in what order the spins are coupled (Fig. 3.3). For all upcoming experiments, we use $j_{cut} = 4$ such that $\Delta E_{cut} < 10^{-7}$.

Since we are interested in the efficiency of the variational optimization, we investigate the average number of connected states $\langle n_{\text{CS}} \rangle$ (see section 3.3) with respect to the lattice properties. In particular, we search for the relation between the number of connected states and the lattice size. We also want to know whether the periodic boundary conditions lead to an efficiency loss. Therefore, the number of spins in the $x$-direction is fixed to $N_{v,x} = 10$ whereas we vary $N_{v,y} \in [2, 6]$. The calculation of the average number of connected states $\langle n_{\text{CS}} \rangle$ is repeated for both coupling schemes (Snake and ZigZag, see Fig. 3.3), for rectangular and cylindrical lattices, and in the standard $s_z$-basis for comparison.

A plot of the average number of connected states $\langle n_{\text{CS}} \rangle$ versus the number of lattice sites in the $y$-direction $N_{v,y}$ is shown in figure 4.19. Note that $\langle n_{\text{CS}} \rangle$ is lower in the $s_z$-basis compared to the coupled basis, and that this is the case for all considered lattice geometries and both $J_2 = 0$ and $J_2 = 1.0$. This originates from the way matrix elements are calculated in the coupled basis (section 3.3), and it proves to be the major disadvantage of using the coupled basis when dealing with a system of dimension higher than one. Two spins that are close on the physical lattice and interact with each other by a (next-to-)nearest neighbour interaction can be far away in the coupling chain (Snake or ZigZag). In the coupled basis, the number of connected states due to an interaction scales exponentially with the distance (along the chain) between the interacting spins. Therefore, the coupled basis becomes inefficient when dealing with AFH-type Hamiltonians in two or more dimensions.

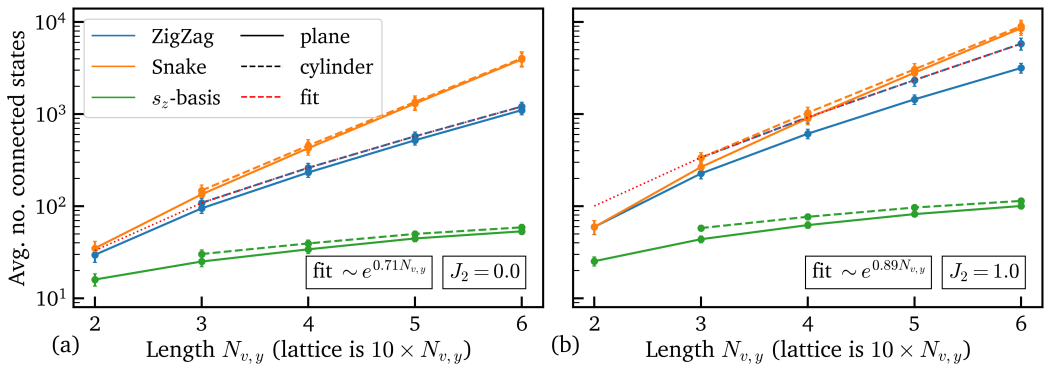



**Fig. 4.19.:** The average number of connected states versus the number of spins in the $y$-direction for the 2D $J_1 - J_2$ model with periodic boundary conditions only in the $y$-direction (cylindrical) or open boundary conditions in both dimension (plane). The Snake and ZigZag coupling schemes of the coupled basis are compared with the standard $s_z$-basis. (a) The results for the Heisenberg point $J_2 = 0$. The fit (in red) approximates the case of a cylinder with ZigZag scheme and corresponds to $y = 17.58e^{0.71 N_{v,y}} - 39.75$. (b) The results for non-zero next-to-nearest neighbour interaction strength $J_2 = 1.0$. The fit (in red, again for cylindrical geometry with ZigZag coupling scheme) is $y = 27.74e^{0.89 N_{v,y}} - 64.84$.

In the coupled basis, we try to minimize $\langle n_{\text{CS}} \rangle$ by choosing an appropriate coupling scheme. As can be seen in Fig. 4.19, the ZigZag scheme is more efficient than the Snake for all considered lattice geometries. This is explained by the fact that the ZigZag scheme has a lower maximal distance (along the chain) between interacting spins than does the ZigZag scheme. The number of connected states associated to larger distances, which can be be interpreted as the importance or weight of these distances, is exponentially higher than that of smaller distances. Thus, even though the minimal distance between interacting spins occurs more often when using the Snake, the ZigZag is more efficient because its maximal distance between interacting spins is smaller.

The average number of connected states $\langle n_{\text{CS}} \rangle$ scales exponentially with the number of spins in the $y$-direction $N_{v,y}$, as shown by the fits in Fig. 4.19. The increase of $\langle n_{\text{CS}} \rangle$ by going from open boundary conditions to a cylindrical geometry is more significant away from the Heisenberg point $J_2 = 0$.

Approximating the ground-state wave function of a two-dimensional AFH model by the variational optimization of an RNN is challenging. The generation and storage of the connected states that are used to estimate the energy gradients Eq. (3.14) at each training iteration becomes intractable for large lattices. Furthermore, we expect the optimization to be slower than for small $(N_v \approx 22)$ 1D systems, i.e. a larger number of iterations is needed to reach convergence. But even then, we find that the RNN has a tendency to get stuck in local minima when searching for the ground state of the 2D cylindrical AFH model. For example, the optimization of an RNN $(d_h = 50, n_l = 1)$ with $N_s = 500$ and $20000$ training iterations took approximately $38$ CPU-hours and led to a relative energy error $\Delta E_0 > 100\%$ for the $8 \times 4$ AFH model, which is clearly unacceptable.

We adapt the strategy proposed in Ref. [97], called iterative retraining: an RNN is optimized to represent the ground state of a small lattice system, after which the lattice size is progressively increased by "growing" new spin sites on top of the previous lattice. When a larger system is presented, the previously trained RNN is retrained until the energy is sufficiently converged. The growing and retraining is repeated until the desired system size is reached. The idea is that the RNN learns the most important features by approximating the small system, and then progressively extrapolates to systems of greater size. The RNN is retrained on the larger lattice to capture the most important new features, which are mostly related to the edge effects. In this way, the majority of the training is offloaded to small lattices. This retraining strategy is possible because the number of variational parameters of the RNN is independent of the lattice size.

