Neural Network Evolution Strategy for Solving Quantum Sign Structures

Master Thesis

Ao Chen
Department of Physics, ETH Zurich

Supervised by
Kenny Choo
Prof. Dr. Titus Neupert
Prof. Dr. Manfred Sigrist

September 28, 2020

Acknowledgements

I would like to thank my supervisors, Kenny Choo and Prof. Dr. Titus Neupert. Their ideas and guidance helped me to determine the topic and the main direction of the thesis. They also gave me many valuable advices necessary for me to overcome the difficulties in the research. I also want to thank Nikita Astrakhantsev for his inspiring discussion and useful data. Furthermore, I feel grateful to Tian Gao for helping me through this tough year. Finally, I want to thank Prof. Dr. Manfred Sigrist who accepted to be my supervisor at ETH.

Abstract

The application of neural networks in solving quantum many-body systems has been a hot point in the research of theoretical quantum physics recently. Despite its success in dealing with some simple quantum systems, the neural network methods encounter difficulties in finding correct sign structures in frustrated systems. In this thesis, we use a numerical optimisation method named evolution strategy (ES) to train the network, which is a practice not attempted in previous studies. We show that ES can optimise a sign network as an independent or auxiliary sign structure of the whole wave function. The performance of ES is tested on the J_1-J_2 model on square and pyrochlore lattices. Our method based on ES gives the best neural network result in these frustrated systems compared with previous studies.

Contents

Chapter 1

Introduction

With its great importance and exponential complexity, the quantum many-body problem has been a central issue in the theoretical studies of quantum physics. The difficulty caused by the complicated sign structures of frustrated quantum systems makes it a big challenge for all existing algorithms to obtain reliable results. The quantum Monte Carlo (QMC)[1] will fail in these systems due to the notorious sign problem [2]. The matrix product state (MPS) [3], very successful in low-dimensional systems, is obstructed by the cut dimensionality in 3D. The traditional variational Monte Carlo (VMC) [4] methods like Gutzwiller-projected fermionic wave function (GWF) [5] also suffer from the frustrated sign structures.

Thanks to the rapid development of modern artificial intelligence techniques, artificial neural networks (ANN) [6] have become a competitive numerical approach in various physical problems. As a tool for approximating arbitrary distributions [7], ANN perfectly matches the numerical requirements of modern physicists in many aspects. The applications of ANN in physics includes solving statistical mechanics [8], classifying phases of matter [9, 10], quantum state tomography [11] as well as solving quantum many-body problems [12]. The extraordinary performance of ANN in quantum physics, especially in quantum many-body systems as neural quantum states (NQS), reveals a new possible perspective on the sign problem.

Unfortunately, although ANN gives similar or better results in some simple quantum systems compared with other methods like MPS [12], it does not work well in frustrated systems. For instance, the ANN does not outperform MPS in the 2D J_1-J_2 model which has strong frustration and complicated sign structures [13].

The difficulty ANN encounters possibly comes from the network architecture. The architectures proposed for the non-frustrated systems in previous studies cannot adapt to the new situations in the frustrated systems. In this thesis, we will propose a new architecture for learning the sign structures of quantum systems. It has great ability in approximating some primary sign structures in various models. Different from other architectures proposed in previous NQS papers, the new sign network is non-differentiable and cannot be optimised by previous gradient-based methods like stochastic gradient descent (SGD).

In addition to SGD, there exists lots of other optimisation methods in the community of machine learning, many of them not relying on the gradient information. In this thesis, we adopt one of these methods named evolution strategy (ES) [14] to optimise the variational wave function without the use of gradients. It provides a possible approach to train the sign network we mentioned above.

To demonstrate the application of the sign network trained by ES in solving quantum sign

structures, we test the performance of ES in J_1-J_2 models on both 2D square lattice and 3D pyrochlore lattice. We will use ES to serve as an independent sign structure or an auxiliary sign structure, with the former for simple sign structures and the latter for complicated sign structures. To solve problems in frustrated systems, a configuration product network (CPN) will be proposed to give accurate results in these systems.

The organisation of the thesis is as follows. In Chapter 2, we will start with the explanation of the frustrated models that we will take as examples to show the performance of our methods. Then we will introduce variational Monte Carlo (VMC) in Chapter 3 as the base of many numerical algorithms including the neural network method we are going to use. Some useful methods in VMC will also be discussed in this chapter. After the introduction of these preparation parts, we are ready to enter the development history of NQS in Chapter 4 where we will talk about the progress of NQS and the advantages and disadvantages of different neural network architectures. At the end of Chapter, the new network architecture for learning sign structures will be proposed, which naturally requires the introduction of ES as an optimisation method not relying on gradients in Chapter 5. These are the introductory chapters of the thesis. Beginning from Chapter 6, we will present our numerical results on the models discussed in Chapter 2. The ES method will be used to train independent and auxiliary sign structures in Chapter 6 and Chapter 7, respectively. Finally, we will give the conclusions and outlooks in Chapter 8.

Chapter 2

Model

2.1 Sign Problem

The sign problem is a distinct problem in quantum physics. Starting with a given Hamiltonian H, the energy of a variational quantum state $|\Psi\rangle$ is

$$E = \langle\Psi|H|\Psi\rangle = \sum_{\sigma,\sigma'} \langle\Psi|\sigma\rangle \langle\sigma|H|\sigma'\rangle \langle\sigma'|\Psi\rangle = \sum_{\sigma,\sigma'} \psi_\sigma^* \psi_{\sigma'} H_{\sigma,\sigma'}, \tag{2.1}$$

where $|\sigma\rangle$ is a basis, $\psi_\sigma = \langle\sigma|\Psi\rangle$ and $H_{\sigma,\sigma'} = \langle\sigma|H|\sigma'\rangle$.

If $H_{\sigma,\sigma'} \leq 0$ for all $\sigma \neq \sigma'$ in a specific basis $\{|\sigma\rangle\}$, we must be able to write the ground state with all components ψ_σ real and positive. To show this, consider $\psi_\sigma^* \psi_{\sigma'} H_{\sigma,\sigma'}$ terms in the expression of energy. If $\sigma = \sigma'$, the sign (or phase) of ψ_σ has no effect. If $\sigma \neq \sigma'$, on the other hand, the sign (or phase) of all components should be the same to make sure that all these terms are negative to obtain the lowest possible energy. Up to a global phase, the components are real and non-negative, i.e. $\psi_\sigma \geq 0$.

On the contrary, if there exists $H_{\sigma,\sigma'} > 0$ for some $\sigma \neq \sigma'$, it will be generally impossible to make all $\psi_\sigma^* \psi_{\sigma'} H_{\sigma,\sigma'}$ terms negative, and $\psi_\sigma \geq 0$ is no longer true. Usually, the ground state components can still be real, but we need to find correct signs for different components. It seems that the task of determining the correct signs is just some additional computation cost, but it is actually the most difficult part in the study of quantum many-body physics. This difficulty is what people encounter in various numerical methods. One of the goals of this thesis is to find a systematic way to approach the correct ground state sign structure in quantum systems.

The strict description of the sign problem is given in the language of quantum Monte Carlo (QMC) [1]. For a quantum partition function $Z = \mathrm{tr}(e^{-\beta H})$ where β is the inverse temperature, a specific QMC method named world line algorithm [15] rewrites it as

$$\begin{aligned} Z &= \sum_{\sigma_1} \langle\sigma_1|e^{-\beta H}|\sigma_1\rangle \\ &= \sum_{\sigma_1,\ldots,\sigma_M} \langle\sigma_1|e^{-\beta H/M}|\sigma_2\rangle \langle\sigma_2|e^{-\beta H/M}|\sigma_3\rangle \ldots \langle\sigma_{M-1}|e^{-\beta H/M}|\sigma_M\rangle \langle\sigma_M|e^{-\beta H/M}|\sigma_1\rangle \\ &\approx \sum_{\sigma_1,\ldots,\sigma_M} \langle\sigma_1|(1-\tau H)|\sigma_2\rangle \ldots \langle\sigma_M|(1-\tau H)|\sigma_1\rangle , \end{aligned} \tag{2.2}$$

where M is the number of slices, $\tau = \beta/M \to 0$. We can take the combination of a series of basis $|\sigma_1\rangle, \ldots, |\sigma_M\rangle$ as a classical configuration C associated with a weight $W(C) =$

$\langle\sigma_1|(1-\tau H)|\sigma_2\rangle \ldots \langle\sigma_M|(1-\tau H)|\sigma_1\rangle$, then the expectation value of a quantity A can be written as

$$\langle A\rangle = \frac{\mathrm{tr}(Ae^{-\beta H})}{Z} = \frac{\sum_C A(C)W(C)}{\sum_C W(C)}. \tag{2.3}$$

In this formula the quantum problem is transformed to a classical statistical problem.

However, in classical statistical problems the weights $W(C)$ should be non-negative. If sign problem exists, for some basis σ and σ' there will be

$$\langle\sigma|(1-\tau H)|\sigma'\rangle = -\tau H_{\sigma,\sigma'} < 0 \tag{2.4}$$

because $H_{\sigma,\sigma'} > 0$. Some configurations may have negative weights and the classical methods will be infeasible for evaluating the expectation value in Eq.(2.3).

To evaluate the expectation value with negative weights, it is possible to rewrite Eq.(2.3) as

$$\langle A\rangle = \frac{\sum_C A(C)S(C)|W(C)|/\sum_C |W(C)|}{\sum_C S(C)|W(C)|/\sum_C |W(C)|} = \frac{\langle AS\rangle'}{\langle S\rangle'}, \tag{2.5}$$

where $S(C)$ is the sign of $W(C)$ and $\langle ...\rangle$ means the expectation value with positive weights $|W(C)|$. In a typical system with sign problems, $\langle S\rangle'$ is close to 0. The vanishing denominator means the measurement of $\langle A\rangle$ is very sensitive to the error of $\langle S\rangle'$, such that we require an exponential amount of time to evaluate an accurate result. This complexity is not acceptable in most numerical practices.

In the discussion above, the sign problem appears to be only a numerical difficulty due to a specific expansion method in QMC. However, in other expansion methods like stochastic series expansion (SSE) [16], the problematic negative contributions $\langle\sigma|(-H)|\sigma'\rangle$ still occur and lead to negative weights.

The sign problem is actually an intrinsic difficulty of quantum mechanics that does not depend on specific numerical methods. In Ref.[2], it is proven that the sign problem is NP-hard, which possibly can never be solved in polynomial time.

2.2 Quantum Spin Liquid

The quantum spin liquid (QSL) is a distinct phase of matter exhibiting strong entanglement in quantum many-body systems, especially those systems with sign problems. The J_1-J_2 models to be introduced in this chapter have typical QSL phases, so we will discuss the basic properties of QSL in this section.

Generally, the interactions in the QSL will suppress ordinary magnetic orders and support non-local excitations [17]. Unlike usual disorders in classical systems caused by thermal fluctuations, the disorder in the QSL caused by quantum fluctuations exists even at very low temperatures. This interesting property makes the QSL a central problem in the studies of modern condensed matter physics.

There are many different kinds of QSL. Here we introduce the resonating valence bond (RVB) state proposed by P. W. Anderson in 1987 [18] as an example. To begin with, consider a state as a product of many dimers shown in Fig.2.1, where each dimer is given by the singlet pairing

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle). \tag{2.6}$$

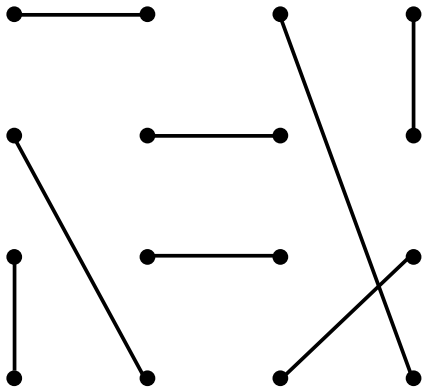


Figure 2.1: Dimer covering of lattice. The connections represent the singlet pairs.

The resonating valence bond state, written as

$$|\Psi\rangle = \sum_C |C\rangle\,, \tag{2.7}$$

is a superposition of states $|C\rangle$ given by dimer products in Fig.2.1. This is a typical QSL that does not show magnetic orders. If all pairs are localised and arranged in a regular manner, it is called a valence bond solid (VBS). In general, however, the dimers in the RVB states does not show regular structures.

The RVB state provides a specific instance for understanding the behaviours of QSL. The suppression of periodic magnetic orders in QSL presents a picture similar to a liquid of disorder spins in comparison to the spin ice with well-organized structures. The state contains many long-range pairs and exhibits non-local entanglement. The non-local excitation as a typical property of QSL can also be achieved by creating unpaired spinons as shown in Fig.2.2.

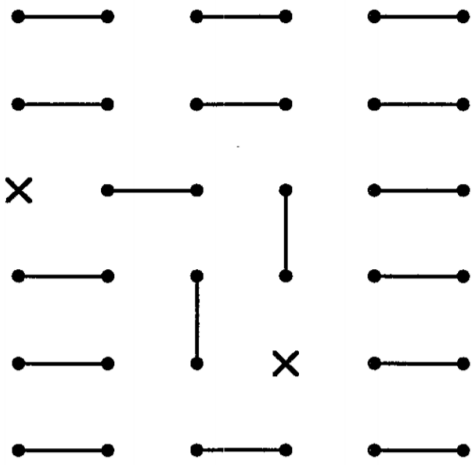


Figure 2.2: Excitation of resonating valence bond state (taken from Ref.[19]). The crosses represent the unpaired spinons.

The concept of QSL will be very helpful for us to comprehend many quantum systems. In the following sections, the QSL phase will play an important role in J_1-J_2 models.

2.3 Square J_1-J_2 model

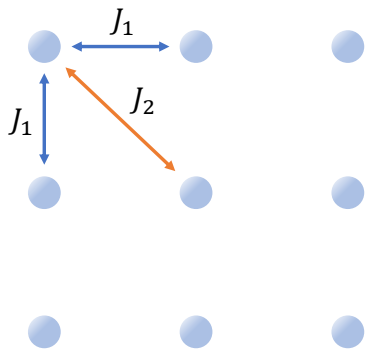


Figure 2.3: Square J_1-J_2 lattice

The spin-$\frac{1}{2}$ J_1-J_2 Hamiltonian is defined as

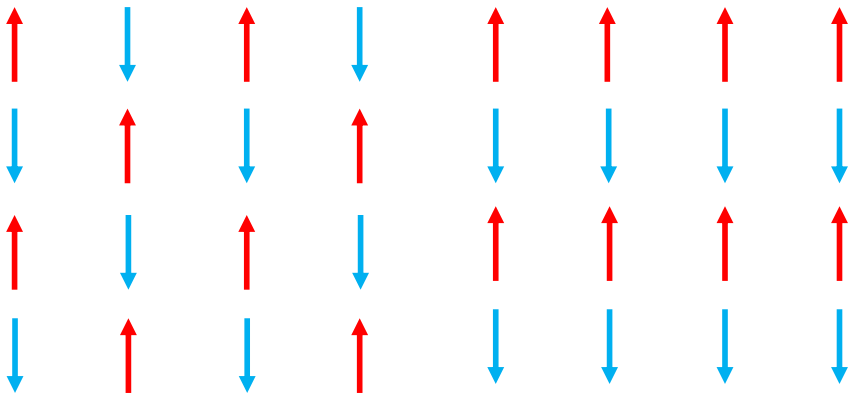
$$H = J_1 \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \mathbf{S}_i \cdot \mathbf{S}_j, \tag{2.8}$$

where $\langle ... \rangle$ and $\langle\langle ... \rangle\rangle$ indicates nearest and next nearest neighbours as shown in Fig.2.3; $\mathbf{S}_i = (S_i^x, S_i^y, S_i^z)$ is the quantum spin operator at site i. We only consider antiferromagnetic interactions in square lattice, so $J_1, J_2 \geq 0$. The spin coupling in the Hamiltonian can be rewritten as

$$\mathbf{S}_i \cdot \mathbf{S}_j = S_i^x S_j^x + S_i^y S_j^y + S_i^z S_j^z = S_i^z S_j^z + \frac{1}{2}(S_i^+ S_j^- + S_i^- S_j^+), \tag{2.9}$$

where $S_i^\pm = S_i^x \pm i S_i^y$. It indicates that the Heisenberg interaction can be interpreted as z-axis coupling and exchange terms. The total number of up spins or down spins is conserved, so this thesis we only consider the sector with equal up and down spins since it is the sector of the ground state.

We start the discussion of square J_1-J_2 model with the simplest situation - the classical systems in two limits $J_2 = 0$ and $J_1 = 0$. The spins in these two systems have a unique choice of direction to achieve the lowest energy, as shown in Fig.2.4.



(a) Checkerboard structure at $J_2 = 0$ (b) Stripe structure at $J_1 = 0$

Figure 2.4: Classical spin configurations at two limit cases

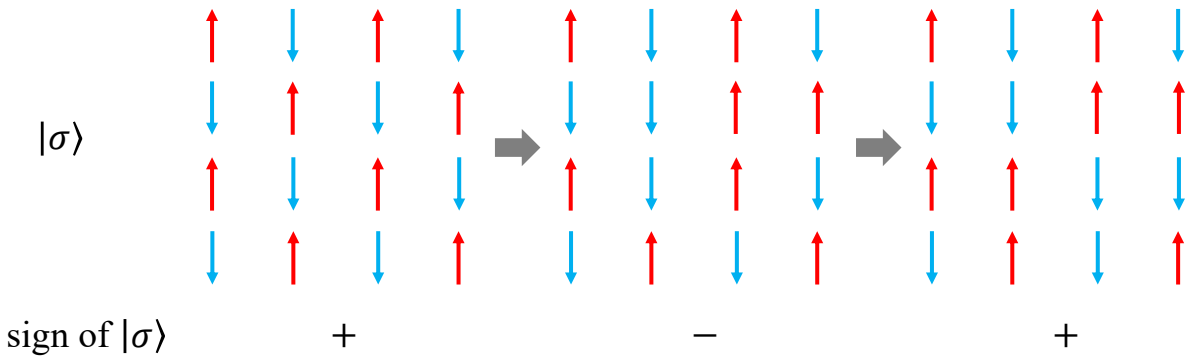


Figure 2.5: The construction of the sign structure in the Heisenberg model

The simple spin structures in the classical systems can be connected with the sign structure of the corresponding quantum problem. It seems that terms like $S_i^+ S_j^-$ are positive off-diagonal elements of the Hamiltonian which will cause a sign problem, but one can solve this difficulty according to the spin structure given by the classical situation as follows.