We now use iterative retraining for the variational optimization of an RNN $(d_h = 50, n_l = 1)$. The goal is to represent the ground state of the 2D AFH model $(J_2 = 0)$ with a cylindrical geometry. We start with a $4 \times 4$ lattice and optimize the RNN using $500$ samples for $20000$ training iterations. The learning rate is set to the default value $\eta = 10^{-3}$. The resulting model is retrained on a lattice of size $6 \times 4$, i.e. with $N_{v,x} = 6$ and $N_{v,y} = 4$, using $200$ samples and for $10000$ iterations. For the retraining steps, we employ a smaller learning rate $\eta = 10^{-4}$. Hereafter, the model is fine-tuned on the same lattice by using $N_s = 2000$ for only $500$ training iterations and with an even smaller learning rate $\eta = 5 \times 10^{-5}$. The model is retrained for $N_{v,x} = 8$, with $N_s = 1000$ and for $2000$ training iterations. All of these runs together took a total time of approximately $18$ CPU-hours, which is half the time we needed for the (failed) direct optimization of the $8 \times 4$ lattice. Furthermore, the relative energy error is acceptable $\Delta E_0 \approx 7 \times 10^{-4}$. We continue the iterative retraining procedure until a lattice of size $20 \times 4$ is obtained. The used hyperparameters can be found in table A.3. The chosen values are anything but arbitrary — a lot of experimentation was needed to find working combinations of hyperparameters.

In figure 4.20, we show the relative energy error $\Delta E_0$ with respect to the training iteration for each of the iterative retraining steps. The converged ground-state energies, which are estimated only once using $10^6$ samples, are also displayed. As expected, the majority of the training is offloaded to the small lattices of size $4 \times 4$ and the retraining to a lattice of size $6 \times 4$ (Fig. 4.20(b)). The relative energy error $\Delta E_0$ at the beginning of a retraining iteration is generally lower for the larger lattices (later in the retraining procedure, e.g. 4.20(c)), which indicates that the RNN accurately extrapolates the learned features. The training histories show large fluctuations, especially when retraining on large lattices. This is due to the low number of samples $N_s \in [200, 500]$ used to estimate the energy in these iterative retraining steps. Possibly related, $\Delta E_0$ seems to not really decrease during the training of the latest steps (Fig. 4.20(d)). To see if there is anything new being learned during these steps, one could estimate the energy with enough samples (e.g. $10^6$) once at the beginning of the training, and once at the end of the training. The converged energies (Fig. 4.20(a)) show a trend of increasing relative energy error $\Delta E_0$ with increasing lattice size. The smallest error $\Delta E_0 \approx 3 \times 10^{-4}$ is obtained for the initial lattice of size $4 \times 4$, whereas the largest error $\Delta E_0 \approx 5 \times 10^{-3}$ occurs for the final lattice of size $20 \times 4$. In conclusion, iterative retraining allows us to approximate the ground state of the 2D AFH model with a cylindrical geometry (in the coupled basis) with $\Delta E_0$ smaller than $1\%$ for lattice sizes up to $20 \times 4$.

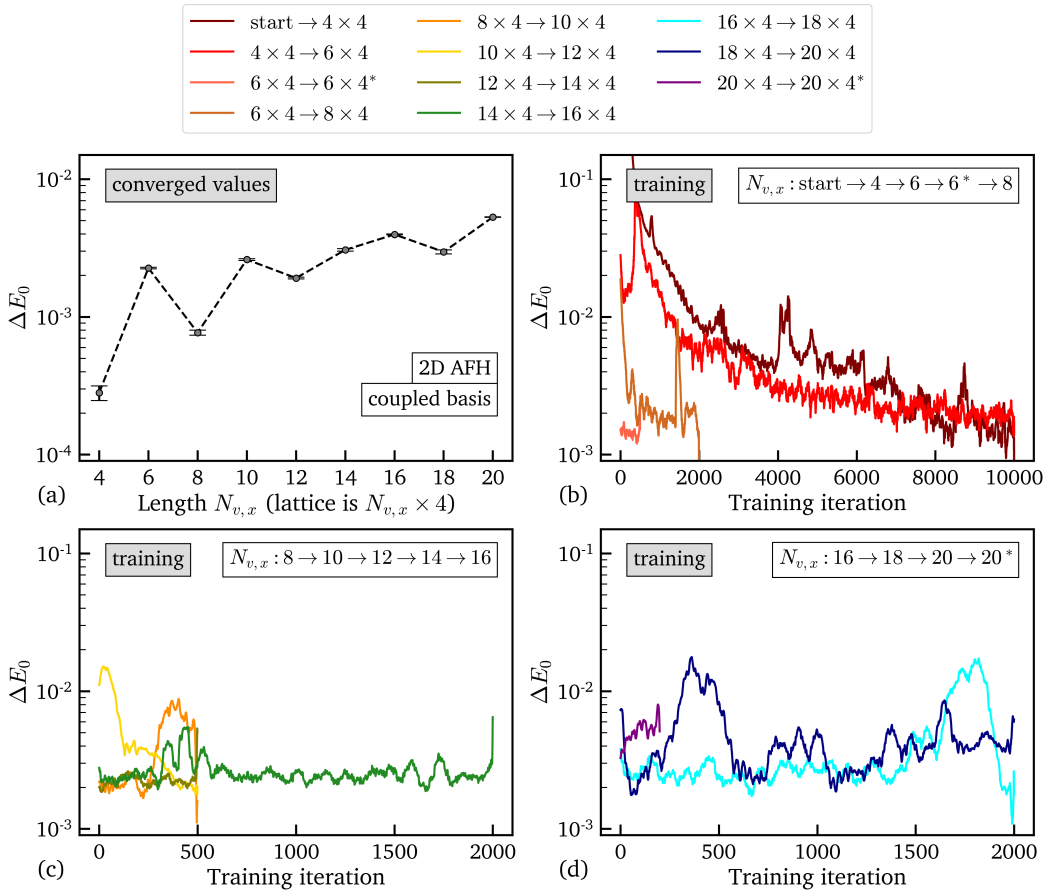



**Fig. 4.20.:** Iterative retraining of an RNN (in the coupled basis) for the 2D AFH model with a cylindrical geometry. The RNN is progressively retrained, starting with a $4 \times 4$ lattice and going up to $20 \times 4$. (b-d) The relative energy error $\Delta E_0$ with respect to the training iteration for each of iterative retraining steps. (a) The converged values obtained at the end of each step. The asterisk $^*$ indicates that the lattice size stays the same in the iterative retraining step: the step corresponds to a fine-tuning of the model. The data is smoothed for visualization purposes.

Finally, we note that a natural extension of the RNN is to make its architecture inherently two-dimensional [36]. However, we found that the relative energy errors of 2D RNNs ($\Delta E_0 \sim \mathcal{O}(10^{-2})$ for a $4 \times 4$ lattice) do not compete with those of the regular 1D architecture used in this work ($\Delta E_0 \sim \mathcal{O}(10^{-4})$). A possible explanation for these findings is that the 2D RNN is suboptimal in the coupled basis. In the $s_z$-basis, the cell obtains additional information that is approachable in the same way as the information propagated in 1D architectures (i.e. processed spin up/down information). In the coupled basis, the additional information is obtained using the intermediate angular momentum on a site that is far away along the coupling chain. This information has to be processed differently than the information of adjacent spins on the chain. The coupling chain breaks the inherent symmetry of 2D RNN architectures, which encourages the use of other coupling schemes.

# 5 Conclusion & Outlook

## 5.1 Conclusion

In this thesis, we investigated the scientific potential of recently introduced variational neural network ansätze, so-called neural network quantum states (NQS). In particular, we explored the restricted Boltzmann machine (RBM) and the recurrent neural network (RNN) as approximate wave functions of established lattice systems: the antiferromagnetic Heisenberg (AFH) model and the $J_1 - J_2$ model. The pioneering work by Carleo et al. [10] (introduction of NQS with RBMs) and Hibat-Allah et al. [36] (introduction of RNNs as NQSs) demonstrated the effectiveness of these wave functions, but also pointed towards a crucial working point. Namely, the original formulation of the NQSs does not preserve SU(2) symmetry. Therefore, total spin is not conserved. This source of error is eliminated by applying the in-house SU(2) invariant strategy of Vieijra et al. [54]. Among other things, we extended the strategy to RNNs and two-dimensional lattices. The quality of the obtained wave functions is comparable to what is found in the literature. We can substantiate that NQSs are competitive ansätze and promising candidates for future development.

In chapters 1−2, we introduced many-body physics and machine learning. This was followed by a detailed discussion of how neural networks are used to represent quantum wave functions in chapter 3. We also explained in detail how SU(2) symmetry can be implemented in NQSs — a method independent of the form of the variational state and extendable to other non-abelian symmetries. This allowed us to study the AFH and $J_1 - J_2$ models in chapter 4. The main results are as follows:

- In the standard $s_z$-basis, an appropriate sign rule initialization is needed.
- The expressivity of both the RBM and the RNN can be systematically increased, in agreement with the variational principle and the universal approximation theorem. The RNN generally outperformed the RBM, and efficiently represented ground states accurately of systems consisting of up to $100$ spins.
- In general, SU(2) invariant NQSs represent the ground state of both ordered and disordered quantum phases of matter better than their non-invariant counterparts. Translational symmetry seemed less important.

- The coupled basis allows us to accurately represent excited states and the energy gap can be computed. This allows the verification of scaling relations and the discovery of points of interest.
- The coupled basis provides profound insight into the structure of the wave function, e.g. the quantification of resonating valence bond state contributions.
- The spin-spin correlation function allows us to deduce the phase of the system; the correlations are intrinsically unbiased in the coupled basis. For some unclear reason, the RNN yielded suboptimal correlation functions. This indicates that an accurate estimate of the ground-state energy does not necessarily imply that the wave function is represented optimally.
- Our study revealed the main disadvantages of the SU(2) symmetry implementation: the efficiency of the method drops considerably when considering higher dimensional lattices. The computation of structure factors is intractable.
- Using iterative retraining [97], we studied the cylindrical AFH model with SU(2) symmetry on a lattice of size $20 \times 4$. The relative energy error remained below $1\%$. The majority of the "learning" can be offloaded to small systems.

To come to these conclusions, we compared our results to those in recent literature and made reference calculations using exact diagonalization (ED) and the density matrix renormalization group (DMRG) method. Furthermore, we discussed optimal hyperparameters, the consistency of the variational optimization (occurrence of outliers), and we proposed methods to improve the NQS approach.

## 5.2 Outlook

A logical continuation of this work substantiates our findings by: i) repeating the simulations with the RNN in the standard $s_z$-basis; ii) exploring new optimization schemes and a broader range of hyperparameters and; iii) test the limits of the coupled basis by investigating new coupling schemes, e.g. hierarchical trees or reflection symmetric cluster approaches. Furthermore, the efficiency of the RNN training can be drastically increased by taking advantage of GPUs. This may push the limits of the proposed SU(2) invariance scheme towards large scale two-dimensional lattice problems. The techniques illustrated in this work, for example the investigation of the wave function structure, can be repeated for frustrated phases and two-dimensional models to shed light on the underlying mechanisms. The implementation of SU(2) symmetry can be readily reconciled with lattice symmetries of the system.

Although preliminary tests with 2D RNNs indicated that the 2D architecture is suboptimal with the current implemented scheme of SU(2) symmetry (a linear chain of coupled spins), an in depth analysis with other coupling schemes may prove worthwhile. In this regard, an interesting idea is to traverse the lattice with more than one coupling chain and then construct the symmetric wave function similar to how it is usually done for the lattice symmetries (see for example Ref. [91]). Investigating the internal state of the NQSs by examining the variational parameters is deemed a fruitful direction for future consideration. This may hint towards appropriate modifications of the RBM and RNN architectures and their internal workings to be more suitable when working in the coupled basis. Of course, entirely different models can also be examined. With a focus on a fully symmetric state (both internal and lattice symmetries), we refer to the recently developed construction in Ref. [53]. Given the rapidly evolving nature of machine learning, we expect that the class of neural network quantum states will provide increasingly relevant tools for studying the quantum many-body problem.

# Bibliography


[1]P. A. M. Dirac and R. H. Fowler, "Quantum mechanics of many-electron systems", Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character **123**, 714–733 (1929) (cit. on p. v).

[2]J. Sethna, *Statistical mechanics: entropy, order parameters and complexity*, Oxford Master Series in Physics (OUP Oxford, 2006) (cit. on p. 4).

[3]E. Ising, "Beitrag zur theorie des ferromagnetismus", Zeitschrift für Physik **31**, 253–258 (1925) (cit. on p. 4).

[4]H. Gould and J. Tobochnik, *Statistical and thermal physics: with computer applications* (Princeton University Press, 2010) (cit. on pp. 6, 7).

[5]K. Christensen and N. R. Moloney, *Complexity and criticality* (Imperial College Press, 2005) (cit. on p. 6).

[6]L. Onsager, "Crystal statistics. I. A two-dimensional model with an order-disorder transition", Phys. Rev. **65**, 117–149 (1944) (cit. on p. 7).

[7]M. Newman and G. Barkema, *Monte Carlo methods in statistical physics* (Clarendon Press, 1999) (cit. on pp. 7, 14, 40).

[8]A. Morningstar and R. G. Melko, "Deep learning the ising model near criticality", Journal of Machine Learning Research **18**, 1–17 (2018) (cit. on p. 7).

[9]N. Hugenholtz, "Perturbation theory of large quantum systems", Physica **23**, 481–532 (1957) (cit. on p. 9).

[10]G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks", Science **355**, 602–606 (2017) (cit. on pp. 9, 38, 40, 99).

[11]J. Haegeman, *Lecture notes: Strongly Correlated Quantum Systems*, Physics course at Ghent University, 2020 (cit. on pp. 10, 12, 17, 67–69).