In the $J_2 = 0$ limit, only nearest neighbour interactions exists and it is usually called the (antiferromagnetic) Heisenberg model. In the Heisenberg model, we can start with the basis $|\sigma\rangle$ corresponding to the classical spin structure in Fig.2.4(a). Every new configuration $|\sigma'\rangle$ generated by the spin exchange terms in the Hamiltonian is rotated to $-|\sigma'\rangle$ so that the off-diagonal terms in Hamiltonian become $\langle\sigma| H (-|\sigma'\rangle) = -H_{\sigma,\sigma'} < 0$. The sign problem is eliminated in this way. The construction of the sign structure is illustrated in Fig.2.5.

Equivalently, we can use the number of up spins in one of the two checkerboard subsets, e.g. the red sites in Fig.2.4(a), to determine the sign of the configuration. The even number means positive and the odd number means negative. If we assume $\psi_\sigma = |\langle\sigma|\Psi\rangle|$ represents the positive amplitude, the ground state Ψ can be written as

$$|\Psi\rangle = \sum_\sigma (-1)^N \psi_\sigma |\sigma\rangle , \tag{2.10}$$

where N is the number of spin up sites in one subset. This is the well known Marshall-Peierls sign rule (MSR) [20]. Similarly, the sign structure of the $J_1 = 0$ limit can also be determined this way. We only need to change the subset according to the stripe configuration (the red sites in Fig.2.4(b)).

The success of eliminating the sign problem in these limiting cases is special. Due to the good structure in these special cases, we can easily construct a basis without sign problem. Generally, however, finding such a construction will take an exponential amount of time since the sign problem is NP-hard.

From now on we turn to the situation with finite J_1 and J_2. In this case the sign problem is non-trivial. In the classical view, the system is frustrated, which means its spins cannot satisfy all antiferromagnetic requirements imposed by the Hamiltonian. As shown in Fig.2.6, the two spins connected by a next neighbour antiferromagnetic interaction should point to opposite directions. Then the third spin as the nearest neighbour of the two previous spins has no possible choice to satisfy the antiferromagnetic condition.

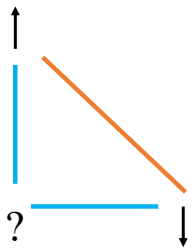


Figure 2.6: Frustration in square J_1-J_2 lattice

The frustration in the classical system has a direct connection with the non-trivial sign problem in the quantum system. Here we only consider 3 spins in J_1-J_2 model with other spins unchanged. Similar to the strategy before, we begin with a configuration and try to determine the sign structure through the exchange of spins in Fig.2.7. However, the existence of both nearest neighbour and next neighbour antiferromagnetic interactions will result in a conflict. After 3 steps of exchanges, the spins come back to the original configuration with an opposite sign. This conflict indicates that it is impossible to find a local basis to eliminate the sign problem in this frustrated system.

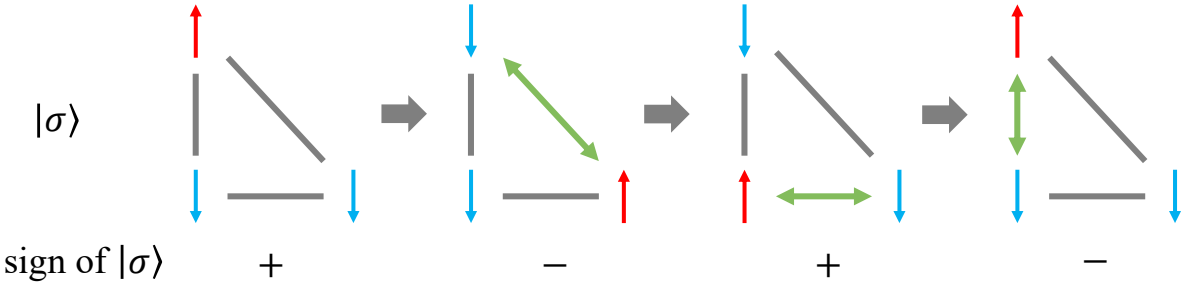


Figure 2.7: Sign problem in J_1-J_2 square lattice

In fact, the square J_1-J_2 model is a typical frustrated system with non-trivial sign problems. There have been numerous studies trying to reveal its properties [21–24]. These papers indicate that the sign structure of J_1-J_2 model can be well approximated by the limiting cases when J_1 or J_2 is weak enough. In a region around $J_2 = 0.5J_1$, the frustration and the severe sign problem may cause a possible QSL phase with very strong entanglement between different sites, but there is still controversy about the existence of the QSL phase. The phase diagram from the newest result in Ref.[24] is shown in Fig.2.8.

In this thesis, we do not attempt to investigate the phase diagram of this model. The aim of introducing this typical system is to compare our new method with some previous numerical results to show its possible advantages and disadvantages.

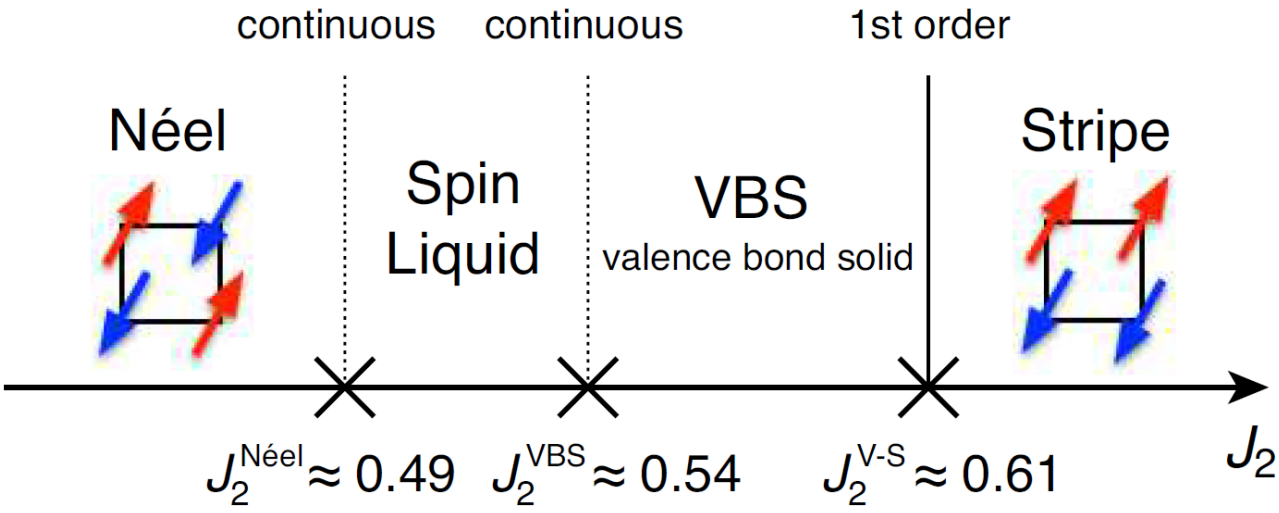


Figure 2.8: Phase diagram of square J_1-J_2 model (taken from Ref.[24])

2.4 Pyrochlore J_1-J_2 model

The pyrochlore lattice is a 3D geometrically frustrated lattice consisting of many tetrahedrons with shared vertices. The J_1-J_2 Hamiltonian in pyrochlore lattice is the same as Eq.(2.8). The nearest and next nearest neighbours in the lattice are shown in Fig.2.9.

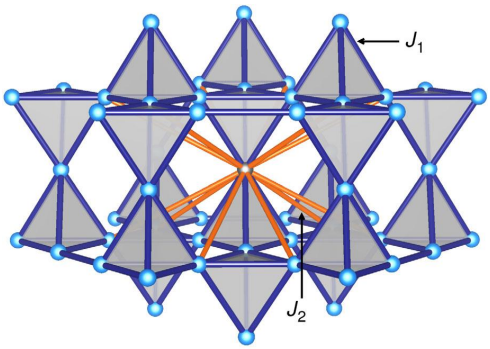


Figure 2.9: Pyrochlore J_1-J_2 lattice (taken from Ref.[25]). The blue and orange lines represent the nearest and next nearest neighbours, respectively.

The most frustrated point of the pyrochlore lattice is the Heisenberg limit at $J_2 = 0$ and J_1 positive. Similar to the square lattice, the frustration of the pyrochlore lattice comes from the triangle structure. As shown in Fig.2.9, the 3D lattice consists of many triangles if we only consider the nearest connections. The frustration is illustrated in Fig.2.10. When two spins in the triangle structure are determined, it will be impossible to find a suitable direction for the third spin under antiferromagnetic interactions. The frustration in the pyrochlore lattice is also associated with a severe sign problem, as illustrated in the square lattice case.

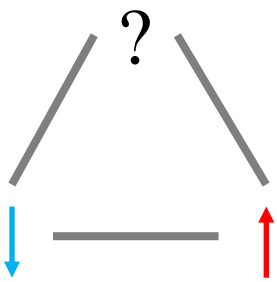


Figure 2.10: Frustration in pyrochlore lattice

In the classical limit without exchange terms, the pyrochlore quantum Heisenberg model becomes

$$H = J \sum_{\langle i,j \rangle} S_i^z S_j^z \tag{2.11}$$

with highly degenerate spin ice ground states given by the famous "two-in, two-out" rule, which means two spins point inwards and two spins point outwards in every sublattice with four spins [26, 27]. Adding a small exchange term, the Hamiltonian becomes

$$H = J \sum_{\langle i,j \rangle} \left[S_i^z S_j^z + \frac{\Delta}{2} (S_i^+ S_j^- + S_i^- S_j^+) \right], \tag{2.12}$$

where Δ represents the strength of the exchange interaction. For small Δ, the perturbation theory indicates the non-local excitations above the spin ice states and reveals the U(1) QSL phase [28].

Many numerical methods have been employed to study the pyrochlore Heisenberg model at $\Delta = 1$. The exact diagonalisation (ED) [29] on very small clusters indicated the QSL phase in the pyrochlore Heisenberg model, but the 3D structure with $N \propto L^3$ spins becomes a big problem for ED on larger clusters. The Gutzwiller-projected fermionic wave function (to be introduced in 3.5) [30, 31], pseudofermion functional renormalisation group method [25] and diagrammatic Monte Carlo simulations [32] were also applied to this model and showed the emergence of the QSL phase, but they suffer from unavoidable bias in the zero temperature and $S = \frac{1}{2}$ regime. The MPS powerful in low-dimensional systems is also obstructed by the 3D structure of the pyrochlore lattice.

In Fig.2.11 we show the phase diagram of the pyrochlore J_1-J_2 model from Ref.[25]. The value of J_1 and J_2 are changed continuously while keeping $J_1^2 + J_2^2 = 1$. The inner ring is the classical phase diagram and the outer ring is the quantum one. The occurrence of a paramagnetic phase in the maximally frustrated region ($J_2 \approx 0$) of the quantum model reveals the possible existence of QSL in this model. In our numerical experiments we will focus on this difficult but interesting region.

As a 3D model with a non-trivial sign problem, the pyrochlore lattice becomes a huge challenge for current numerical methods. The neural network method based on variational Monte Carlo in this thesis, however, can be successfully applied in the zero temperature and $S = \frac{1}{2}$ regime. We will evaluate the performance of our method on the pyrochlore J_1-J_2 model to illustrate its capability in solving cutting-edge problems.

State	Wave vector	Ordering	Classical domain	Quantum $S = 1/2$ domain
Paramagnet				$[345.6° \pm 1.8°, 12.6° \pm 1.8°]$
$\mathbf{k} = \mathbf{0}$	$2\pi(2, 0, 0)$	Coplanar	$(0°, 26.56°]$	$[12.6° \pm 1.8°, 26.56°]$
Planar spiral	$2\pi(k, 0, 0)$	Coplanar	$[26.56°, 145.78°]$	$[26.56°, 151.74° \pm 0.36°]$
Double-twist	$2\pi(\frac{3}{4}, \frac{3}{4}, 0)$	Noncoplanar	$[145.78°, 154.59°]$	$[151.74° \pm 0.36°, 160.83° \pm 0.09°]$
Multiply modulated spiral	$2\pi(\frac{3^*}{4}, \frac{1}{2}, \frac{1^*}{4})$	Noncoplanar	$[154.59°, 158.37°]$	$[160.83° \pm 0.09°, 161.91° \pm 0.09°]$
Cuboctahedral stack	$2\pi(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	Noncoplanar	$[158.37°, 170.30°]$	$[161.91° \pm 0.09°, 171.27° \pm 0.27°]$
Ferromagnet	$2\pi(0, 0, 0)$	Coplanar	$[170.30°, 312.53°]$	$[171.27° \pm 0.27°, 308.61° \pm 0.27°]$
Kawamura	$2\pi(\frac{5^*}{4}, \frac{5^*}{4}, 0)$	Noncoplanar	$[312.53°, 0°)$	$[308.61° \pm 0.27°, 345.6° \pm 1.8°]$

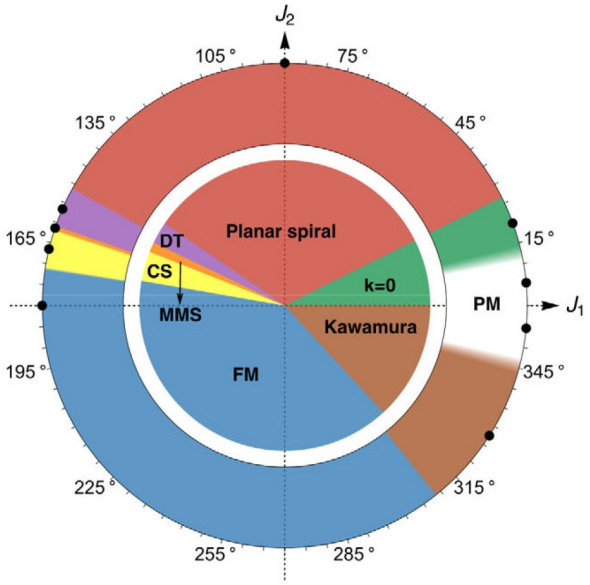


Figure 2.11: Phase diagram of pyrochlore J_1-J_2 model (taken from Ref.[25]). The ordering is labeled as coplanar if there exists a subset of states which are coplanar.

Chapter 3

Variational Monte Carlo

3.1 Classical Monte Carlo Method

The Monte Carlo method in classical systems is the basis of variational Monte Carlo technique we will use in this thesis. In this section, we shall introduce classical Monte Carlo and its application in classical statistical systems.

From now on, we denote σ as configurations of the spin system. For example, in a system consists of 2 spin-$\frac{1}{2}$ sites, σ can take 4 possible values

$$\sigma = \uparrow\uparrow, \uparrow\downarrow, \downarrow\uparrow \text{ or } \downarrow\downarrow . \tag{3.1}$$

Given the Hamiltonian $H(\sigma)$ of a classical spin system, the weight of a specific configuration σ is

$$W(\sigma) = e^{-\beta H(\sigma)} \tag{3.2}$$

and the partition function is

$$Z = \sum_{\sigma} W(\sigma) \tag{3.3}$$

such that the expectation value of observable O can be expressed as

$$\langle O \rangle = \frac{1}{Z} \sum_{\sigma} O(\sigma) W(\sigma). \tag{3.4}$$

In principle, one can sum over all configurations σ to obtain the partition function and the observables, but this is not practical because the total number of configurations grows exponentially with the number of spins. The spin-$\frac{1}{2}$ system with N sites, for instance, has 2^N possible configurations. Consequently, one has to invoke the Monte Carlo method to obtain relatively accurate estimates of observables with an acceptable computational cost.

The spirit of Monte Carlo method is that for a system with very large or even infinite degrees of freedom, one can use suitable samples to estimate its property instead of calculating the exact value through brute-force methods. We will introduce the details of how this powerful idea works below.

Usually, the aim of Monte Carlo method in classical statistical system is to generate samples σ with normalised probability proportional to $W(\sigma)$ such that the observables can be written as

$$\langle O \rangle = \frac{1}{N} \sum_{\sigma'} O(\sigma'), \tag{3.5}$$

where N is the total number of samples and the $\frac{W(\sigma)}{Z}$ factor in Eq.(3.4) is already included through the sampling process. The configurations with more importance (larger $W(\sigma)$) has a higher probability to be sampled and consequently a few samples can represent the important part of all possible configurations very well.

The remaining problem is how to generate samples with the target probability. To achieve this, we begin with a random configuration σ_0 and generate a new configuration from a previous one, i.e. $\sigma_i \to \sigma_{i+1}$. For every transition from σ_i to σ_{i+1}, there is a transition probability $T(\sigma_{i+1}|\sigma_i)$. That is,

$$\sigma_0 \overset{T(\sigma_1|\sigma_0)}{\longrightarrow} \sigma_1 \to ... \to \sigma_i \overset{T(\sigma_{i+1}|\sigma_i)}{\longrightarrow} \sigma_{i+1} \to ... \tag{3.6}$$

We hope that the occurrence probability of every configuration σ at site i can be proportional to $W(\sigma)$ when i is large enough to make the probability stable. Then we can take configurations from σ_i to σ_{i+N} as the required configurations generated by the suitable probability we want. This process is also called Markov chain Monte Carlo (MCMC) as the combination of configurations $\sigma_0 \to \sigma_1 \to \sigma_2 \to ...$ forms a Markov chain [33].

In practice, we take samples with very large distances on the Markov chain. It is because the neighbour configurations in Markov chain are strongly correlated but we prefer uncorrelated samples.

To calculate a suitable transition probability $T(\sigma_{i+1}|\sigma_i)$, the principle of detailed balance is needed. This principle says that if the transition rate satisfies

$$W(\sigma)T(\sigma'|\sigma) = W(\sigma')T(\sigma|\sigma'), \tag{3.7}$$

then the configurations σ will be distributed according to weights $W(\sigma)$ after the markov chain is equilibrated. To understand this equation, we can imagine an ensemble with many spin systems which are already stable and show weights $W(\sigma)$. In every Markov step, the number of systems flowing from configuration σ to σ' will equal to the number of inverse flows if Eq.(3.7) is satisfied, and the weights in this ensemble are unchanged. If the ensemble has not reached the stable state, the weights will change and finally converge to the stable value. The transition process should also be ergodic to ensure that all configurations are sampled.

This problem is still not finished because the total number of new possible configurations σ' is exponentially large and we cannot consider all possible transitions. To solve it, we separate the transition probability into two parts - proposal probability $P(\sigma'|\sigma)$ and acceptance probability $A(\sigma'|\sigma)$, i.e.

$$T(\sigma'|\sigma) = P(\sigma'|\sigma)A(\sigma'|\sigma). \tag{3.8}$$

The number of new configurations σ' possible to be proposed is limited to be very small. Usually we propose new samples by flipping one spin or switching neighbour spins in the previous configuration. Consequently, the number of possible σ' is in line with the system size, which is acceptable. Another advantage of this choice is that $P(\sigma|\sigma') = P(\sigma'|\sigma)$, so the detailed balance equation Eq.(3.7) becomes

$$W(\sigma)A(\sigma'|\sigma) = W(\sigma')A(\sigma|\sigma'). \tag{3.9}$$

Usually we hope the acceptance probability can be as large as possible so that more transitions can happen and the algorithm will be more efficient, so we choose

$$A(\sigma'|\sigma) = \min\left(1, \frac{W(\sigma')}{W(\sigma)}\right) = \min\left(1, e^{-\beta(H(\sigma')-H(\sigma))}\right). \tag{3.10}$$

With this choice Eq.(3.9) is automatically satisfied. The problem of generating suitable samples to evaluate observables is complete.

Algorithm 1 Metropolis algorithm

Choose a system size N
Choose an initial configuration σ
for $i = 0, 1, 2, ...$ **do**
  Propose a new configuration σ' by flipping a spin in σ randomly
  $A \leftarrow e^{-\beta(H(\sigma')-H(\sigma))}$
  Generate a random number $R \in (0, 1)$ with uniform distribution
  if $R < A$ **then**
    $\sigma \leftarrow \sigma'$
  end if
  if $i \bmod \mathrm{N} == 0$ and $i >= 100N$ **then**
    $O_{i/N-100} \leftarrow O(\sigma)$
  end if
end for
$\langle O \rangle \leftarrow \langle O_n \rangle$

The algorithm we introduced above is called the Metropolis algorithm, named after Nicholas Metropolis who proposed it in 1953 [34]. Here we summarise the whole process.

3.2 Variational Monte Carlo

Variational Monte Carlo (VMC) [4] is a common way to solve quantum many-body systems, especially those complicated systems with sign problems. VMC can help to measure the observables in a variational wave function and optimise variational wave function by reducing its variational energy. Although VMC does not tell us the property of the target ground state directly due to the limit of the sign problem, it will obtain a variational state close to the ground state with properties accurate enough for estimating the behaviour of the target state. In this section we will introduce how VMC works.

We begin with the measurement of variational energy in spin lattice systems given the variational wave function. In most cases, the variational wave function can be mathematically formulated as a map $\sigma \to \psi_\sigma$ which takes a spin configuration σ as the input and returns the corresponding component $\psi_\sigma = \langle \sigma | \Psi \rangle$, where $|\Psi\rangle$ is the variational quantum state the map represents.

For example, the quantum ground state with even parity ($\psi_\sigma = \psi_{-\sigma}$) can be written as

$$|\Psi\rangle = (|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle) + \alpha(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle) \tag{3.11}$$

in a system with two spins, where α is the variational parameter controlling the variational wave function. In this case, the components for different configurations are

$$\begin{aligned} \psi_{\uparrow\uparrow} &= \psi_{\downarrow\downarrow} = 1, \\ \psi_{\uparrow\downarrow} &= \psi_{\downarrow\uparrow} = \alpha. \end{aligned} \tag{3.12}$$

Similarly, the Hamiltonian H can also be given by its components

$$H_{\sigma,\sigma'} = \langle \sigma | H | \sigma' \rangle . \tag{3.13}$$

In the transverse field Ising model in a 2-spin system, for instance, the Hamiltonian is

$$H = -\sigma_1^z \sigma_2^z - h(\sigma_1^x + \sigma_2^x), \tag{3.14}$$

where $\sigma_i^{x,y,z}$ are Pauli matrices at site i and h is the transverse field. Then the components $H_{\sigma,\sigma'}$ with different σ and σ' form a matrix

$$\begin{bmatrix} -1 & -h & -h & 0 \\ -h & 1 & 0 & -h \\ -h & 0 & 1 & -h \\ 0 & -h & -h & -1 \end{bmatrix}, \tag{3.15}$$

where the first to forth rows (columns) represent $\uparrow\uparrow$, $\uparrow\downarrow$, $\downarrow\uparrow$ and $\downarrow\downarrow$, respectively.

We can estimate the variational energy of the quantum system as

$$E = \frac{\langle\Psi|H|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\sigma,\sigma'} \langle\Psi|\sigma\rangle \langle\sigma|H|\sigma'\rangle \langle\sigma'|\Psi\rangle}{\sum_{\sigma} \langle\Psi|\sigma\rangle \langle\sigma|\Psi\rangle} = \frac{\sum_{\sigma,\sigma'} \psi_\sigma^* \psi_{\sigma'} H_{\sigma,\sigma'}}{\sum_\sigma |\psi_\sigma|^2}. \tag{3.16}$$

Similar to the statistical system we introduced in the last section, it is usually impossible to sum over all configurations and calculate the exact value, so the Monte Carlo method should be used here. To do so, we rewrite the formula of variational energy as

$$E = \sum_{\sigma,\sigma'} \frac{\psi_\sigma^*}{||\Psi||} \frac{\psi_{\sigma'}}{||\Psi||} H_{\sigma,\sigma'} = \sum_\sigma \frac{|\psi_\sigma|^2}{||\Psi||^2} \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma,\sigma'} = \left\langle \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma,\sigma'} \right\rangle, \tag{3.17}$$

where $\langle ... \rangle$ means sampling σ with a weight $|\psi_\sigma|^2$ like what we did in the last section, and

$$||\Psi|| \equiv \sqrt{\langle\Psi|\Psi\rangle} = \sqrt{\sum_\sigma |\psi_\sigma|^2}. \tag{3.18}$$

We can define the local energy

$$E_{\mathrm{loc},\sigma} \equiv \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma,\sigma'} \tag{3.19}$$

so that the variational energy can be simply written as

$$E = \langle E_{\mathrm{loc},\sigma} \rangle . \tag{3.20}$$

There seems to be a sum over all configurations $\sum_{\sigma'}$ in the formula of $E_{\mathrm{loc},\sigma}$ which will make the computation very heavy. However, the Hamiltonian is usually very sparse and most components $H_{\sigma,\sigma'}$ are 0 for a given σ. The number of nonzero terms is in line with the system size, so the complexity is acceptable. A general observable O can be evaluated by

$$\langle O \rangle = \left\langle \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} O_{\sigma,\sigma'} \right\rangle, \tag{3.21}$$

where $O_{\sigma,\sigma'} = \langle\sigma|O|\sigma'\rangle$.

Here we return to the example in Eq.(3.12) and Eq.(3.15) to illustrate how to obtain the ground state through optimising the variational parameter α. The local energy in this example is

$$\begin{aligned} E_{\mathrm{loc},\uparrow\uparrow} &= E_{\mathrm{loc},\downarrow\downarrow} = -1 - 2h\alpha, \\ E_{\mathrm{loc},\uparrow\downarrow} &= E_{\mathrm{loc},\downarrow\uparrow} = 1 - \frac{2h}{\alpha}. \end{aligned} \tag{3.22}$$

Due to the small system size in this example, we do not have to use the Monte Carlo method. The exact variational energy is

$$E = \sum_{\sigma} \frac{|\psi_\sigma|^2}{||\Psi||^2} E_{\mathrm{loc},\sigma} = \frac{\alpha^2 - 4h\alpha - 1}{\alpha^2 + 1}. \tag{3.23}$$

The parameter α can be tuned to minimise the variational energy. The minimum is at

$$\alpha = \frac{-1 + \sqrt{1 + 4h^2}}{2h} = \tan\frac{\theta}{2}, \tag{3.24}$$

where we have defined $h = \frac{1}{2}\tan\theta$. The corresponding energy is

$$E = \frac{\alpha^2 - 4h\alpha - 1}{\alpha^2 + 1} = -\sec\theta = -\sqrt{1 + 4h^2}, \tag{3.25}$$

which is exactly the ground state energy obtained by diagonalising the matrix in Eq.(3.15).

3.3 Stochastic Reconfiguration

Stochastic reconfiguration (SR) [35] is a gradient-based optimisation method used in VMC. In this section we will introduce why we need SR to optimise the variational wave function and the details of this method.

In most cases, the system size is too large to perform analytic calculations. An alternative approach is to calculate the gradient of energy with respect to all parameters. This quantity represents the direction of energy reduction that can be used to optimise the parameters. In analogy with classical mechanics, we denote the gradient as an equivalent force

$$F_k = \frac{\partial E}{\partial W_k^*}, \tag{3.26}$$

where W_k is a parameter to be optimised; $k = 1, 2, ..., M$ is the index of the parameters. Fortunately, the equivalent force can also be measured by VMC as shown below. Notice that

$$\frac{\partial}{\partial \psi_\sigma} \sum_{\sigma',\sigma''} \psi^*_{\sigma'} \psi_{\sigma''} H_{\sigma',\sigma''} = \sum_{\sigma'} \psi^*_{\sigma'} H_{\sigma',\sigma} = \psi^*_\sigma E^*_{\mathrm{loc},\sigma} \tag{3.27}$$

and

$$\frac{\partial}{\partial \psi_\sigma} \sum_{\sigma'} |\psi_{\sigma'}|^2 = \psi^*_\sigma, \tag{3.28}$$

we have

$$\frac{\partial E}{\partial \psi_\sigma} = \frac{\partial}{\partial \psi_\sigma} \frac{\sum_{\sigma',\sigma''} \psi^*_{\sigma'} \psi_{\sigma''} H_{\sigma',\sigma''}}{\sum_{\sigma'} |\psi_{\sigma'}|^2} = \frac{\psi^*_\sigma}{||\Psi||^2} (E^*_{\mathrm{loc},\sigma} - E) \tag{3.29}$$

and

$$\begin{aligned} F_k = \frac{\partial E}{\partial W_k^*} &= \sum_{\sigma} \frac{\partial E}{\partial \psi^*_\sigma} \frac{\partial \psi^*_\sigma}{\partial W_k^*} \\ &= \sum_{\sigma} \frac{|\psi_\sigma|^2}{||\Psi||^2} (E_{\mathrm{loc},\sigma} - E) \frac{1}{\psi^*_\sigma} \frac{\partial \psi^*_\sigma}{\partial W_k^*} \\ &= \langle E_{\mathrm{loc},\sigma} O^*_{\sigma,k} \rangle - \langle E_{\mathrm{loc},\sigma} \rangle \langle O^*_{\sigma,k} \rangle , \end{aligned} \tag{3.30}$$

where

$$O_{\sigma,k} \equiv \frac{1}{\psi_\sigma}\frac{\partial \psi_\sigma}{\partial W_k}. \tag{3.31}$$

This formula tells us how to evaluate the gradient of energy through Monte Carlo sampling.

Naively, one can optimise the wave function through modifying the parameters a little bit towards the energy decreasing direction, i.e.

$$\Delta W_k = -\gamma \frac{\partial E}{\partial W_k^*} = -\gamma F_k, \tag{3.32}$$

where the derivative is with respect to W_k^* instead of W_k to give the correct direction [36]. γ is a small number to control the changing rate of parameters. In the vector form, the above formula can be written as

$$\Delta \mathbf{W} = -\gamma \mathbf{F}. \tag{3.33}$$

In the language of machine learning, it is equivalent to a widely used optimisation method called stochastic gradient descent (SGD).

The performance of SGD in optimizing variational wave functions is not satisfactory. For a fixed step size γ in the parameter space $\{W_k\}$, the update can correspond to a very large step in the Hilbert space making the optimisation unstable, or a very small change with the risk of being trapped in local minima. Therefore, we should find the optimal change in the Hilbert space $\{\psi_\sigma\}$ and then map this change back to the parameter space $\{W_k\}$.

To work in the Hilbert space, we consider an imaginary time evolution

$$|\Psi'_H\rangle = e^{-\gamma H}|\Psi_H\rangle \approx (1-\gamma H)|\Psi_H\rangle, \tag{3.34}$$

where γ is a small imaginary time. We will show its similarity with γ in Eq.(3.32) later. Every imaginary time evolution step corresponds to a change of the quantum state

$$|\Delta\Psi_H\rangle = |\Psi'_H\rangle - |\Psi_H\rangle = -\gamma H|\Psi_H\rangle = -\gamma \sum_{\sigma,\sigma'} H_{\sigma,\sigma'}\psi_{\sigma'}|\sigma\rangle = -\gamma \sum_\sigma \psi_\sigma E_{\mathrm{loc},\sigma}|\sigma\rangle \tag{3.35}$$

and a change of the normalised quantum state

$$\begin{aligned} |\Delta\tilde{\Psi}_H\rangle &= \frac{|\Psi'_H\rangle}{||\Psi'_H||} - \frac{|\Psi_H\rangle}{||\Psi_H||} \approx \frac{|\Delta\Psi_H\rangle - \mathrm{Re}\,\langle\Psi_H|\Delta\Psi_H\rangle|\Psi_H\rangle}{||\Psi||} \\ &= -\gamma \sum_\sigma \frac{\psi_\sigma}{||\Psi||}(E_{\mathrm{loc},\sigma} - \langle E_{\mathrm{loc},\sigma}\rangle)|\sigma\rangle. \end{aligned} \tag{3.36}$$

This is the change we want to achieve.

On the other hand, when we tune the parameters there is also a change of the quantum state

$$|\Delta\Psi_W\rangle \approx \sum_{\sigma,k} \Delta W_k \frac{\partial\psi_\sigma}{\partial W_k}|\sigma\rangle = \sum_{\sigma,k}\Delta W_k \psi_\sigma O_{\sigma,k}|\sigma\rangle \tag{3.37}$$

and a change of the normalised quantum state

$$|\Delta\tilde{\Psi}_W\rangle \approx \frac{|\Delta\Psi_W\rangle - \mathrm{Re}\,\langle\Psi_W|\Delta\Psi_W\rangle|\Psi_W\rangle}{||\Psi||} = \sum_{\sigma,k}\frac{\psi_\sigma}{||\Psi||}[\Delta W_k O_{\sigma,k} - \mathrm{Re}(\Delta W_k\langle O_{\sigma,k}\rangle)]|\sigma\rangle. \tag{3.38}$$

We hope this change $|\Delta\tilde{\Psi}_W\rangle$ can be equal to the change $|\Delta\tilde{\Psi}_H\rangle$ we want. This is generally impossible, because the dimension of Hilbert space grows exponentially with the system size

while the dimension of parameter space is usually with a polynomial magnitude. There are not enough parameters for us to satisfy all equations. The best result we can reach is to minimise the distance between two changes $|| \, |\Delta\tilde{\Psi}_W\rangle - |\Delta\tilde{\Psi}_H\rangle \, ||$. By solving this we get the target equation

$$\Delta\mathbf{W} = -\gamma S^{-1}\mathbf{F}, \tag{3.39}$$

where S is a matrix with its elements given by

$$S_{k,k'} = \langle O^*_{\sigma,k} O_{\sigma,k'} \rangle - \langle O^*_{\sigma,k} \rangle \langle O_{\sigma,k'} \rangle \tag{3.40}$$

and $\mathbf{F}$ is defined in Eq.(3.30). Compared with Eq.(3.32) there is an additional transformation S^{-1} which implements the optimal optimisation step.

In practice, to make sure that ΔW does not diverge we need to add a small diagonal shift to S, i.e.

$$S^{\text{reg}}_{k,k'} = (1 + \lambda\delta_{k,k'}) S_{k,k'}, \tag{3.41}$$

where λ gives the scale of the shift. In this thesis we choose $\lambda = 0.001$.

There are two difficulties in Eq.(3.39). One is that the number of elements in S is too large for constructing the whole matrix explicitly. Another difficulty is that solving the inverse of such a big matrix is very time consuming. To solve these two problems, we use an iterative solver [37] which can solve for $\gamma S^{-1}\mathbf{F}$ in polynomial time without constructing S explicitly. The expression Eq.(3.40) is enough as long as we obtain all $O_{\sigma,k}$ values.

Below we summarise how to perform VMC optimisation with SR

Algorithm 2 VMC optimisation with SR

Initialise parameters W_k
Choose a batch of 1000 initial configurations $\{\sigma\}$
for $t = 0, 1, 2, ...$ **do**
  Update $\{\sigma\}$ through Metropolis sampling with weights $|\psi_\sigma|^2$
  $E_{\text{loc},\sigma} \leftarrow \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma,\sigma'}$
  $E \leftarrow \langle E_{\text{loc},\sigma} \rangle$ (for estimating the convergence condition)
  $O_{\sigma,k} \leftarrow \frac{1}{\psi_\sigma} \frac{\partial \psi_\sigma}{\partial W_k}$
  $F_k \leftarrow \langle E_{\text{loc},\sigma} O^*_{\sigma,k} \rangle - \langle E_{\text{loc},\sigma} \rangle \langle O^*_{\sigma,k} \rangle$
  $\Delta\mathbf{W} \leftarrow -\gamma (S^{\text{reg}})^{-1}\mathbf{F}$ using iterative solver
  $W_k \leftarrow W_k + \Delta W_k$
end for

3.4 Reweighted Sampling

Reweighted Monte Carlo has been widely used in the research of statistical systems[38] as well as QMC[39] and VMC[40]. A reweighted VMC optimisation is adopted in this thesis to improve the accuracy of the results. Here we use the evaluation of energy as an example to illustrate how it works.

In addition to the usual 2-norm defined in Eq.(3.18), we define the 1-norm of the quantum state $|\Psi\rangle$

$$|\Psi| \equiv \sum_\sigma |\psi_\sigma|. \tag{3.42}$$

In traditional VMC, the energy is given by Eq.(3.16)

$$E = \left\langle \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma\sigma'} \right\rangle_2 , \tag{3.43}$$

where $\langle ... \rangle_2$ means sampling according to $|\psi_\sigma|^2$. When $|\psi_{\sigma'}|$ happens to be far larger than $|\psi_\sigma|$, the $\psi_{\sigma'}/\psi_\sigma$ term results in the instability of energy measurement.