[12]M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2000) (cit. on pp. 11, 12).

[13]M. Suzuki, "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems", Communications in Mathematical Physics **51**, 183–190 (1976) (cit. on p. 13).

[14]M. Suzuki, "Relationship between d-Dimensional Quantal Spin Systems and (d+1)-Dimensional Ising Systems: Equivalence, Critical Exponents and Systematic Approximants of the Partition Function and Spin Correlations", Progress of Theoretical Physics **56**, 1454–1469 (1976) (cit. on p. 14).

[15]M. Troyer and U.-J. Wiese, "Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations", Physical Review Letters **94** (2005) (cit. on p. 14).

[16]J. Schwichtenberg, *Physics from symmetry*, Undergraduate Lecture Notes in Physics (Springer International Publishing, 2015) (cit. on pp. 15, 17).

[17]K. Brading and E. Castellani, *Symmetries and invariances in classical physics*, Forthcoming in J. Butterfield and J. Earman (eds.), Handbook of the Philosophy of Physics, North-Holland. (2005) (cit. on p. 15).

[18]E. Noether, "Invariante variationsprobleme", Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse **1918**, 235–257 (1918) (cit. on p. 16).

[19]D. J. Gross, "The role of symmetry in fundamental physics", Proceedings of the National Academy of Sciences **93**, 14256–14259 (1996) (cit. on p. 16).

[20]A. Beekman, L. Rademaker, and J. van Wezel, "An introduction to spontaneous symmetry breaking", SciPost Physics Lecture Notes (2019) (cit. on p. 16).

[21]K. Brading, E. Castellani, and N. Teh, "Symmetry and Symmetry Breaking", in *The Stanford encyclopedia of philosophy*, edited by E. N. Zalta, Winter 2017 (Metaphysics Research Lab, Stanford University, 2017) (cit. on p. 16).

[22]T. Mitchell, *Machine learning* (McCraw Hill, 1997) (cit. on p. 19).

[23]G. Carleo, I. Cirac, K. Cranmer, et al., "Machine learning and the physical sciences", Reviews of Modern Physics **91** (2019) (cit. on p. 19).

[24]Y. Bahri, J. Kadmon, J. Pennington, et al., "Statistical mechanics of deep learning", Annual Review of Condensed Matter Physics **11**, 501–528 (2020) (cit. on p. 19).

[25]Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database", ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010) (cit. on p. 24).

[26]D.-y. Ge, X.-f. Yao, W.-j. Xiang, X.-j. Wen, and E.-c. Liu, "Design of high accuracy detector for MNIST handwritten digit recognition based on convolutional neural network", in 12th international conference on intelligent computation technology and automation (icicta) (2019), pp. 658–662 (cit. on p. 24).

[27]Q. Song, W. Hu, and W. Xie, "Robust support vector machine with bullet hole image classification", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **32**, 440–448 (2002) (cit. on p. 24).

[28]I. T. Jollife and J. Cadima, "Principle component analysis: a review and recent developments", Phil. Trans. R. Soc. A. **374** (2016) (cit. on pp. 24, 25).

[29]S. J. Pan and Q. Yang, "A survey on transfer learning", IEEE Transactions on Knowledge and Data Engineering **22**, 1345–1359 (2010) (cit. on p. 25).

[30]C. Cortes and V. Vapnik, "Support-vector networks", Machine Learning **20**, 273–297 (1995) (cit. on p. 26).

[31]A. Ben-Hur and J. Weston, "A user's guide to support vector machines", Methods in molecular biology (Clifton, N.J.) **609**, 223–39 (2010) (cit. on p. 27).

[32]M. Collins, *Convergence proof for the perceptron algorithm*, Course notes, Columbia University (2012) (cit. on p. 29).

[33]K. Hornik, "Approximation capabilities of multilayer feedforward networks", Neural Networks **4**, 251–257 (1991) (cit. on p. 30).

[34]R. Hecht-Nielsen, "III.3 - Theory of the Backpropagation Neural Network Based on "nonindent" by Robert Hecht-Nielsen ... .", in *Neural networks for perception*, edited by H. Wechsler (Academic Press, 1992), pp. 65–93 (cit. on p. 31).

[35]A. Fischer and C. Igel, "An introduction to restricted boltzmann machines", in Progress in pattern recognition, image analysis, computer vision, and applications, edited by L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo (2012), pp. 14–36 (cit. on p. 31).

[36]M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, "Recurrent neural network wave functions", Physical Review Research **2** (2020) (cit. on pp. 33, 46, 47, 73, 74, 97, 99).

[37]Z. C. Lipton, "A critical review of recurrent neural networks for sequence learning", CoRR **abs/1506.00019** (2015) (cit. on p. 33).

[38]K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder–decoder approaches", in Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation (Oct. 2014), pp. 103–111 (cit. on p. 33).

[39]G. P. Styan, "Hadamard products and multivariate statistical analysis", Linear Algebra and its Applications **6**, 217–240 (1973) (cit. on p. 33).

[40]X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in Proceedings of the thirteenth international conference on artificial intelligence and statistics, Vol. 9, edited by Y. W. Teh and M. Titterington, Proceedings of Machine Learning Research (2010), pp. 249–256 (cit. on p. 35).

[41]K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification", CoRR **abs/1502.01852** (2015) (cit. on p. 35).

[42]W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications", Biometrika **57**, 97–109 (1970) (cit. on p. 40).

[43]H. Katzgraber, "Introduction to Monte Carlo methods", arXiv: Statistical Mechanics (2009) (cit. on p. 40).

[44]J. C. Slater, "A simplification of the Hartree-Fock method", Phys. Rev. **81**, 385–390 (1951) (cit. on p. 41).

[45]S. R. White, "Density matrix formulation for quantum renormalization groups", Phys. Rev. Lett. **69**, 2863–2866 (1992) (cit. on pp. 41, 65).

[46]S.-i. Amari, "Backpropagation and stochastic gradient descent method", Neurocomputing **5**, 185–196 (1993) (cit. on p. 42).

[47]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, Conference Track Proceedings, edited by Y. Bengio and Y. LeCun (2015) (cit. on pp. 42, 43).

[48]S. Sorella, "Green function Monte Carlo with stochastic reconfiguration", Phys. Rev. Lett. **80**, 4558–4561 (1998) (cit. on p. 43).

[49]S. Sorella, M. Casula, and D. Rocca, "Weak binding between two aromatic rings: feeling the van der Waals attraction by quantum Monte Carlo methods", The Journal of Chemical Physics **127**, 014105 (2007) (cit. on p. 43).

[50]E. Neuscamman, C. J. Umrigar, and G. K.-L. Chan, "Optimizing large parameter sets in variational quantum Monte Carlo", Phys. Rev. B **85**, 045103 (2012) (cit. on p. 44).