We rewrite the energy formula as

$$\begin{aligned} E &= \sum_{\sigma,\sigma'} \frac{\psi_\sigma^*}{||\Psi||} \frac{\psi_{\sigma'}}{||\Psi||} H_{\sigma,\sigma'} = \frac{|\Psi|}{||\Psi||^2} \sum_{\sigma} \frac{|\psi_\sigma|}{|\Psi|} \frac{|\psi_\sigma|}{\psi_\sigma} \sum_{\sigma'} \psi_{\sigma'} H_{\sigma,\sigma'} \\ &= \frac{|\Psi|}{||\Psi||^2} \left\langle \mathrm{sgn}(\psi_\sigma) \sum_{\sigma'} \psi_{\sigma'} H_{\sigma,\sigma'} \right\rangle_1 , \end{aligned} \tag{3.44}$$

where $\langle ... \rangle_1$ means sampling according to $|\psi_\sigma|$. In the last step we take ψ_σ as a real number. This is the reweighted sampling we will use in this thesis. After the implementation of the reweighted sampling, the evaluation of energy will be stable even if $|\psi_{\sigma'}| \gg |\psi_\sigma|$.

The remaining factor $|\Psi|/||\Psi||^2$ is given by

$$\frac{|\Psi|}{||\Psi||^2} = \frac{|\Psi|}{\sum_\sigma |\psi_\sigma|^2} = \frac{1}{\sum_\sigma \frac{|\psi_\sigma|}{|\Psi|} |\psi_\sigma|} = \frac{1}{\langle |\psi_\sigma| \rangle_1}. \tag{3.45}$$

Finally, the formula of energy in reweighted VMC is

$$E = \frac{1}{\langle |\psi_\sigma| \rangle_1} \left\langle \mathrm{sgn}(\psi_\sigma) \sum_{\sigma'} \psi_{\sigma'} H_{\sigma,\sigma'} \right\rangle_1 , \tag{3.46}$$

where we still take ψ_σ as a real number.

More generally,

$$\langle ... \rangle_2 = \frac{\langle |\psi_\sigma| ... \rangle_1}{\langle |\psi_\sigma| \rangle_1}. \tag{3.47}$$

A useful reweighted quantity in SR is

$$\langle O_{\sigma,k} \rangle_2 = \left\langle \frac{1}{\psi_\sigma} \frac{\partial \psi_\sigma}{\partial W_k} \right\rangle_2 = \frac{1}{\langle |\psi_\sigma| \rangle_1} \left\langle \mathrm{sgn}(\psi_\sigma) \frac{\partial \psi_\sigma}{\partial W_k} \right\rangle_1 . \tag{3.48}$$

A possible advantege of the reweighted sampling is that this method makes it easier to update components with a small amplitude ($P \propto |\psi|^2 \to P \propto |\psi|$). The flatter sampling distribution may lead to a better result.

The reweighted sampling was not used in the studies of NQS before. However, we will find it very helpful in improving the accuracy of VMC optimisation in many cases. For example, in the pair product (PP) state (to be introduce in 3.6), the reweighted sampling can help to reduce the variational energy, as shown in Fig.3.1. The training curve of reweighted method converge to a lower value compared with the traditional method.

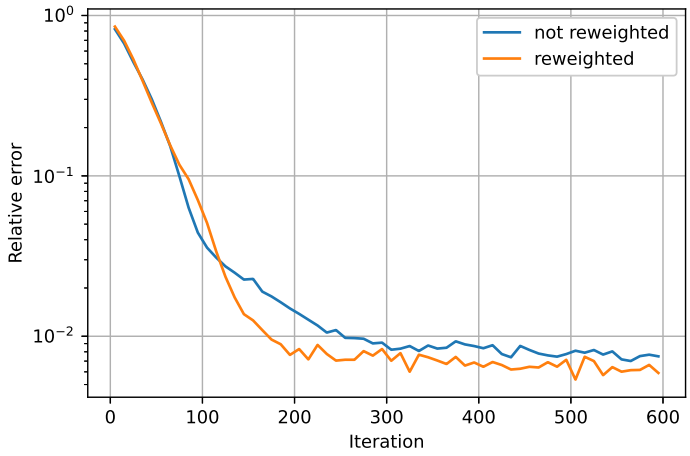


Figure 3.1: Training curve comparison between traditional and reweighted VMC in 6×6 square J_1-J_2 model with $J_2 = 0.5J_1$. The pair product (PP) state is used as the variational ansatz.

To make sure that the results in our numerical tests are reliable when comparing with other methods, the observables like energy are still measured without reweighted to avoid possible bias. The reweighted method is only used during the training process.

3.5 Gutzwiller-Projected Fermionic Wave Function

Gutzwiller-projected fermionic wave function (GWF) is a common variational wave function used in VMC. To construct GWF in quantum spin systems, we should express the spin degrees of freedom through Abrikosov fermions [41]

$$\mathbf{S}_i = \frac{1}{2} \sum_{\alpha,\beta} c_{i,\alpha}^\dagger \boldsymbol{\sigma}_{\alpha\beta} c_{i,\beta}, \tag{3.49}$$

where $\boldsymbol{\sigma} = (\sigma^x, \sigma^y, \sigma^z)$ are Pauli matrices. $c_{i,\alpha}^\dagger$ and $c_{i,\alpha}$ represent the creation and annihilation of a fermion with spin α at site i. In this representation, the spin $|\uparrow\rangle$ and $|\downarrow\rangle$ states at every site have a one-to-one correspondence to the single fermion occupation states $|\uparrow\rangle$ and $|\downarrow\rangle$, respectively. The fermion Hilbert space, however, is too large for describing spin systems because of the two remaining degrees of freedom - the zero occupation state $|0\rangle$ and the double occupation state $|\uparrow\downarrow\rangle$. This redundancy can be removed by a Gutzwiller projection operator

$$\mathcal{P}_G = \prod_i (n_{i,\uparrow} - n_{i,\downarrow})^2, \tag{3.50}$$

where $n_i = c_i^\dagger c_i$ is the fermion number operator at site i. For a fermion state $|\Psi_f\rangle$, the projected state $\mathcal{P}_G |\Psi_f\rangle$ has the correct degrees of freedom for the corresponding spin system. That is the reason why it is called a Gutzwiller-projected fermionic wave function. In practice, this projection is achieved by sampling the states with only $|\uparrow\rangle$ and $|\downarrow\rangle$ at every site.

To construct a good GWF, one usually begins with a mean field Hamiltonian with parameters to be optimised by VMC. This Hamiltonian can be solved to give a mean field wave function as the ground state of the mean field system, which corresponds to a variational wave function

in the spin system after Gutzwiller projection. The parameters in the mean field Hamiltonian actually become the parameters of the variational wave function in the VMC framework.

We describe this process through equations mainly based on Ref.[5]. To begin with, we contract the site and spin index together by

$$\begin{aligned} c_i &= c_{i,\uparrow}, \\ c_{i+N} &= c_{i,\downarrow}, \end{aligned} \tag{3.51}$$

where N is the total number of sites. Then the mean field Hamiltonian can be written as

$$H_{\mathrm{MF}} = \sum_{i,j} t_{ij} c_i^\dagger c_j, \tag{3.52}$$

where t_{ij} is the hopping constant serving as the parameters to be tuned.

If all t_{ij} can be arbitrarily tuned in optimisation, there will be $2N \times 2N$ parameters. To reduce the number of parameters, one usually uses a specific Hamiltonian, a typical one of which is the BCS Hamiltonian

$$H_{\mathrm{MF}} = \sum_{ij\alpha} t_{ij} c_{i,\alpha}^\dagger c_{j,\alpha} + \sum_{ij} \Delta_{ij} c_{i,\uparrow}^\dagger c_{j,\downarrow}^\dagger + h.c. \tag{3.53}$$

with hopping terms t_{ij} (different from the general t_{ij} we mentioned before) and singlet pairing terms Δ_{ij}. It is worth noting that although the Δ_{ij} terms with $c^\dagger c^\dagger$ do not appear in the standard Hamiltonian in Eq.(3.52), one can perform a particle-hole transformation

$$\begin{aligned} c_{i,\uparrow} &\to c_{i,\uparrow}, \\ c_{i,\downarrow} &\to c_{i,\downarrow}^\dagger, \end{aligned} \tag{3.54}$$

to change it into the standard form.

In a compact form, Eq.(3.52) becomes

$$H_{\mathrm{MF}} = \mathbf{c}^\dagger \mathbf{T} \mathbf{c}, \tag{3.55}$$

where

$$\mathbf{c}^\dagger = (c_1^\dagger, c_2^\dagger, ..., c_{2N}^\dagger) \tag{3.56}$$

and

$$(\mathbf{T})_{ij} = t_{ij}. \tag{3.57}$$

The diagonalisation of Eq.(3.55) gives

$$H_{\mathrm{MF}} = (\mathbf{c}^\dagger \mathbf{U}^\dagger)(\mathbf{U}\mathbf{T}\mathbf{U}^\dagger)(\mathbf{U}\mathbf{c}) = \mathbf{d}^\dagger \mathbf{\Lambda} \mathbf{d}, \tag{3.58}$$

where $\mathbf{U}$ is a unitary matrix, $\mathbf{d} = \mathbf{U}\mathbf{c}$ is the fermion operator for the eigenstates, and $\mathbf{\Lambda} = \mathbf{U}\mathbf{T}\mathbf{U}^\dagger$ is a diagonal matrix whose diagonal elements are the energy eigenvalues in the ascending order. This diagonalisation calculation can be easily done by common packages of linear algebra.

The ground state is given by filling N lowest eigenstates, so

$$|\Psi_f\rangle = \prod_{i=1}^{N} d_i^\dagger |0\rangle = \prod_{i=1}^{N} \left(\sum_{j=1}^{2N} c_j^\dagger (\mathbf{U}^\dagger)_{ji} \right) |0\rangle = \prod_{i=1}^{N} \left(\sum_{j=1}^{2N} U_{ij}^* c_j^\dagger \right) |0\rangle . \tag{3.59}$$

The usual basis can be written as

$$|\sigma\rangle = c^\dagger_{R_1} c^\dagger_{R_2} ... c^\dagger_{R_N} |0\rangle \,, \tag{3.60}$$

where R_i can vary from 1 to $2N$. The basis is properly sampled in VMC so that it represents a possible state for spin system and $\mathcal{P}_G |\sigma\rangle = |\sigma\rangle$. Finally we obtain the wave function

$$\begin{aligned} \psi_\sigma &= \langle\sigma|\mathcal{P}_G|\Psi_f\rangle = \langle\sigma|\Psi_f\rangle \\ &= \langle 0|c_{R_N} ... c_{R_1} \left(\sum_j U^*_{1j} c^\dagger_j\right) ... \left(\sum_j U^*_{Nj} c^\dagger_j\right) |0\rangle \\ &= \det \mathbf{V}_\sigma, \end{aligned} \tag{3.61}$$

where the elements of $\mathbf{V}_\sigma$ satisfy $(\mathbf{V}_\sigma)_{ij} = \mathbf{U}^*_{i,R_j}$. The calculation of determinants is slow, but usually we only need the ratio between different wave function components like $\psi_\sigma/\psi_{\sigma'}$. This equals to the ratio between determinants, which can be calculated efficiently if two basis configurations σ and σ' are only different by a flip in a single site or an exchange between two sites.

To make VMC optimisation possible, it is also necessary to calculate $O_{\sigma,k}$ defined in Eq.(3.31). Fortunately, for a determinant we have

$$\frac{1}{\det \mathbf{V}} \frac{\partial(\det \mathbf{V})}{\partial V_{ij}} = (\mathbf{V}^{-1})_{ij} \tag{3.62}$$

which makes the fast calculation of $O_{\sigma,k}$ possible.

As an alternative approach, one can also take all U_{ij} elements instead of t_{ij} as the parameters in VMC optimisation to construct a wave function with many variables while avoiding the calculation cost in diagonalizing the $\mathbf{T}$ matrix.

In addition to the mean field contributions, one also needs to include the correlation factors to express general quantum states. To simplify the calculation, they are defined to be diagonal operators

$$\mathcal{P}(\sigma) = \langle\sigma|\mathcal{P}|\sigma\rangle \tag{3.63}$$

so that

$$\langle\sigma|\mathcal{P}|\Psi_f\rangle = \mathcal{P}(\sigma) \det \mathbf{V}_\sigma. \tag{3.64}$$

A widely-used correlation factor is the Jastrow factor, whose definition in spin system is

$$\mathcal{P}(\sigma) = \exp\left(\sum_{ij} W_{ij} \sigma_i \sigma_j\right), \tag{3.65}$$

where W_{ij} are parameters to be optimised in VMC, and σ_i is the spin configuration at site i.

In many cases we hope the wave function has specific symmetries to obtain specific excited states or increase the accuracy of the ground state, so the symmetry projectors should also be added into GWF. For an Abelian symmetry G with group members $\{\hat{T}_1, \hat{T}_2, ..., \hat{T}_\nu\}$ (ν members in total) and corresponding characters $\{\omega_1, \omega_2, ..., \omega_\nu\}$, the following equation should exist for an eigenstate $|\Psi\rangle$

$$\hat{T}_i |\Psi\rangle = \omega_i |\Psi\rangle \,. \tag{3.66}$$

One can define a symmetry projector

$$\mathcal{L} = \frac{1}{\nu} \sum_{i=1}^{\nu} \omega_i^{-1} \hat{T}_i \tag{3.67}$$

so that for any symmetrised state $|\Psi\rangle = \mathcal{L}|\Psi_f\rangle$ the symmetry condition Eq.(3.66) is automatically satisfied, i.e.

$$\hat{T}_i|\Psi\rangle = \frac{1}{\nu}\sum_{j=1}^{\nu}\omega_j^{-1}\hat{T}_i\hat{T}_j|\Psi_f\rangle = \frac{\omega_i}{\nu}\sum_{j=1}^{\nu}(\omega_i\omega_j)^{-1}\hat{T}_i\hat{T}_j|\Psi_f\rangle = \omega_i|\Psi\rangle. \tag{3.68}$$

In this thesis we only consider ground states in the trivial representation with $\omega_i = 1$, so

$$\mathcal{L} = \frac{1}{\nu}\sum_{i=1}^{\nu}\hat{T}_i. \tag{3.69}$$

In summary, the complete expression for GWF to be used in this thesis is

$$\psi_\sigma = \langle\sigma|\mathcal{P}_G\mathcal{P}\mathcal{L}|\Psi_f\rangle = \frac{\mathcal{P}(\sigma)}{\nu}\sum_{i=1}^{\nu}\det(\mathbf{V}_{\hat{T}_i\sigma}). \tag{3.70}$$

Here $\mathcal{P}(\sigma)$ should have the symmetry implied by $\mathcal{L}$ to ensure the symmetry still exists in the final ψ_σ.

3.6 Pair Product State

As a generalisation of the determinant form we introduced above, the pair product state (PP) [42] is also a type of GWF which can express more possible quantum states. Following the discussion in the last section, we can write a more general mean field Hamiltonian

$$H_{\mathrm{MF}} = \sum_{i,j}(t_{ij}c_i^\dagger c_j + \Delta_{ij}c_i^\dagger c_j^\dagger) + h.c., \tag{3.71}$$

whose solution takes the form

$$|\Psi_f\rangle = \exp\left(\frac{1}{2}\sum_{ij}F_{ij}c_i^\dagger c_j^\dagger\right)|0\rangle, \tag{3.72}$$

where F_{ij} can be solved from the mean field Hamiltonian and $F_{ij} = -F_{ji}$. Different from the approach in the last section where t_{ij} and Δ_{ij} serve as VMC parameters, F_{ij} are directly taken as the parameters to be optimised in most application cases of PP.

After Gutzwiller projection, only the terms with N creation operators remain, so

$$\mathcal{P}_G|\Psi_f\rangle = \frac{1}{(N/2)!}\mathcal{P}_G\left(\frac{1}{2}\sum_{ij}F_{ij}c_i^\dagger c_j^\dagger\right)^{\frac{N}{2}}|0\rangle. \tag{3.73}$$

Here we construct an explicit example to illustrate this state. For $|\sigma\rangle = |\uparrow\uparrow\downarrow\downarrow\rangle$, the wave function component is

$$\psi_\sigma = \langle\sigma|\mathcal{P}_G|\Psi_f\rangle = \frac{1}{(N/2)!}(F_{1\uparrow,2\uparrow}F_{3\downarrow,4\downarrow} + F_{1\uparrow,3\downarrow}F_{2\uparrow,4\downarrow} + F_{1\uparrow,4\downarrow}F_{2\uparrow,3\downarrow}). \tag{3.74}$$

The wave function is a sum over products of pair entanglement factors F_{ij}, so it is called a pair product state. If only singlet pairings are considered, it will be the resonating valence

bond state introduced in 2.2. Consequently, PP is expected to work well in frustrated systems exhibiting a QSL phase.

Generally, the wave function is

$$\psi_\sigma = \langle\sigma|\mathcal{P}_G|\Psi_f\rangle = \frac{1}{(N/2)!}\langle 0|c_{R_N}...c_{R_1}\left(\frac{1}{2}\sum_{ij}F_{ij}c_i^\dagger c_j^\dagger\right)^{\frac{N}{2}}|0\rangle = \mathrm{Pf}(\mathbf{V}_\sigma), \tag{3.75}$$

where $(\mathbf{V}_\sigma)_{ij} = F_{R_i,R_j}$ and $\mathrm{Pf}(\mathbf{V})$ is the pfaffian of matrix $\mathbf{V}_\sigma$. In this thesis we will use PfaPack [43] to calculate pfaffians.

Analogous to the determinant in the last section, the ratio between pfaffians can be calculated efficiently and the derivative can also be obtained through a simple equation

$$\frac{1}{\mathrm{Pf}(\mathbf{V})}\frac{\partial\mathrm{Pf}(\mathbf{V})}{\partial V_{ij}} = -(\mathbf{V}^{-1})_{ij}. \tag{3.76}$$

The complete expression for PP should also contain correlation factors and symmetry projections, so

$$\psi_\sigma = \langle\sigma|\mathcal{P}_G\mathcal{P}\mathcal{L}|\Psi_f\rangle = \frac{\mathcal{P}(\sigma)}{\nu}\sum_{i=1}^{\nu}\mathrm{Pf}(\mathbf{V}_{\hat{T}_i\sigma}). \tag{3.77}$$

Chapter 4

Neural-Network Quantum State

4.1 Artificial Neural Network

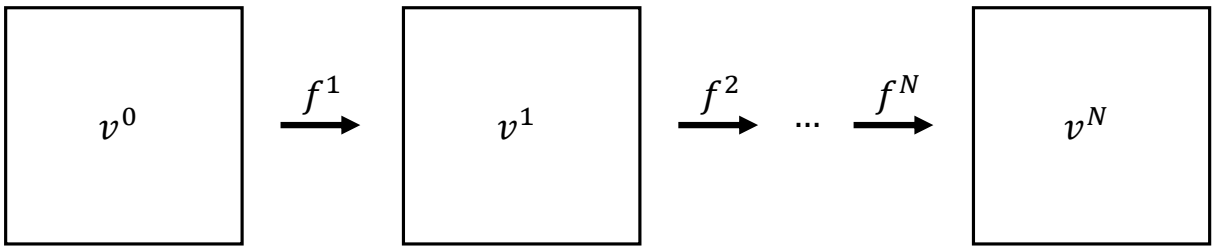


Figure 4.1: Feed forward neural network

Imitating the structure and the function of biological neural networks, the artificial neural network (ANN) has been widely used in various fields of machine learning [6]. As a fast-developing technique in solving complicated problems that cannot be simply summarised by tractable logic, it also proves to be a useful tool in various applications of science subjects, including the quantum many-body physics we study in this thesis. In this section, we will introduce the general structure of feed-forward neural network (FFNN), which is the most common network architecture and is also the architecture we are going to use.

FFNN is a multi-layer structure with non-linear maps connecting two neighbour layers. Each layer takes the output of the last map as its inputs and also produces outputs to serve as the input of the next map, as shown in Fig.4.1. Every layer v^i can be a scalar, vector or tensor. v^0 is the input of the whole network, and the value of each layer after that is given by

$$v^n = f^n(v^{n-1}). \tag{4.1}$$

The last layer v^N is taken as the output of the network. The whole calculation process from input in the first layer to the output in the last layer is called a forward pass.

A common and convenient implementation of the non-linear map f is the fully-connected structure, as shown in Fig.4.2. To illustrate it, we consider v^n and v^{n-1} as vectors with many scalar values. Analogous to the biological neural networks, we call them neurons because they are nodes of the network. The first part of the fully-connected structure is the linear transformation which connects neurons in neighbour layers

$$\boldsymbol{v} \to W\boldsymbol{v} + \boldsymbol{b}, \tag{4.2}$$

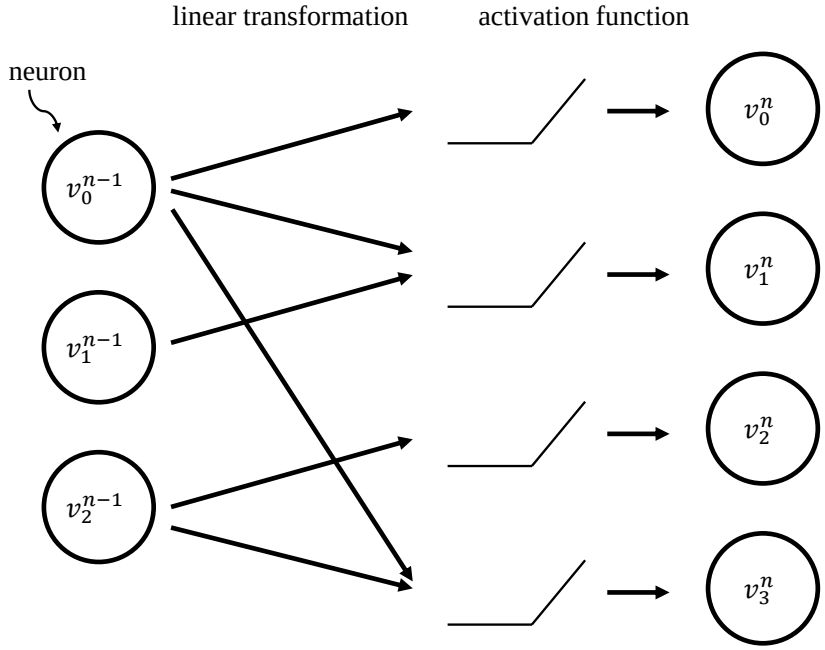


Figure 4.2: Fully-connected structure. The arrows in the linear transformation shows the weights connecting different neurons. Not all possible connections are shown in the figure.

where W and $\boldsymbol{b}$, having different values for different layers, are weight and bias implementing the linear transformation of this specific layer. They are also the parameters we need to optimise. The second part is a non-linear activation function $g(x)$ applying to each output of the linear transformation. The activation function makes sure that the map f can represent a non-linear relation. A common activation function is the rectified linear unit (ReLU) [44] defined by

$$\mathrm{ReLU(x)} = \max(0, \mathrm{x}). \tag{4.3}$$

Combining the two parts, a fully-connected structure can be expressed as

$$f(\boldsymbol{v}) = g(W\boldsymbol{v} + \boldsymbol{b}). \tag{4.4}$$

There are two advantages of this fully-connected structure. First, both the linear transformation and the activation function are easy to implement efficiently, especially on GPU where the computation can be made parallel to achieve an obvious acceleration. Second, the gradients of the network output with respect to the weight and bias are often required to optimise the network. The fully-connected structure in FFNN makes it possible to obtain the gradient of a layer from the gradient of the next layer, so it is possible to gain the gradient information of the whole network through a backward pass from the last layer to the first layer.

4.2 Neural-Network Quantum State

In the section of VMC, we know that a variational wave function can be essentially formulated as a map from any configuration σ to the corresponding component $\psi_\sigma = \langle\sigma|\Psi\rangle$. When ANN serves as this map, it is named a neural-network quantum state (NQS) [12]. In numerical practices, σ is taken as the input of the neural network v^0, and the output of the network v^N is designed to be a scalar representing ψ_σ or $\ln\psi_\sigma$. Usually people choose $v^N = \ln\psi_\sigma$ so that

$$O_{\sigma,k} = \frac{1}{\psi_\sigma}\frac{\partial\psi_\sigma}{\partial W_k} = \frac{\partial(\ln\psi_\sigma)}{\partial W_k} \tag{4.5}$$

is directly the gradient of the network output. All the weights and bias mentioned in Eq.(4.4) are regarded as tunable parameters W_k in Eq.(3.30) that we need to optimise to obatin a better variational wave function.

NQS does not show any symmetry in general, but symmetries are often required to improve the training accuracy of the network. In 4.4, we will introduce CNN to implement translation symmetry into the network architecture. Some other symmetries like rotation symmetry, however, are hard to implement directly through architectures.

Ref.[45] discussed how to implement these symmetries. For every equivalence class of the configurations, a canonical configuration is chosen. In trivial representations with $\omega_i = 1$, the symmetric wave function can be expressed as

$$\psi_{\text{sym}}(\sigma) = \psi(\sigma_{\text{canonical}}), \tag{4.6}$$

where ψ is obtained through the network output v^N. A problem in this formula is that only the canonical configuration in the whole equivalence class is used in calculation. Most of the network information is abandoned. Therefore, we choose a method similar to the symmetry projection discussed in GWF section to create symmetric wave functions. Using the definition of symmetry projection in Eq.(3.69), the wave function becomes

$$\psi_{\text{sym}}(\sigma) = \frac{1}{\nu}\sum_{i=1}^{\nu}\psi(\hat{T}_i\sigma). \tag{4.7}$$

A typical symmetry we will use is the parity symmetry $\sigma \to -\sigma$. The symmetric wave function considering this symmetry is

$$\psi_{\text{sym}}(\sigma) = \frac{1}{2}(\psi(\sigma) + \psi(-\sigma)). \tag{4.8}$$

Having solved the symmetry projection, we turn to the problem during optimisation. While training the network with SR, $E_{\text{loc},\sigma}$ can be easily obtained through ψ_σ. However, calculating $O_{\sigma,k}$ is a hard task in practice as all elements of $O_{\sigma,k}$ form a Jacobian matrix of the neural network, whose parallel implementation is not directly supported in mainstream neural network packages like PyTorch or TensorFlow. To solve this problem in PyTorch, we use BackPACK[46] in which the parallel Jacobian calculation is implemented in both 2D and 3D systems.

4.3 Restricted Boltzmann Machine

The early studies of NQS started with a focus on restricted Boltzmann machine (RBM) [12], a simple shallow structure with extraordinary ability in capturing the entanglement in quantum systems.

The original idea of RBM comes from separating Ising-like spins into two groups - visible units and hidden units [47]. Connections only exists between a visible unit and a hidden unit, but not between two visible units or two hidden units. This allows for more efficient training algorithms because one can fix all visible units and update all hidden units at once, or vice versa. In this process, one does not have to consider the connection between visible units or hidden units. The structure of NQS is shown in Fig.4.3. The architecture of NQS contains many long-range connections. This helps RBM to capture the entanglement and achieve great performance in the quantum systems [48].

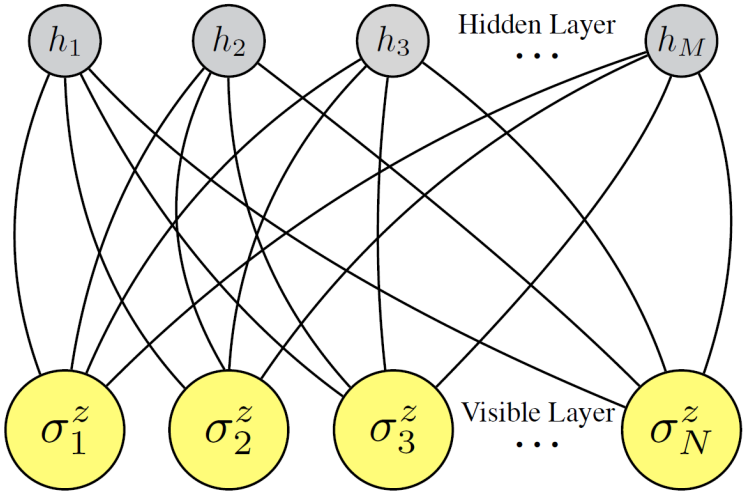


Figure 4.3: Structure of restricted Boltzmann machine (taken from Ref.[12])

When taken as an NQS, RBM can be regarded as a two-layer FFNN. The first layer is the visible layer where we input a configuration σ. Then this layer is followed by a fully-connected structure to connect it with the second layer (hidden layer), with an activation function $g(x) = \ln\cosh \mathrm{x}$. Finally, the sum over all values in the last layer is taken as $\ln\psi_\sigma$. As a simple two-layer architecture, the wave function can be written as a simple formula

$$\ln\psi_\sigma = \sum_{i=1}^{M} \ln\cosh\left(\sum_{j=1}^{N} W_{ij}\sigma_j + b_i\right), \tag{4.9}$$

where N and M are the total number of visible and hidden units, respectively. W_{ij} and b_i are the parameters implementing the linear transformation in the fully-connected structure. All parameters are complex-valued so that RBM can represent wave functions with both amplitude and phase in principle. In the non-frustrated systems where the sign problem does not exist and all components ψ_σ are positive in a suitable basis, the parameters can be real-valued to accelerate the optimisation process.

In numerical practices, translation symmetries are also implemented to increase the accuracy of the optimisation result. In this case RBM is essentially a convolutional neural network that we are going to introduce in the next section.

Despite of its simple architecture, RBM reaches extraordinary accuracy in various quantum systems. In Fig.4.4 we take 2D Heisenberg model as an example. As $\alpha = M/N$ grows, the network architecture becomes more complicated and the relative error compared with the exact ground state energy obtained by QMC[49] improves systematically. RBM reaches an error lower than 10^{-3}, which is a very accurate result compared with EPS[50] and PEPS[51].

Following the success of RBM, Ref.[52] exhibited the relation between RBM and tensor network states, and Ref.[53] shows the limits of shallow RBM architecture in representing general quantum states and introduced a deep Boltzmann machine (DBM) architecture. Some other papers took RBM as the correlation factor in GWF defined in Eq.(3.63), which gives a huge improve to traditional GWF method [54, 55].

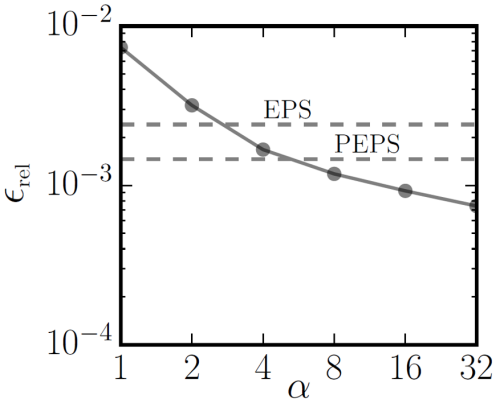


Figure 4.4: RBM result in 10×10 square antiferromagnetic Heisenberg model (taken from Ref.[12]). The x-axis is $\alpha = M/N$. The y-axis is the relative error of variational energy.

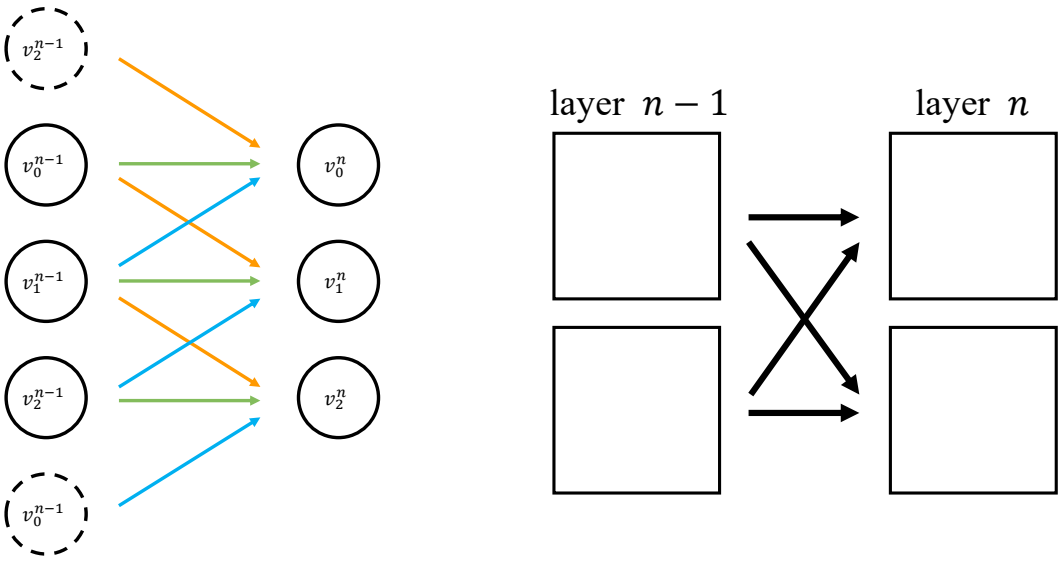
4.4 Convolutional Neural Network

After the studeis of RBM, it is natural to consider deep convolutional neural networks (CNN) widely used in the field of deep learning [56]. CNN is a kind of neural network architecture which directly implements the translation symmetry into its structures to reduce the number of parameters.

As shown in Fig.4.5(a), the arrows represent the weight connecting different neurons. In this simple case where each layer only contains 3 neurons, there are 3 kinds of connections, each represented by a color. The central idea of CNN is that the weights with the same color have shared values, so the number of weights connecting two layers with N neurons are reduced from N^2 to N. Specifically, the linear transformation connecting different layers is called a kernel. One can limit the largest distance of connections (the kernel size) to further reduce the number of parameters. In addition, we choose the periodic boundary condition (the dashed neuron) in Fig.4.5(a) to make sure that the network is exactly translational invariant. Apart from the weight, the bias also has shared values which means there is only one unique bias for different nodes in the same kernel. Similar to fully-connected networks, the activation function is also used after the linear transformations to implement the non-linearity in CNN. The details about how weights are shared can be found in Ref.[56].

The simple network in Fig.4.5(a) only has 1 channel in both layers. A general CNN can have multiple channels. The channel can be viewed as a new dimension of the tensor different from the spatial dimensions. In a CNN with multiple layers, two neighbor layers are still connected by linear transformations. In Fig.4.5(b) we use a simple network with two channels in each layer to illustrate how channels work. Every arrow represents a kernel with shared weights inside as represented in Fig.4.5(a), but different kernels do not share weights between each other. The output of different kernels at the same channel are simply added together. There is still only one bias for each channel no matter how many channels are connected to it through convolutional kernels. The same activation function is applied to different channels.

The translation invariant feature of CNN proves to be very helpful in solving physical problems where symmetries play a central role, including determining phases of matter [9] and solving quantum many-body systems [13].



(a) Weight sharing (b) Channel

Figure 4.5: Architecture of CNN

We will stress on CNN's function as an NQS. In Ref.[13], the CNN architecture shown in Fig.4.6 is taken as the variational ansatz to represent quantum states. The parameters are complex-valued to represent the amplitudes as well as the phases so that the network is able to solve frustrated systems.

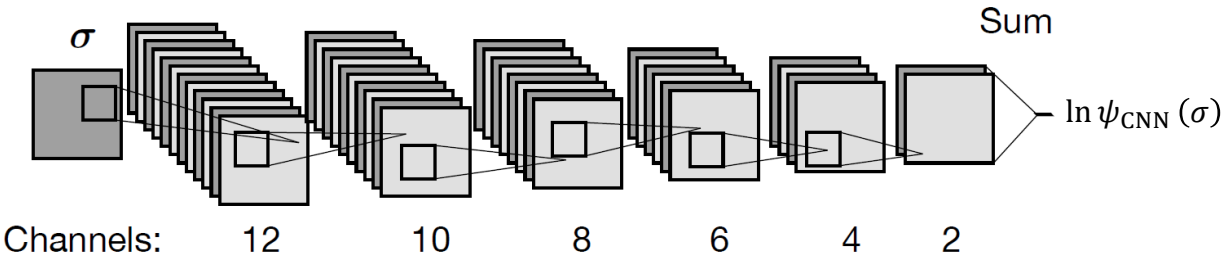


Figure 4.6: CNN architecture as an NQS (taken from Ref.[13])

Instead of the non-frustrated system which can be solved by a simple RBM architecture, the CNN research focuses on frustrated J_1-J_2 systems. In Fig.4.7 the energy difference per site with respect to CNN energy is shown for various methods. The result of CNN outperforms GWF (VMC-Fermions in Fig.4.7) across different J_2 values. However, in the frustrated region around $J_2 = 0.5$, DMRG [23] shows significantly better performance than CNN.