[51]S. Sorella and F. Becca, *SISSA Lecture notes on numerical methods for strongly correlated electrons*, pp. 61-88, SISSA/ISAS - International School for Advanced Studies, June 2016 (cit. on pp. 44, 45).

[52]S.-X. Zhang, Z.-Q. Wan, and H. Yao, *Automatic differentiable Monte Carlo: theory and application*, 2019 (cit. on p. 48).

[53]T. Vieijra and J. Nys, "Many-body quantum states with exact conservation of non-abelian and lattice symmetries through variational Monte Carlo", Physical Review B **104** (2021) (cit. on pp. 50, 101).

[54]T. Vieijra, C. Casert, J. Nys, et al., "Restricted Boltzmann machines for quantum states with non-abelian or anyonic symmetries", Physical Review Letters **124** (2020) (cit. on pp. 50, 72–74, 76, 77, 79, 84, 89, 99).

[55]J. J. Sakurai and J. Napolitano, *Modern quantum mechanics*, 2nd ed. (Cambridge University Press, 2017), pp. 252–255 (cit. on pp. 55, 89).

[56]S. Singh and G. Vidal, "Tensor network states and algorithms in the presence of a global SU(2) symmetry", Phys. Rev. B **86**, 195114 (2012) (cit. on p. 55).

[57]J. Rasch and A. C. H. Yu, "Efficient storage scheme for precalculated Wigner 3j, 6j and Gaunt coefficients", SIAM J. Sci. Comput. **25**, 1416–1428 (2003) (cit. on p. 58).

[58]E. Anderson, Z. Bai, C. Bischof, et al., *LAPACK users' guide*, 3rd ed. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999) (cit. on p. 59).

[59]C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", Journal of the National Bureau of Standards **45**, 255–282 (1950) (cit. on p. 59).

[60]I. S. Dhillon, "A new $n^2$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem", PhD thesis (EECS Department, University of California, Berkeley, Oct. 1997) (cit. on p. 60).

[61]M. Gu and S. C. Eisenstat, "A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem", SIAM Journal on Matrix Analysis and Applications **16**, 172–191 (1995) (cit. on p. 60).

[62]E. S. Coakley and V. Rokhlin, "A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices", Applied and Computational Harmonic Analysis **34**, 379–414 (2013) (cit. on p. 61).

[63]P. Prelovsek and J. Bonca, "Strongly correlated systems", Springer Series in Solid-State Sciences, 1–30 (2013) (cit. on p. 61).

[64]R. Orús, "A practical introduction to tensor networks: matrix product states and projected entangled pair states", Annals of Physics **349**, 117–158 (2014) (cit. on pp. 61, 63, 65, 66).

[65]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) (cit. on pp. 62, 66).

[66]P. Silvi, F. Tschirsich, M. Gerster, et al., "The tensor networks anthology: simulation techniques for many-body quantum lattice systems", SciPost Physics Lecture Notes (2019) (cit. on p. 62).

[67]U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states", Annals of Physics **326**, 96–192 (2011) (cit. on pp. 64–66).

[68]G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, "Entanglement in quantum critical phenomena", Phys. Rev. Lett. **90**, 227902 (2003) (cit. on p. 64).

[69]D.-L. Deng, X. Li, and S. Das Sarma, "Quantum entanglement in neural network states", Phys. Rev. X **7**, 021021 (2017) (cit. on p. 64).

[70]I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, "Neural-network quantum states, string-bond states, and chiral topological states", Physical Review X **8** (2018) (cit. on p. 64).

[71]F. Mezzacapo, N. Schuch, M. Boninsegni, and J. I. Cirac, "Ground-state properties of quantum many-body systems: entangled-plaquette states and variational Monte Carlo", New Journal of Physics **11**, 083026 (2009) (cit. on p. 64).

[72]N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, "Simulation of quantum many-body systems with strings of operators and Monte Carlo tensor contractions", Phys. Rev. Lett. **100**, 040501 (2008) (cit. on p. 64).

[73]M. Fannes, B. Nachtergaele, and R. F. Werner, "Finitely correlated states on quantum spin chains", Commun. Math. Phys. **144**, 443–490 (1992) (cit. on p. 65).

[74]S. Rommer and S. Östlund, "Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group", Physical Review B **55**, 2164–2181 (1997) (cit. on pp. 65, 66).

[75]A. E. Feiguin, "The density matrix renormalization group", in *Strongly correlated systems: numerical methods*, edited by A. Avella and F. Mancini (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 31–65 (cit. on p. 66).

[76]N. F. Mott, "The basis of the electron theory of metals, with special reference to the transition metals", Proceedings of the Physical Society. Section A **62**, 416–422 (1949) (cit. on p. 67).

[77]W. Marshall, "Antiferromagnetism", Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **232**, 48–68 (1955) (cit. on p. 69).

[78]E. Lieb and D. Mattis, "Ordering energy levels of interacting spin systems", Journal of Mathematical Physics **3**, 749–751 (1962) (cit. on p. 69).

[79]M. B. Hastings, "Lieb-Schultz-Mattis in higher dimensions", Phys. Rev. B **69**, 104431 (2004) (cit. on p. 70).

[80]B. Nachtergaele and R. Sims, "A multi-dimensional Lieb-Schultz-Mattis theorem", Communications in Mathematical Physics **276**, 437–472 (2007) (cit. on p. 70).

[81]N. D. Mermin and H. Wagner, "Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models", Phys. Rev. Lett. **17**, 1133–1136 (1966) (cit. on p. 70).

[82]P. Hohenberg, "Existence of long-range order in one and two dimensions", Physical Review **158**, 383–386 (1967) (cit. on p. 70).

[83]S. Coleman, "There are no Goldstone bosons in two dimensions", Communications in Mathematical Physics **31**, 259–264 (1973) (cit. on p. 70).

[84]R. F. Bishop, D. J. J. Farnell, and J. B. Parkinson, "Phase transitions in the spin-half J1-J2 model", Phys. Rev. B **58**, 6394–6402 (1998) (cit. on pp. 70, 83).

[85]H. Bethe, "Zur theorie der metalle", Zeitschrift für Physik **71**, 205–226 (1931) (cit. on p. 70).

[86]K. Choo, T. Neupert, and G. Carleo, "Two-dimensional frustrated J1-J2 model studied with neural network quantum states", Physical Review B **100** (2019) (cit. on pp. 70, 74, 79, 83).

[87]K. Choo, G. Carleo, N. Regnault, and T. Neupert, "Symmetries and many-body excitations with neural-network quantum states", Physical Review Letters **121** (2018) (cit. on pp. 77, 79).

[88]P. W. Anderson, "The resonating valence bond state in La2CuO4 and superconductivity", Science **235**, 1196–1198 (1987) (cit. on p. 80).