The deep architecture of CNN is efficient in capturing entanglement in quantum systems [57], which helps it to outperform RBM. However, the improvement of CNN mainly comes from the amplitude part. A common problem of both RBM and CNN is that the sign structure cannot be optimised by the usual SR method, although they choose complex-valued parameters to express the phase of wave functions. In the CNN result of Fig.4.7 at $J_2 = 0.5$, for example, the sign structure of the variational wave function is initialised to obey the Marshall sign rule (MSR) before the optimisation. This simple sign structure is not optimised to express a frustrated sign structure during the training process.

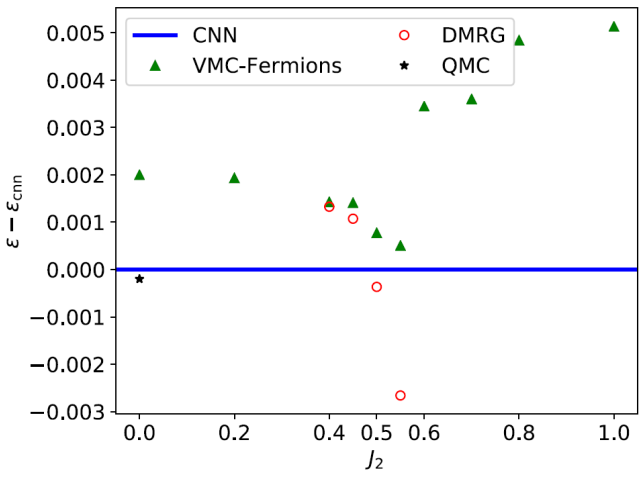


Figure 4.7: Comparison of the variational energy between CNN and other methods in 10×10 J_1-J_2 model (taken from Ref.[13])

In spite of its insufficiency in learning sign structures when SR is used, CNN still inspired people to apply popular architectures in deep learning to quantum systems. A successful example is the use of PixelCNN [58] as an autoregressive network in quantum many-body problems [59] which implements direct sampling and further improves the variational energy.

4.5 Sign Network

In RBM and CNN we introduced above, complex-valued parameters are used so that the network output $\ln \psi_\sigma$ is complex and expresses both the amplitude and the phase. However, this choice proves to be insufficient when we encounter complicated models with sign problems.

Some existing papers introduced a combination of two separate networks - one for the amplitude part and one for the sign part - to solve this problem. The wave function in these papers can be written as

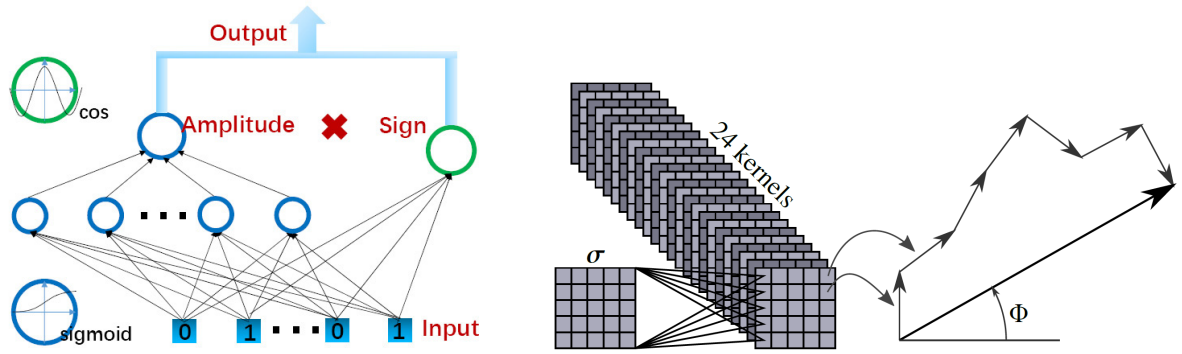
$$\psi(\sigma) = \psi_{\text{amp}}(\sigma)\psi_{\text{sgn}}(\sigma), \tag{4.10}$$

where both the amplitude part $\psi_{\text{amp}}(\sigma)$ and the sign part $\psi_{\text{sgn}}(\sigma)$ are real values given by networks with real parameters.

Usually $\phi(\sigma)$ is real and $\phi_{\text{sgn}}(\sigma)$ only needs to be a sign factor $+1$ or -1 indicating the sign of $\phi(\sigma)$. This choice of discrete output values, however, makes the network non-differentiable and cause problems when we calculate $O_{\sigma,k}$ in Eq.(3.31) during VMC optimisation.

To avoid this problem, there are two choices in previous papers. In Ref.[60], the sign network is a single-layer fully-connected network with a $\cos(x)$ activation function. The output is a continuous real value ranging from -1 to $+1$. Another paper, Ref.[61], uses a single-layer convolutional network to express a phase Φ and $\psi_{\text{sgn}}(\sigma) = e^{i\Phi(\sigma)}$ becomes a continuous phase factor.

Nevertheless, none of these architectures achieve results better than a complex-valued CNN. The attempt to create a continuous sign network introduces some extra complexity into the ansatz. In Ref.[60], the amplitude and the sign network are coupled because the sign network still contains an amplitude. In Ref.[61], complex degrees of freedom are introduced, which is redundant for most ground state wave functions.



(a) Combined network in Ref.[60] (b) Phase network in Ref.[61]

Figure 4.8: Two network choices for $\psi_{\text{sgn}}(\sigma)$

Unlike the networks in Fig.4.8, in this thesis we use a sign network with binary output values, $+1$ or -1, to indicate the sign of wave functions, i.e. $\psi_{\text{sgn}}(\sigma) = \pm 1$. The network architecture, shown in Fig.4.9, is a single-layer CNN using $\cos(x)$ as the activation function. The kernel size is as large as the system size, and the width is chosen to range from 1 to 8. The sign of the sum over the output of the convolutional layer is taken as $\psi_{\text{sgn}}(\sigma)$.

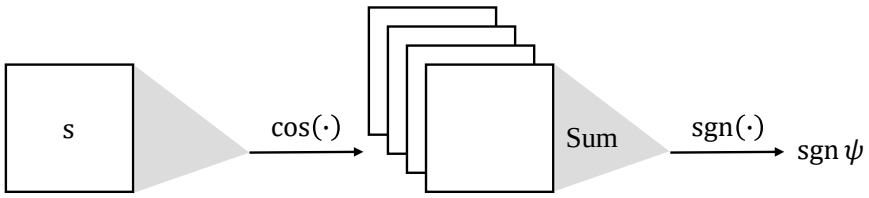


Figure 4.9: Sign network architecture.

This choice, although avoids the redundant complexity, introduces a problem we have mentioned above - this architecture is non-differentiable and cannot be optimised with gradient-based methods like SGD or SR. This is the reason why we need evolution strategy as an optimisation method not relying on the gradient of networks.

To implement the symmetry in the sign network, we input many configurations associated with some symmetry operations like rotations into many identical sign networks. Instead of summing different channels directly as what we do in Fig.4.9, we put all channels corresponding to different configurations together and then sum them up and get the sign. Similar to Eq.(3.69), it is essentially an average over different configurations.

The single-layer architecture we use in the sign network is good at expressing sign structures. For instance, in 2D Heisenberg model on square lattice, one requires only 1 channel with kernel values

$$\frac{\pi}{4}\begin{bmatrix} +1 & -1 & +1 & \dots \\ -1 & +1 & -1 & \dots \\ +1 & -1 & +1 & \dots \\ \dots & & & \end{bmatrix} \tag{4.11}$$

to express the Marshall-Peierls sign rule (MSR)[20] exactly. Another example is the $J_2 \to \infty$ limit in 2D square J_1-J_2 model, whose stripe sign structure corresponds to similar stripe kernel

values

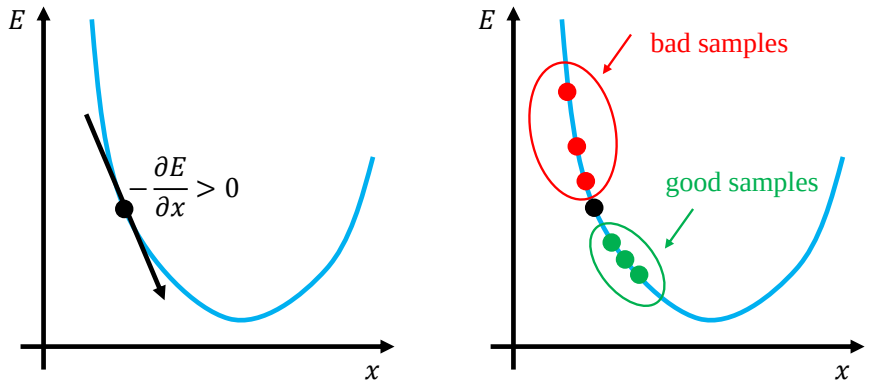
$$\frac{\pi}{4}\begin{bmatrix} +1 & +1 & +1 & ... \\ -1 & -1 & -1 & ... \\ +1 & +1 & +1 & ... \\ ... & & & \end{bmatrix}. \tag{4.12}$$

The similarity between kernel values and Fig.2.4 indicates that the weights in the sign network can tell us some primary information of the sign structures in the quantum system.

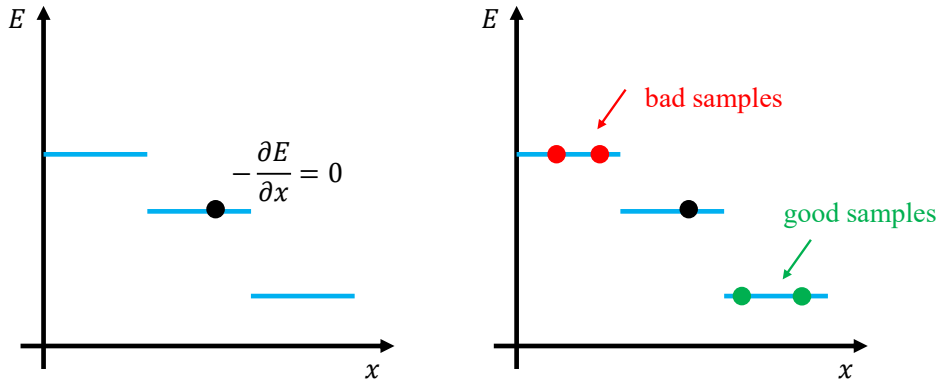
Chapter 5

Evolution Strategy

5.1 Main Idea of Evolution Strategy



(a) Comparison in continuous function



(b) Comparison in discrete function

Figure 5.1: Comparison between SGD and ES in 1-variable optimisation problem. E is the energy we want to minimise. x is the variable optimised to reduce E.

Based on the idea of evolution in biology, the evolution strategy (ES) is an optimisation technique which can work without the gradient information [14]. In this section we will introduce the main idea of ES.

In Fig.5.1 we illustrate the difference between ES and gradient-based optimisation methods like SGD. If the energy E is a continuous function of variable x, SGD can be used to determine

how to optimise x through calculating the gradient descent direction $-\frac{\partial E}{\partial x}$. ES, different from SGD, generates many samples and estimates the direction of energy decreasing through these samples. The good samples indicate the correct direction.

In continuous function, ES does not show much superiority over SGD. Nevertheless, for the cases when energy is a discrete function of variables like what we encountered in sign networks, ES still helps to optimise the variational wave function while SGD does not. This is the reason why we need to introduce ES in this thesis.

Here we introduce a simplified version, simple ES, to demonstrate the central idea of ES. In simple ES, the new samples are generated by a Gaussian distribution for every parameter

$$X_k \sim \mathcal{N}(W_k, \sigma^2), \tag{5.1}$$

where σ is the variance of the normal distribution, X_k is a sample of the new value of W_k. We use $n = 1, 2, ..., \lambda$ to represent λ groups of samples. Every group corresponds to a variational energy $\boldsymbol{X}^{(n)} \to E^{(n)}$ which can be estimated by VMC.

Although it is not the method we are going to use in the numerical experiments of this thesis, simple ES serves as a good starting point to understand the main idea of ES. The detail of simple ES is shown in Algorithm 3. In an ideal situation, the calculation of the energy of all samples is parallel, so the time complexity of ES should be in line with SGD.

Algorithm 3 Update sign network using simple ES

```
Set parameters σ, λ
Initialise all network weights W_k ← 0
for t = 0, 1, 2, ... do
  for n = 1, 2, ..., λ do
    Sample X_k^(n) ~ N(W_k, σ^2) for each k
    Measure corresponding energy E^(n) through VMC
  end for
  Find lowest energy E^(m)
  W_k ← X_k^(m)
end for
```

ES can be easily combined with the VMC framework. In every iteration of the VMC training process, the sign network is optimised by ES while other networks are trained by SR, as shown in Algorithm 4.

Algorithm 4 VMC with ES

```
Initialise all parameters
for t = 0, 1, 2, ... do
  Sample 1000 {σ} with reweighted probability P ∝ |ψ_σ|
  Measure variational energy E (for estimating convergence)
  optimise the sign network using ES
  optimise other networks using SR
end for
Measure target observables without reweighted
```

Here the same group of samples $\{\sigma\}$ can be used to optimise all networks in every iteration. It is because the optimisation of the sign network leaves the probability of sampling unchanged and does not influence the following SR optimisation. Sampling is the most time-consuming part, so this characteristic makes the training process efficient.

5.2 Multivariate Normal Distribution

In the following sections, we are going to discuss a common variant of ES - covariance matrix adaptation evolution strategy (CMA-ES) [62]. This section will begin with the difference between simple ES and CMA-ES in generating samples.

In the process of generating samples, different variables are perturbed by independent Gaussian distributions in simple ES, like the left panel of Fig.5.2. If we go one step further, it is possible to allow different variables to have different variance in new samples, as shown in the central panel of Fig.5.2. However, in most cases ideal sample points should have a distribution in the right panel of Fig.5.2. Different variables should be correlated to give better sample points.

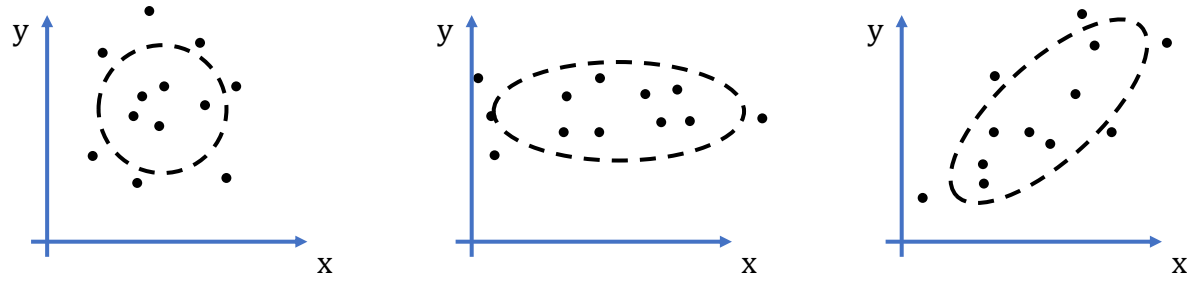


Figure 5.2: ES samples in 3 cases. x and y are parameters to be optimised. The dash line and points shows the sample distribution we want to have and the samples generated.

To achieve the sampling with different variables correlated, it is necessary to introduce the multivariate normal distribution. Slightly different from the independent Gaussian distribution in Eq.(5.1), the multivariate normal distribution can be written as

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{W}, \Sigma) \sim \boldsymbol{W} + \sqrt{\Sigma}\mathcal{N}(0, \mathbf{1}), \tag{5.2}$$

where $\boldsymbol{X}$ is a new group of sample, $\boldsymbol{W}$ is the current weight serving as the mean value of the samples, $N(0, \mathbf{1})$ is the standard Gaussian distribution in the multi-variable case, and Σ is a covariance matrix containing the correlation information between variables. The elements of Σ are given by

$$\Sigma_{i,j} = \langle (X_i - W_i)(X_j - W_j) \rangle = \mathrm{Cov}(X_i, X_j), \tag{5.3}$$

where Cov means covariance.

As a positive definite matrix, the eigendecomposition of Σ is

$$\Sigma = BD^2B^{\mathrm{T}}, \tag{5.4}$$

where B is an orthogonal matrix containing a group of orthonormal eigenvectors of Σ. D^2 is a diagonal matrix whose diagonal elements are the eigenvalue of Σ. In this notation, the multivariate normal distribution becomes

$$\mathcal{N}(\boldsymbol{W}, \Sigma) \sim \boldsymbol{W} + \sqrt{\Sigma}\mathcal{N}(0, \mathbf{1}) \sim \boldsymbol{W} + BDB^{\mathrm{T}}\mathcal{N}(0, \mathbf{1}) \sim \boldsymbol{W} + BD\mathcal{N}(0, \mathbf{1}). \tag{5.5}$$

The last equation is because the standard Gaussian distribution is unchanged under a rotation represented by B^{T}. Eq.(5.5) tells us how to generate samples from the multivariate normal distribution in practice.

5.3 Update of Distribution

Having understood how to generate samples from the multivariate normal distribution, the next problem is how is the distribution updated. We will discuss the update of mean value $\boldsymbol{W}$ and covariance matrix Σ in this section.

In simple ES, the mean value is updated to be the best sample in the last generation. If we sort the samples in an energy ascending order, i.e.

$$\begin{aligned} &\boldsymbol{X}^{(n)} \to E^{(n)} \\ \text{with} \quad &E^{(1)} \leq E^{(2)} \leq ... \leq E^{(\lambda)} \end{aligned} \tag{5.6}$$

then the update of simple ES can be simply written as

$$\boldsymbol{W}' = \boldsymbol{X}^{(1)}, \tag{5.7}$$

where $\boldsymbol{W}'$ represents the new mean value, different from $\boldsymbol{W}$ which represents the previous mean value before the update. An obvious problem in this update rule is that only the best sample is adopted and most information from other samples are abandoned.

As an improvement to this problem, a weighted sum of different samples is used in CMA-ES. The equation for the update of mean values becomes

$$\boldsymbol{W}' = \boldsymbol{W} + \sum_{i=1}^{\lambda} w_i(\boldsymbol{X}^{(i)} - \boldsymbol{W}), \tag{5.8}$$

where $\boldsymbol{X}^{(i)}$ is sorted in the energy ascending order, w_i with $w_1 > w_2 > ... > w_\lambda$ is the weights for different samples. This is the equation telling us how to update the mean value of the distribution.