[89]L. Wang, D. Poilblanc, Z.-C. Gu, X.-G. Wen, and F. Verstraete, "Constructing a gapless spin-liquid state for the spin-1/2 J1-J2 Heisenberg model on a square lattice", Physical Review Letters **111** (2013) (cit. on p. 83).

[90]F. Mezzacapo, "Ground-state phase diagram of the quantum J1-J2 model on the square lattice", Phys. Rev. B **86**, 045115 (2012) (cit. on p. 83).

[91]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*, (2021) (cit. on pp. 83, 101).

[92]J. Sirker, V. Y. Krivnov, D. V. Dmitriev, et al., "The J1-J2 Heisenberg model at and close to its z=4 quantum critical point", Physical Review B **84** (2011) (cit. on p. 83).

[93]S. N. Saadatmand, B. J. Powell, and I. P. McCulloch, "Phase diagram of the spin-half triangular J1-J2 Heisenberg model on a three-leg cylinder", Physical Review B **91** (2015) (cit. on p. 83).

[94]R. Chitra, S. Pati, H. R. Krishnamurthy, D. Sen, and S. Ramasesha, "Density-matrix renormalization-group studies of the spin-1/2 Heisenberg system with dimerization and frustration", Phys. Rev. B **52**, 6581–6587 (1995) (cit. on p. 83).

[95]S. R. White and I. Affleck, "Dimerization and incommensurate spiral spin correlations in the zigzag spin chain: analogies to the Kondo lattice", Physical Review B **54**, 9862–9869 (1996) (cit. on p. 83).

[96]C. K. Majumdar, "Antiferromagnetic model with known ground state", Journal of Physics C: Solid State Physics **3**, 911–915 (1970) (cit. on p. 83).

[97]C. Roth, *Iterative retraining of quantum spin models using recurrent neural networks*, arXiv preprint: arXiv:2003.06228 (2020) (cit. on pp. 95, 100).

# List of Figures

# List of Tables

# List of Symbols

| | |
|---|---|
| $\alpha$ | Hidden unit density (RBM) |
| $\beta$ | Inverse temperature $(k_B T)^{-1}$ |
| $\gamma$ | Quasi long-range order power law exponent |
| $\delta_{a,b}$ | Kronecker delta function with inputs $a$ and $b$ |
| $\epsilon$ | Weight of excited states in ground state wave function |
| $\varepsilon^{abc}$ | Levi-Civita symbol |
| $\eta$ | Learning rate |
| $\lambda$ | Eigenvalue of matrix |
| $\hat{\rho}$ | Density operator |
| $\hat{\sigma}^a$ | Pauli operator ($a = x, y$, or $z$) |
| $\hat{\boldsymbol{\sigma}}$ | Pauli vector |
| $\boldsymbol{\sigma}$ | Many-body spin configuration |
| $\|\boldsymbol{\sigma}\rangle$ | Many-body spin configuration as basis state |
| $\sigma^a$ | Eigenvalue of ($a = x, y$ or $z$) Pauli operator |
| $\phi(\boldsymbol{\sigma})$ | Phase associated to state $\|\boldsymbol{\sigma}\rangle$ |
| $\psi(\boldsymbol{\sigma})$ | Complex amplitude associated to state $\|\boldsymbol{\sigma}\rangle$ |
| $\|\Psi\rangle$ | Quantum wave function (or state vector) |
| $\Omega(E)$ | Number of microstates with energy E |
| $a_i$ | Bias of visible node $i$ (also called visible unit) |
| $b_j^q$ | Bias of hidden node $j$ in layer $q$ |
| $\mathcal{C}$ | Cost function |
| $\mathbb{C}$ | Complex field |
| $C(i,j)$ | Spin-spin correlation function between spins on sites $i$ and $j$ |
| $C_{\xi,\zeta}(i,j)$ | Components of the correlation function ($\xi, \zeta = x, y$ or $z$) |
| $C_{\parallel}^{s_z}(i,j)$ | Longitudinal correlation function |
| $C_{\perp}^{s_z}(i,j)$ | Transverse correlation function |
| $d$ | Lattice dimension (spatial) |
| $d_h$ | Hidden vector state length a.k.a. number of memory units (RNN) |
| $D$ | Local Hilbert space dimension |
| $\mathcal{D}$ | Many-body Hilbert space dimension |
| $e_L$ | Local energy |
| $E$ | Energy |

| | |
|---|---|
| $E_0$ | Ground state energy |
| $E_1$ | Energy of the first excited state |
| $\Delta E_0$ | Relative energy error of the ground state |
| $\Delta E_1$ | Relative energy error of the first excited state |
| $E_{\text{gap}}$ | Energy gap $E_1 - E_0$ |
| $\Delta E_{cut}$ | Relative truncation error due to $j_{cut}$ |
| $F$ | Free energy / non-linear activation function |
| $g_w$ | Gradient of energy w.r.t. parameter $w$ |
| $g$ | Group element |
| $G$ | Group |
| $\hbar$ | Planck constant (reduced) |
| $h_{x(z)}$ | External magnetic field strength in $x$-direction ($z$-direction) |
| $\hat{h}$ | Term of Hamiltonian |
| $\mathbf{h}$ | Vector of hidden units (RBM) / hidden vector state (RNN) |
| $h_j^q$ | Output of hidden node $j$ in layer $q$ |
| $\hat{H}$ | Hamiltonian |
| $\mathcal{H}$ | Hilbert space |
| $\|i\rangle$ | Arbitrary single particle basis state |
| $\hat{I}$ | Identity operator |
| $j_i$ | The $i$-th intermediate angular momentum |
| $j_{cut}$ | Cut-off angular momentum |
| $j_{max}$ | Maximal angular momentum |
| $J$ | Total angular momentum / interaction strength (Ising model) |
| $J_1$ | Nearest neighbour interaction strength |
| $J_2$ | Next-to-nearest neighbour interaction strength |
| $\hat{\boldsymbol{J}}$ | Total angular momentum operator |
| $\hat{J}_z$ | Angular momentum projection operator on the $z$-axis |
| $k_B$ | Boltzmann constant |
| $\mathcal{L}$ | Loss function |
| $m$ | Magnetization |
| $M_J$ | Projection of total angular momentum J |
| $\langle n_{CS} \rangle$ | Average number of connected states |
| $n_l$ | Number of layers of stacked recurrent cells (RNN) |
| $n_{\text{par}}$ | Total number of variational parameters in a model |
| $N$ | Number of degrees of freedom |
| $N_h$ | Number of hidden units (RBM) |
| $N_v$ | Number of visible units (usually equal to the number of spins) |
| $N_{v,x(y)}$ | Number of visible units in $x$-direction ($y$-direction) |
| $N_s$ | Number of samples |