The remaining part of this section will be about updating covariance matrix - actually this is where the name "covariance matrix adaptation" comes from. We begin with the equation for estimating the original covariance matrix Σ from sample points

$$\Sigma = \frac{1}{\lambda}\sum_{i=1}^{\lambda}(\boldsymbol{X}^{(i)} - \boldsymbol{W})(\boldsymbol{X}^{(i)} - \boldsymbol{W})^{\mathrm{T}}. \tag{5.9}$$

This equation comes from Eq.(5.3), the definition of covariance matrix. To update the covariance matrix, we introduce the same weights in Eq.(5.8) to indicate the importance of different samples, then

$$\Sigma' = \frac{1}{\lambda}\sum_{i=1}^{\lambda} w_i(\boldsymbol{X}^{(i)} - \boldsymbol{W})(\boldsymbol{X}^{(i)} - \boldsymbol{W})^{\mathrm{T}}. \tag{5.10}$$

The figure below shows the update process of the mean value and the covariance matrix. The linear function to be minimised is $f(x, y) = -x - y$ with x, y the x and y axis in the figure. Among 150 samples, the weights are chosen to be $w_{i\leq 50} = 1/50$ and $w_{i>50} = 0$. The mean value

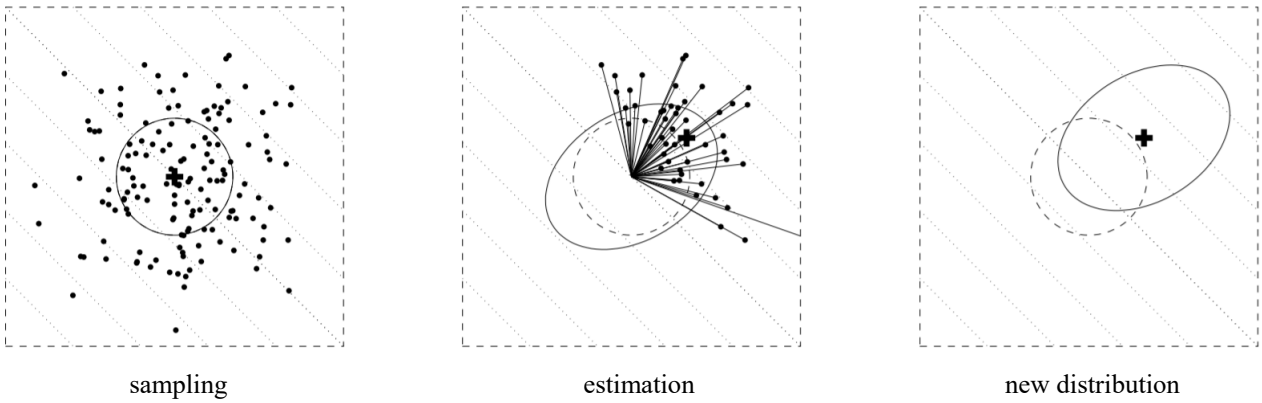


Figure 5.3: Update of covariance matrix (taken from Ref.[62])

is moved towards the direction of reducing f. The covariance matrix is updated in the right panel and the new distribution will generate more samples in the energy descending direction, which is exactly what we want.

Apart from the convariance matrix adaptation we introduced above, CMA-ES also contains many components, including rank-μ-update, rank-one-update and step-size control, all of which takes the information of many previous generations into consideration to fully make use of the previous samples and accelerate the training process. See Ref.[62] for details of CMA-ES.

5.4 Covariance Matrix Adaptation Evolution Strategy

In this section, we are going to give a summary to CMA-ES. With the multivariate normal distribution in sampling and the update of distribution we introduced in the previous sections, CMA-ES becomes an efficient algorithm and one of the most popular variant of ES.

In Fig.5.4 we show that CMA-ES can help to improve the accuracy of ES optimisation in a quantum many-body problem. Compared with simple ES, CMA-ES gives much better training accuracy in the end. Although the 4×4 Heisenberg model is very simple, this comparison still shows the importance of using CMA-ES.

CMA-ES is a sophisticated method with many parameters to control the training rate and ensure its stability, but in most cases one does not have to tune all these parameters. Most parameters can be determined by previous studies and fixed at an optimal value. In this thesis we use the pycma package to implement CMA-ES and the default values in the package are used in the numerical experiments [63]. The only two parameters needed to be specified are the number of samples generated in each iteration λ and the initial variance σ. In this thesis we choose $\lambda = 100$ and $\sigma = \pi/4$.

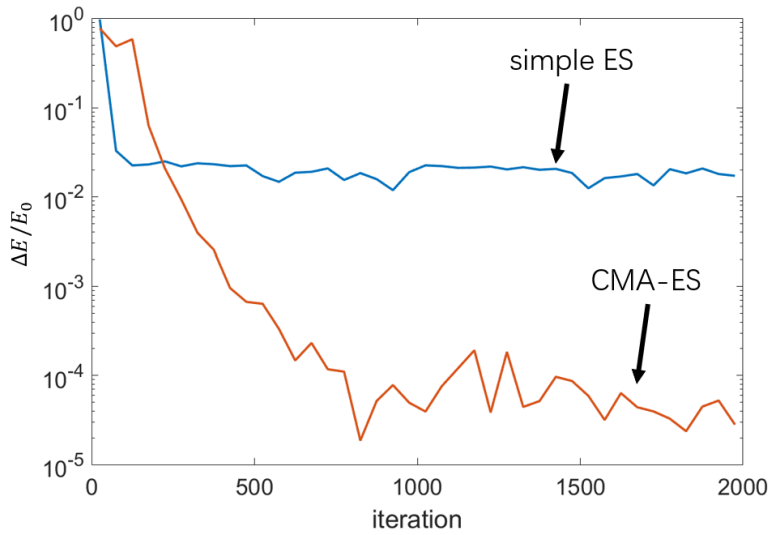


Figure 5.4: Comparison between simple ES and CMA-ES in 2D 4×4 square Heisenberg model. The correct amplitude is already given through ED so that the training focuses on the sign structure.

5.5 Necessity of reweighted in ES

In 3.1, we discuss a reweighted sampling method which can improve the training result in VMC. In ES, the effect of reweighted is more determinant. The application of reweighted in ES is actually the reason we use it in VMC in the beginning.

ES optimises the sign structure through the variational energy. The energy with and without reweighted is given by

$$E = \left\langle \sum_{\sigma'} \frac{\psi_{\sigma'}}{\psi_\sigma} H_{\sigma\sigma'} \right\rangle_2 = \frac{1}{\langle |\psi_\sigma| \rangle_1} \left\langle \mathrm{sgn}(\psi_\sigma) \sum_{\sigma'} \psi_{\sigma'} H_{\sigma,\sigma'} \right\rangle_1 , \tag{5.11}$$

where, as usual, $\langle ... \rangle_2$ is the traditional sampling $P \propto |\psi_\sigma|^2$ and $\langle ... \rangle_1$ is the reweighted sampling $P \propto |\psi_\sigma|$. Due to the fluctuations in the amplitude part, sometimes $\frac{\psi_{\sigma'}}{\psi_\sigma}$ will give very big values. In this case, ES will only focus on the sign of this component while neglecting other components. This will cause a huge difficulty in the training. The difficulty will become more obvious in the frustrated systems where the amplitudes of different components show up to be more irregular. Fig.5.5 shows the training curve with and without reweighted when ES is used to optimise the network. Obviously, the variational energy will be very far from the variational energy if reweighted Monte Carlo is not used.

Consequently, whenever we use ES to optimise a network, the reweighted sampling is always used to ensure its performance.

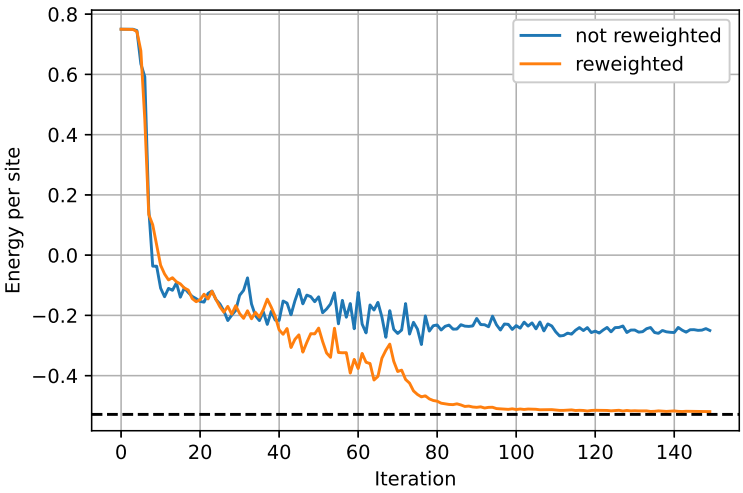


Figure 5.5: ES training curve comparison between traditional and reweighted VMC in 4×4 square J_1-J_2 model with $J_2 = 0.5J_1$.

Chapter 6

Evolution Strategy for Independent Sign Structures

6.1 Architecture of Amplitude Network

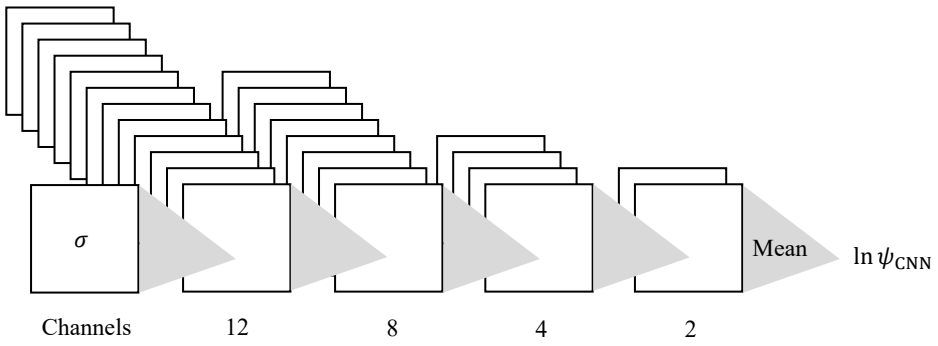


Figure 6.1: Architecture of the amplitude network for square lattice

In this chapter, we are going to use variational wave functions with separate amplitude and sign parts given by a multi-layer CNN trained by SR and a single-layer sign network trained by ES, respectively. We will introduce the architecture of the amplitude network in this section.

The CNN architecture we are going to use for the square lattice in this thesis is shown in Fig.6.1. The kernel sizes are the same as the system size, with channel number 12, 8, 4 and 2 in each layer. The periodic boundary is used in the convolution operation. The first layer uses a lncosh activation function without bias while the following layers use ReLU with bias. This choice ensures the parity symmetry $\psi(\sigma) = \psi(-\sigma)$. The mean value of the last layer is taken as the output $\ln \psi$. The total number of real parameters to be optimised in the network is $148N + 14$ for a system with N sites. The use of CNN in square lattice is direct because the square structure of the lattice fits the shape of CNN kernels. The configuration basis is transformed to a matrix with ± 1 elements. In a simple 2×2 square lattice, for instance, the transformation from a configuration to an input matrix is

$$\begin{matrix} \uparrow & \downarrow \\ \downarrow & \uparrow \end{matrix} \Longrightarrow \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{6.1}$$

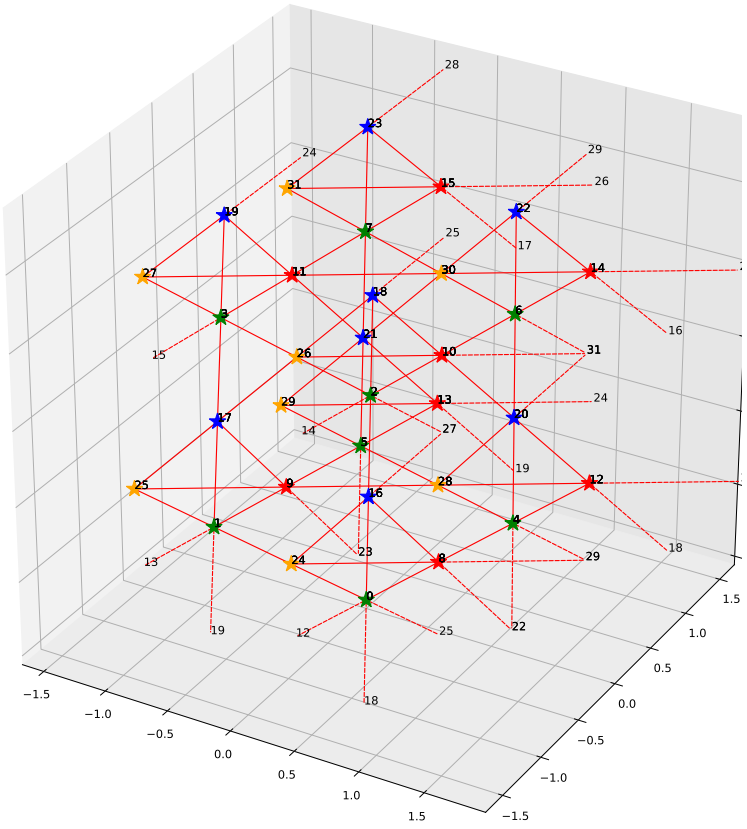


Figure 6.2: Pyrochlore lattice

Different from the square lattice, the configurations in the pyrochlore lattice cannot be directly taken as the input of the network. In Fig.6.2 we show a $2 \times 2 \times 2$ square lattice to illustrate how to deal with the configuration before inputting to the network. For a pyrochlore lattice with N sites, there are $N/4$ cells with 4 sites in each cell. For instance, the sites with number 0, 8, 16 and 24 are in the same cell. They are treated as different channels with the same spatial position in the network. On the other hand, the sites in the same position of different cells have the same color in Fig.6.2. They are the input in the same channel with different spatial positions. This choice makes sure that the translation symmetry is still automatically satisfied in the CNN architecture. It also works for the sign network which also has a CNN architecture.

The architecture of the amplitude network for the pyrochlore lattice is in Fig.6.3. The input layer has 4 channels corresponding to 4 sites in every pyrochlore cell. The following layers have 32 and 16 channels. The arrows in the figure except the last one represent 3D convolutional structures. Other settings including the activation functions are the same as the CNN for the square lattice.

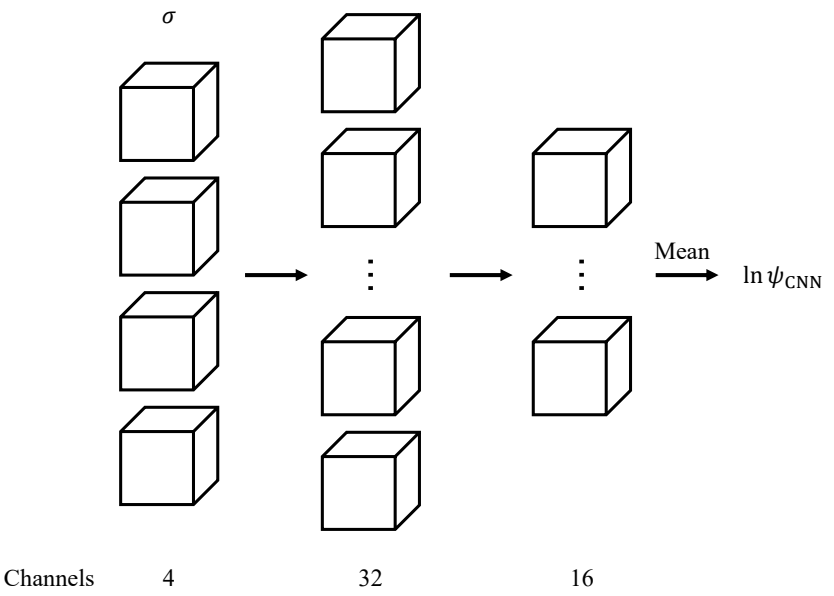


Figure 6.3: Architecture of the amplitude network for pyrochlore lattice. The dots represent some hidden channels.

6.2 Training Strategy

Training a neural network is usually a technical task with many details. They seem to be secondary but actually these details can have deterministic effects on training results. In this thesis we separate the amplitude and the sign parts, so the training strategy about the training sequence of different parts is of great importance.

In Ref.[61] introduced in 4.5, the training starts with the optimisation of the phase network with equal amplitude until convergence, and then two parts are optimised together. Another possible choice is to train the sign part to convergence, and then train the amplitude part to convergence. This procedure is performed back and forth until overall convergence. This choice will also work for non-frustrated systems. These choices are reasonable for non-frustrated systems like square Heisenberg model, because the correct sign structure will lower the variational energy for any form of the amplitude part.

In frustrated systems the situation is different. In Fig.6.4 we show the variational energy obtained by exact diagonalisation (ED) in 4×4 square lattice with $J_2 = 0.5$ which is a typical and simple frustrated system. When the amplitudes of components are equal or best for MSR, MSR will give lower variational energy compared with the ground state (GS) sign structure. This is distinct from the non-frustrated Heisenberg model where the GS sign structure always gives the best energy. In frustrated J_1-J_2 models, GS sign structure will give better energy only when the amplitudes are also GS amplitudes. If the sign part and the amplitude part are trained separately, the system will first converge to MSR with equal amplitude (the first arrow in the picture) during the optimisation of the sign part, and then to best amplitude for MSR (the second arrow) in the amplitude part training. After that, tuning the sign part again will not achieve the GS sign structure because its energy is slightly higher if the amplitude is fixed. The optimisation will stop when it reaches the local minimum at MSR.

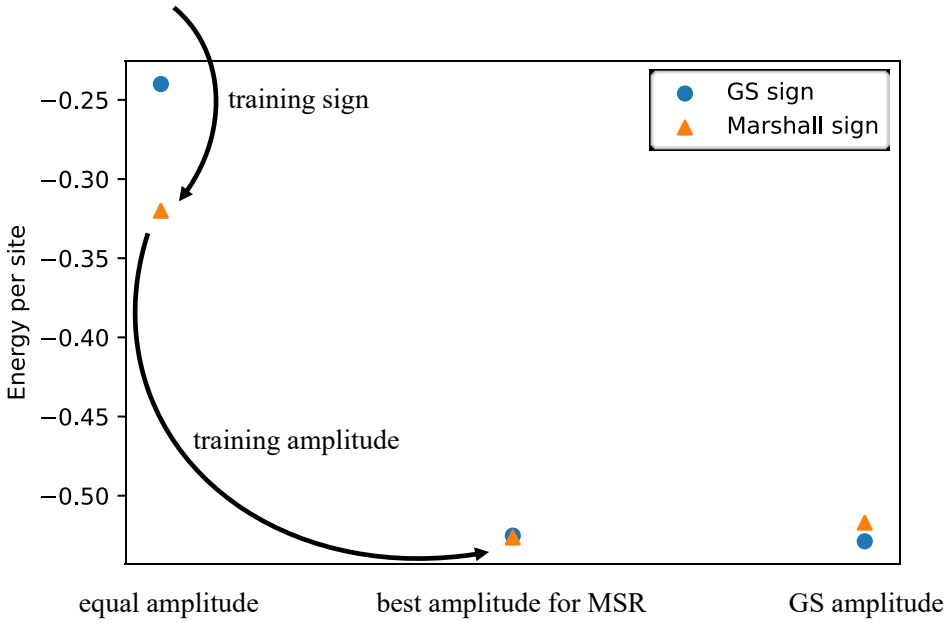


Figure 6.4: The variational energy for different amplitude parts and sign structures in square 4×4 lattice with $J_2 = 0.5$. A possible problematic training path is shown if the amplitude part and the sign part is trained separately.