| | |
|---|---|
| $O_L(\boldsymbol{\sigma})$ | Local estimator of operator $\hat{O}$ for state $\|\boldsymbol{\sigma}\rangle$ |
| $\hat{O}$ | Arbitrary Hermitian observable |
| $\mathcal{O}_w(\boldsymbol{\sigma})$ | Logarithmic derivative of complex amplitude $\psi(\boldsymbol{\sigma})$ w.r.t. $w$ |
| $\mathcal{O}$ | Big O notation |
| $p$ | Probability distribution |
| $\hat{P}$ | Spin-flip operator |
| $\vec{q}$ | Pitch vector $(q_1, q_2)$ |
| $q$ | Pitch angle |
| $\hat{R}(\theta)$ | Spin-rotation operator by angle $\theta$ |
| $\mathbb{R}$ | Field of real numbers |
| $s$ | Classical "spin" (discrete two-level variable) |
| $\hat{s}_a$ | Spin-projection operator on $(a = x, y,$ or $z)$-axis |
| $s_a$ | Spin projection on $(a = x, y,$ or $z)$-axis |
| $\hat{\boldsymbol{s}}$ | Spin vector |
| $S$ | Entropy |
| $S_e$ | Entanglement entropy |
| $S^2(\vec{q})$ | Spin-spin correlation structure factor |
| $\boldsymbol{S}$ | Covariance matrix |
| $\mathcal{S}$ | Softmax layer (RNN) |
| $T$ | Temperature |
| $T_c$ | Critical temperature |
| $\boldsymbol{T}$ | Transfer matrix |
| $\hat{T}$ | Translational symmetry operator |
| $\hat{U}$ | Arbitrary unitary operator |
| $V$ | Volume |
| $\mathcal{V}$ | Arbitrary vector space |
| $\mathcal{W}$ | Set of all variational parameters |
| $w_{ij}^l$ | Weight from node $i$ to node $j$ with $j$ in layer $l$ |
| $\mathbf{w}$ | Weight matrix |
| $Z$ | Partition function |
| $Z_q$ | Quantum mechanical partition function |

# Appendix

# A

## A.1 Hyperparameter sweeps

Both the RBM (section 3.1) and the RNN (section 3.2) possess hyperparameters that impact their performance. For the RBM, the most prominent hyperparameters are: the learning rate $\eta$, the optimizer (e.g. Adamax or SGD, with or without stochastic reconfiguration (section 3.1.4)), the number of samples used to estimate gradients at each step $N_s$, the standard deviation $\sigma$ for initializing the variational parameters, and the number of hidden units $N_h$. It is in practice impossible to run simulations for all possible combinations of hyperparameters. Fortunately, many of the hyperparameters are almost independent. We start with a set of well-chosen simulations to find suitable values for the hyperparameters of the models. We illustrate this for the RBM, and afterwards summarize the results of the RNN. For these simulations, we keep the system size relatively small $N_v \in \{10, 22\}$ and use open boundary conditions. The RBM's hidden unit density is fixed $\alpha = N_h/N_v = 1$. Sets of hyperparameters are selected and the ground state is approximated. Subsequently, dependencies between the parameters are investigated.

The most notable results of the hyperparameter sweep can be found in Fig. A.1. We show either SGD or Adamax as optimizer, a learning rate $\eta \in [0.1, 0.001]$ and a standard deviation of the initial parameters $\sigma \in [0.1, 0.01]$. This was done both in the $s_z$-basis and the coupled basis. All runs made use of stochastic reconfiguration (section 3.1.4), ran for a total of $5000$ training iterations, and used $N_s = 1000$ samples per iteration to estimate expectation values and gradients (Eqs. (3.8) and (3.25)). The results show the importance of choosing appropriate hyperparameters, since the obtained accuracy varies greatly throughout the explored regions. In the $s_z$-basis, the SGD optimizer performs best for small learning rates $\eta \approx 0.1$. The opposite is true for the coupled basis, where we find that $\eta \approx 0.01$ generally yields the lowest energies. The Adamax optimizer obtains the lowest energies for the smallest learning rate $\eta = 0.001$, but the results depend on the system size $N_v$, especially in the coupled basis. In the rest of this work, we only use the Adamax optimizer for the RBM. The standard deviation $\sigma$ shows no clear trend regarding the ground state accuracy, and is fixed to $\sigma = 0.1$ in all experiments.

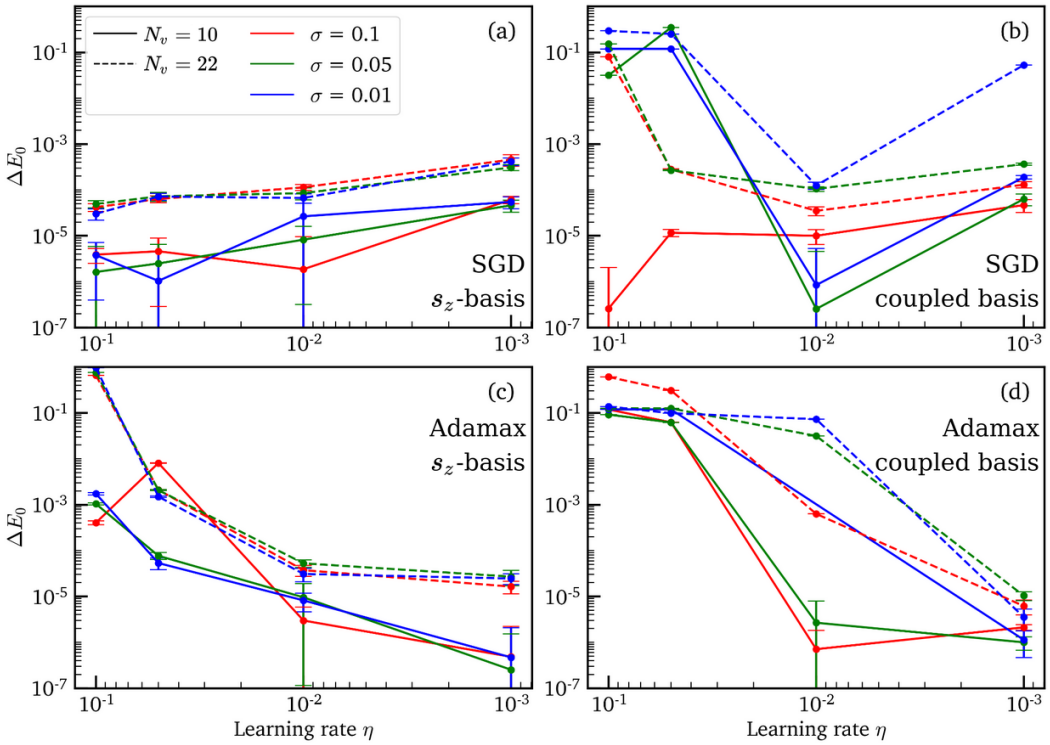



**Fig. A.1.:** The relative energy errors $\Delta E_0$ of a hyperparameter sweep for the RBM with $\alpha = 1$ and using stochastic reconfiguration. The results are for the AFH chain, and two system sizes with open boundary conditions are investigated: (a) and (b) SGD optimizer; (c) and (d) Adamax optimizer; (a) and (c) $s_z$-basis; (b) and (d) coupled basis.

For the RNN, the lowest relative energy errors $\Delta E_0$ are obtained using the Adam optimizer (section 3.1.4) with a learning rate decaying according to Eq. (3.34) and initialized to $\eta \in [10^{-4}, 5 \times 10^{-3}]$, depending on the system size and the network complexity. The variational parameters are Xavier initialized (section 2.3.3).

## A.2 Tables of hyperparameters

**Tab. A.1.:** A summary of the hyperparameters that are used in section 4.1.2. For the RBM architecture, we specify the hidden unit density $\alpha$. For the RNN architecture, the number of memory units $d_h$ and the number of layers $n_l$ are given. Additional information can be found in Appendix A.1.

| Figure | Hyperparameter | Value | |
|---|---|---|---|
| | | RBM | RNN |
| Fig. 4.3 | Architecture | $\alpha = 1, 10, 20$ | |
| | Number of samples | $N_s = 2000$ | |
| | Learning rate | $\eta = 0.001$ | N/A |
| | Training iterations | 20000 | |
| | Basis | $s_z$-basis | |
| Fig. 4.4 | Architecture | $\alpha = 0.5, 1, 2, 4$ | |
| | Number of samples | $N_s = 1000$ | |
| | Learning rate | $\eta = 0.001$ | N/A |
| | Training iterations | 20000 | |
| | Basis | $s_z$- and coupled basis | |
| Fig. 4.5 | Architecture | | $d_h = 16, 32, 64, 128$<br>$n_l = 1, 2, 3, 4$ |
| | Number of samples | | $N_s = 1000$ |
| | Learning rate | N/A | $\eta = 0.001$ |
| | Training iterations | | 20000 |
| | Basis | | coupled basis |
| Fig. 4.6 | Architecture | $\alpha = 1$ | $d_h = 32$, $n_l = 1$ |
| | Number of samples | $N_s = 1000$ | $N_s = 1000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | coupled basis | coupled basis |
| Fig. 4.7 | Architecture | | $d_h = 32$, $n_l = 1$ |
| | Number of samples | | $N_s = 1000$ |
| | Learning rate | N/A | $\eta = 0.001$ |
| | Training iterations | | 10000 |
| | Basis | | coupled basis |
| Fig. 4.8 | Architecture | $\alpha = 1$ | $d_h = 32$, $n_l = 1$ |
| | Number of samples | $N_s = 1000$ | $N_s = 1000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | coupled basis | coupled basis |
| Fig. 4.9 & Fig. 4.10 | Architecture | $\alpha = 1$ | $d_h = 32$, $n_l = 1$ |
| | Number of samples | $N_s = 1000$ | $N_s = 1000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | coupled basis | coupled basis |

**Tab. A.2.:** A summary of the hyperparameters that are used in section 4.2.2. For the RBM architecture, we specify the hidden unit density $\alpha$. For the RNN architecture, the number of memory units $d_h$ and the number of layers $n_l$ are given. Additional information can be found in Appendix A.1.

| Figure | Hyperparameter | Value | |
|---|---|---|---|
| | | RBM | RNN |
| Fig. 4.12 | Architecture | $\alpha = 1$ | $d_h = 50$, $n_l = 1$ |
| | Number of samples | $N_s = 2000$ | $N_s = 2000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | $s_z$- and coupled basis | coupled basis |
| Fig. 4.13 | Architecture | $\alpha = 1$ | N/A |
| | Number of samples | $N_s = 2000$ | |
| | Learning rate | $\eta = 0.001$ | |
| | Training iterations | 20000 | |
| | Basis | $s_z$-basis | |
| Fig. 4.14 | Architecture | $\alpha = 1$ (RBM), $\alpha = 20$ (TRBM) | N/A |
| | Number of samples | $N_s = 2000$ | |
| | Learning rate | $\eta = 0.001$ | |
| | Training iterations | 20000 | |
| | Basis | $s_z$-basis | |
| Fig. 4.15 & Fig. 4.17 | Architecture | $\alpha = 1$ | $d_h = 50$, $n_l = 1$ |
| | Number of samples | $N_s = 2000$ | $N_s = 2000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | $s_z$- and coupled basis | coupled basis |
| Figs. 4.16 | Architecture | $\alpha = 1$ | $d_h = 50$, $n_l = 1$ |
| | Number of samples | $N_s = 2000$ | $N_s = 2000$ |
| | Learning rate | $\eta = 0.001$ | $\eta = 0.001$ |
| | Training iterations | 20000 | 20000 |
| | Basis | coupled basis | coupled basis |
| Figs. 4.18 | Architecture | N/A | $d_h = 50$, $n_l = 1$ |
| | Number of samples | | $N_s = 2000$ |
| | Learning rate | | $\eta = 0.001$ |
| | Training iterations | | 20000 |
| | Basis | | coupled basis |

**Tab. A.3.:** A summary of the hyperparameters that are used during the iterative retraining of an RNN in section 4.2.2, Fig. 4.20. The RNN expresses the wave function in the coupled basis, and it has a number of memory units $d_h = 50$ and a single layer $n_l = 1$. Additional information can be found in Appendix A.1. The asterisk $^*$ indicates a fine-tuning step.

| Iterative retraining step | Hyperparameter | Value |
|---|---|---|
| start $\rightarrow 4 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 500$<br>$\eta = 10^{-3}$<br>10000 |
| $4 \times 4 \rightarrow 6 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 200$<br>$\eta = 10^{-4}$<br>10000 |
| $6 \times 4 \rightarrow 6 \times 4^*$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 2000$<br>$\eta = 5 \times 10^{-5}$<br>500 |
| $6 \times 4 \rightarrow 8 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 1000$<br>$\eta = 10^{-4}$<br>2000 |
| $8 \times 4 \rightarrow 10 \times 4$<br>$10 \times 4 \rightarrow 12 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 1000$<br>$\eta = 10^{-4}$<br>500 |
| $12 \times 4 \rightarrow 14 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 500$<br>$\eta = 10^{-4}$<br>500 |
| $14 \times 4 \rightarrow 16 \times 4$<br>$16 \times 4 \rightarrow 18 \times 4$<br>$18 \times 4 \rightarrow 20 \times 4$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 200$<br>$\eta = 10^{-4}$<br>2000 |
| $20 \times 4 \rightarrow 20 \times 4^*$ | Number of samples<br>Learning rate<br>Training iterations | $N_s = 500$<br>$\eta = 5 \times 10^{-5}$<br>200 |

# Colophon

This thesis was typeset with LaTeX $2_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.