The strategy in Ref.[61] can overcome this problem because the two parts are trained together after the first step of training the sign part. In this thesis, we use a slightly different training strategy. We will train two parts of the wave function together from the very beginning. This proves to be effective in solving some frustrated models.

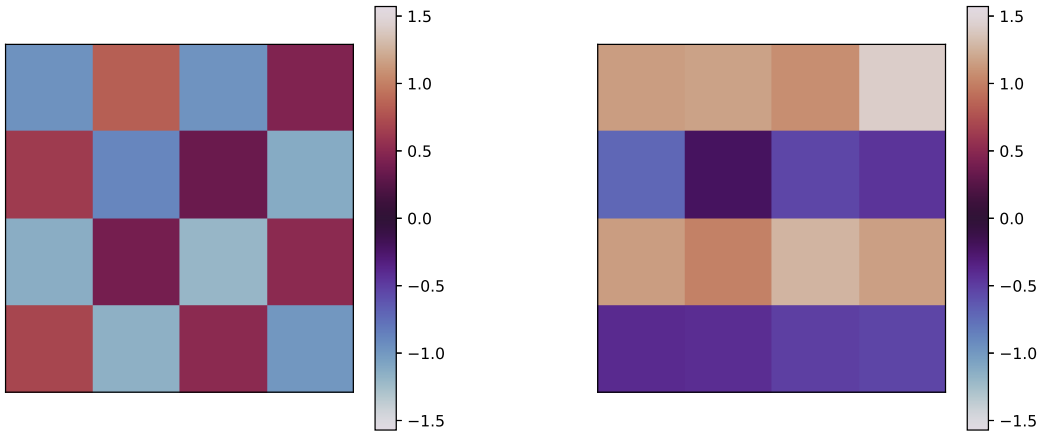
6.3 2D 4×4 Square Lattice

The 2D 4×4 square lattice is a simple system with frustrated sign structures. There are only 16 sites in the system so that the ED result can be easily obtained. In the last section we have used the ED data to determine the training strategy we are going to use. In this section, we will give the main numerical results in this system.

In square lattice, J_1 is fixed to 1 and we vary J_2 to go over the phase space. The kernel values of the sign network show an obvious pattern of the corresponding sign structures for different J_2 values.

When the sign network only has 1 channel and $J_2 = 0$ or $J_2 \to \infty$ (here we choose $J_2 = 1$), ES can optimise the kernel values of the sign network to fit the theoretical result Eq.(4.11) and Eq.(4.12) to a great accuracy, as shown in Fig.6.5. The output of the sign network is invariant if a weight value is added of subtracted by π. Consequently, in this and the following pictures the kernel values are modified into the interval $[-\pi/2, \pi/2]$.

The pattern learned by the network has a close relation with the sign structure of the system. The checkerboard and the stripe pattern are exactly the pattern people usually use to represent the sign structures at $J_2 = 0$ and $J_1 = 0$, respectively. Consequently, the sign network trained by ES can be a useful method for us to have a basic understanding of the quantum sign structures.



(a) $J_2 = 0$ (b) $J_2 = 1$

Figure 6.5: Kernel patterns of the sign network (1 channel).

A single-layer sign network with 1 channel is not enough for the frustrated region around $J_2 = 0.5$. Here we use 4 channels so that the sign network is able to express the more complicated sign structure in the frustrated case. The patterns of the kernels at $J_2 = 0.5$ are in Fig.6.6. The complexity of the frustrated sign structure makes the pattern not as regular as the previous patterns, but we can still find some clues of checkerboard patterns (the last two channels) and stripe patterns (the first two channels). The mixture of different patterns is exactly what we expect in the frustrated region of the J_1-J_2 model.

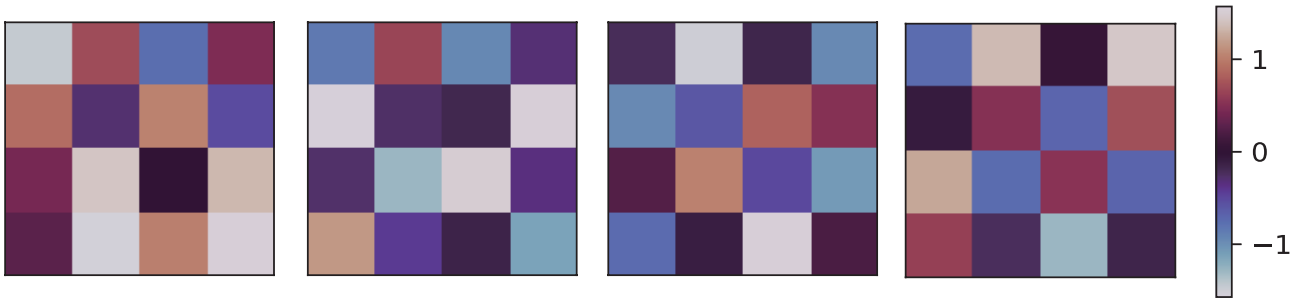
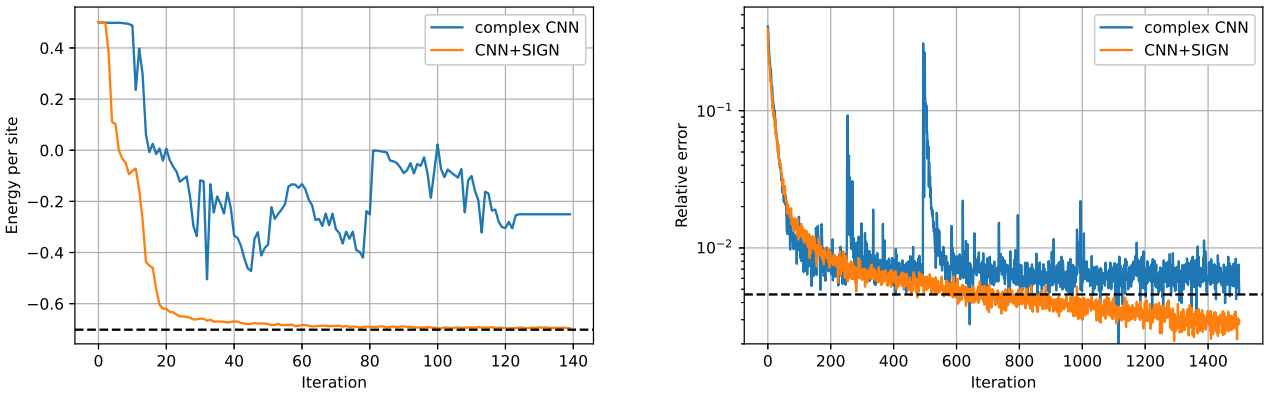


Figure 6.6: Kernel patterns of the sign network (4 channels) at $J_2 = 0.5$

Furthermore, we compare the learning curve of our method with complex-valued CNN. As Fig.6.7 indicates, complex-valued CNN does not learn the sign structures beyond what we impose before training. In previous papers using complex-valued parameters including Ref.[12] and Ref.[13], the network training will be pretty tough in Heisenberg model ($J_2 = 0$) if one does not impose the correct sign structure before training. Usually the network cannot converge to the ground state, as the blue curve in Fig.6.7(a) shows. This problem does not exist if we use a sign network trained by ES to express the sign structure. In this case, the variational energy converges to the ground state energy very fast. For the frustrated sign structure, the situation is similar. In Fig.6.7(b), we impose MSR before the training. MSR is close to the ground state sign structure but not the same. The dash line shows the limit one can reach within MSR. Our network trained by ES goes beneath this limit significantly, while the complex-valued CNN can only reach the MSR limit. In summary, the complex-valued networks widely adopted before can only be trained to express a good amplitude. In order to train the network to adapt to correct sign structures, some other methods like ES in this thesis are necessary.



(a) $J_2 = 0$ (b) $J_2 = 0.5$

Figure 6.7: Comparison between complex-valued CNN and ES in 4×4 square lattice. The results of complex-valued CNN are obtained through the NetKet package [64]. (a) Initial sign structure is all positive. The dash line shows the ground state energy obtained by exact diagonalisation. (b) Initial sign structure is MSR. The dash line indicates the best variational energy if MSR is imposed.

The relative error of variational energy for different J_2 values is shown in Fig.6.8. The error is lower than 10^{-4} across different J_2 values, which is a very good result for frustrated models. The rotation symmetry with 4 group members is used to increase the accuracy of the result, except for $J_2 = 1$ where the symmetry does not help to increase the accuracy.

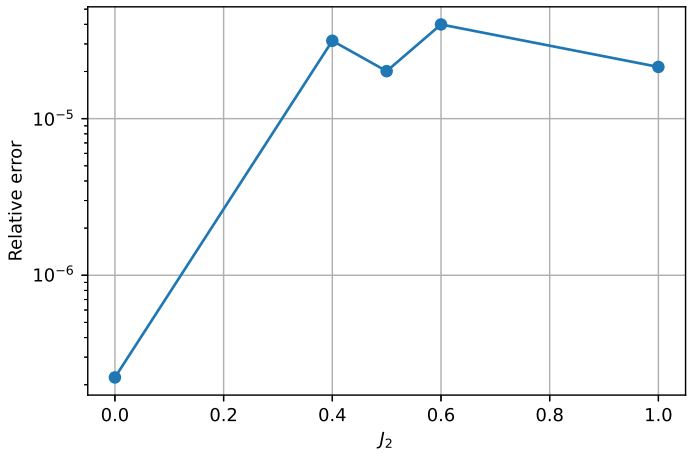
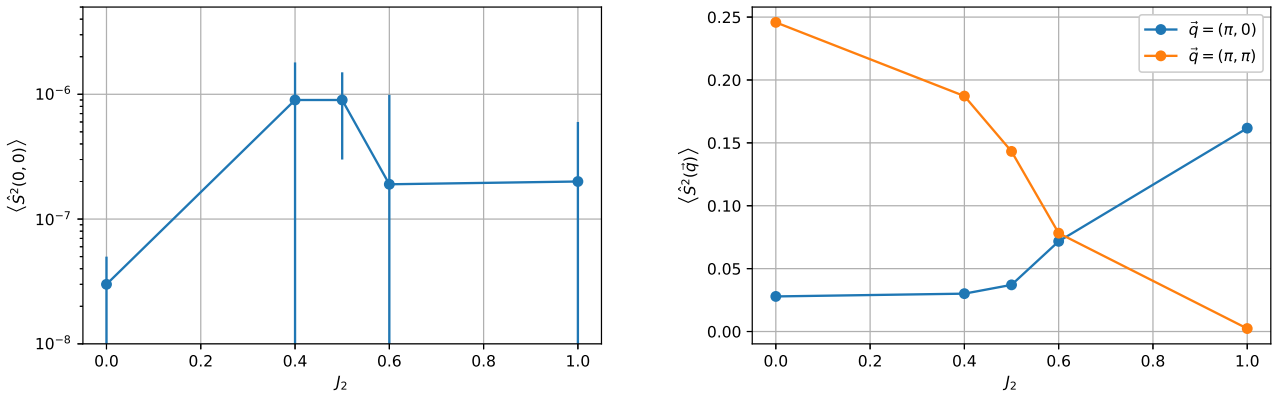


Figure 6.8: Relative error of variational energy in 4*4 square lattice. J_1 is fixed to be 1. The sign network has 4 channels.

We also measure the structure factor, defined as

$$S^2(\boldsymbol{q}) = \frac{1}{N(N+2)} \sum_{i,j} \langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle \, e^{i\boldsymbol{q}\cdot(\boldsymbol{r}_i - \boldsymbol{r}_j)}. \tag{6.2}$$

Its values when $\boldsymbol{q} = (0,0)$, $(0,\pi)$ and (π,π) are shown in Fig.6.9.



(a) $\boldsymbol{q} = (0, 0)$ with error bar

(b) $\boldsymbol{q} = (0, \pi)$ and $(\pi, 0)$

Figure 6.9: $S^2(\boldsymbol{q})$ for different $\boldsymbol{q}$ values

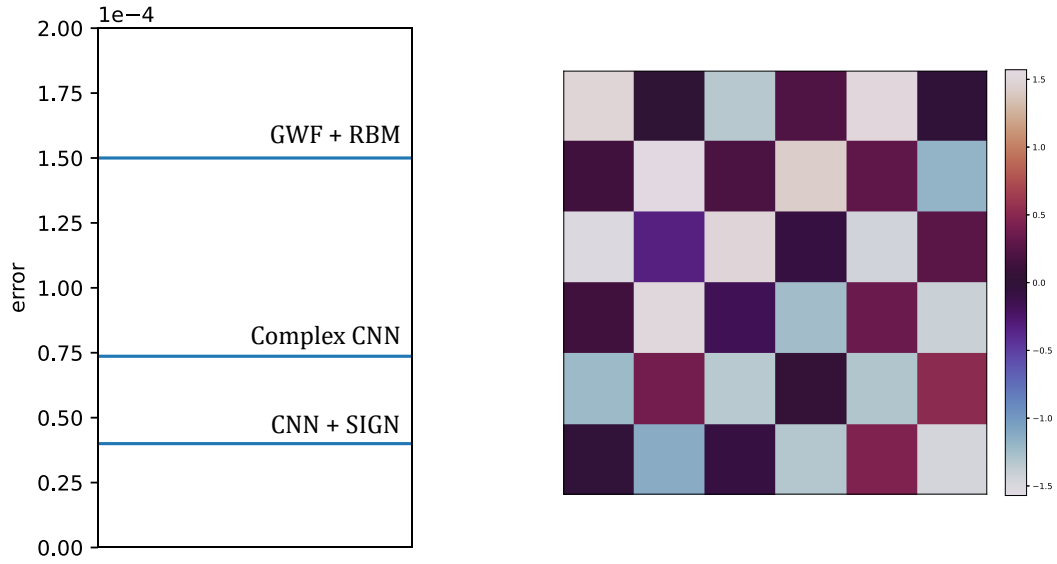
The ground state is in the $S^2(0,0) = 0$ sector. The result in the left panel shows that $S^2(0,0)$ is very close to 0 for all J_2 values. This means that this architecture gives wave functions which almost fall into this correct sector. Furthermore, $S^2(\pi,\pi)$ tends to be larger if the checkerboard sign structure is dominant and $S^2(0,\pi)$ is larger if the stripe structure is dominant. The change of S^2 shows the tendency we expect. As J_2 increases, $S^2(\pi,\pi)$ gets smaller while $S^2(0,\pi)$ gets larger. The sign structure goes smoothly from the checkerboard structure to the stripe structure and they cross at the frustrated region around $J_2 = 0.6$, as we expect. This validates that the sign network has learnt correct sign structures under the help of ES.

6.4 2D 6×6 Square Lattice

The 4×4 square lattice is very simple. The small size can reduce the complexity of its sign structure. To investigate the performance of ES, it is necessary to go to a larger lattice. In this section we try to use similar methods to 2D 6×6 square lattice. This system, although still not big enough to go beyond the applicable range of ED, is large enough to show some important differences.

From now on, we will abbreviate the sign network trained by ES as SIGN. We do not name it as ES because all abbreviaions like GWF and RBM are named by the form of the variational wave functions but not the optimisation method. The wave function in our current method is thus CNN+SIGN.

We begin with the discussion about $J_2 = 0$ case. It corresponds to Heisenberg model with only nearest interactions. The comparison of the relative error and the pattern of the sign network is shown in Fig.6.10. The sign network expresses the correct MSR, and the energy is the best compared with other neural network methods. This is because our network uses most variational parameters. Another possible reason is that the variational wave function in our method is real. The elimination of redundant complex degrees of freedom may be helpful in improving the variational energy.



(a) Relative error of variational energy. GWF+RBM result is from Ref.[55], and complex CNN result is from Ref.[13]

(b) Pattern of the sign network

Figure 6.10: ES result in $J_2 = 0$ square 6×6 lattice

The performance of ES at $J_2 = 0$ is satisfactory as the sign structure in this model is very simple. It will be totally different when we go to the frustrated region. Here we use $J_2 = 0.5$ as an example. In Fig.6.11 we show the variational energy obtained through different methods. The CNN+SIGN combination, although outperforms complex-valued CNN, does not get better than the MSR limit. This means that the sign network does not obtain result beyond the simple sign structure given by MSR, and it does not adapt to the important frustrated sign structure. This behavior is different from the situation in small 4×4 lattice where our method gets a result better than MSR limit. For a larger system size like 10×10, this problem still exists.

In Ref.[61] introduced in 4.5 where gradient-based method is adopted to optimise a phase network there is a similar result that the network can only be trained to express MSR in frustrated regions. The reason behind this is that the correct sign structure becomes harder to be expressed by a simple sign network when the system size goes up.

The similar results based on totally different optimisation methods indicate that the problem may be the ability of variational wave function in expressing the correct sign structure. To verify this, we attempt to impose a good amplitude part very close to the ground state amplitude given by PP and train the sign network independently. As expected, the sign network cannot be optimised to express the correct sign structure. This inspires us to use some new network architectures to express the frustrated sign structure, which will be discussed in the next chapter.

The difference between non-frustrated systems and frustrated systems come from the relation of the amplitude part and the sign part. In non-frustrated systems, the correct sign structure will always give the best energy compared with other sign structures for all possible amplitude distributions. The two parts are relatively separated and can be well approximated by two separate networks. However, in frustrated systems some sign structures different from the ground state sign structure can give the best energy for some amplitude distributions, as shown in Fig.6.4. The amplitude part and the sign part should work together to give the best possible energy. An artificial separation is possible to make the sign part very irregular and

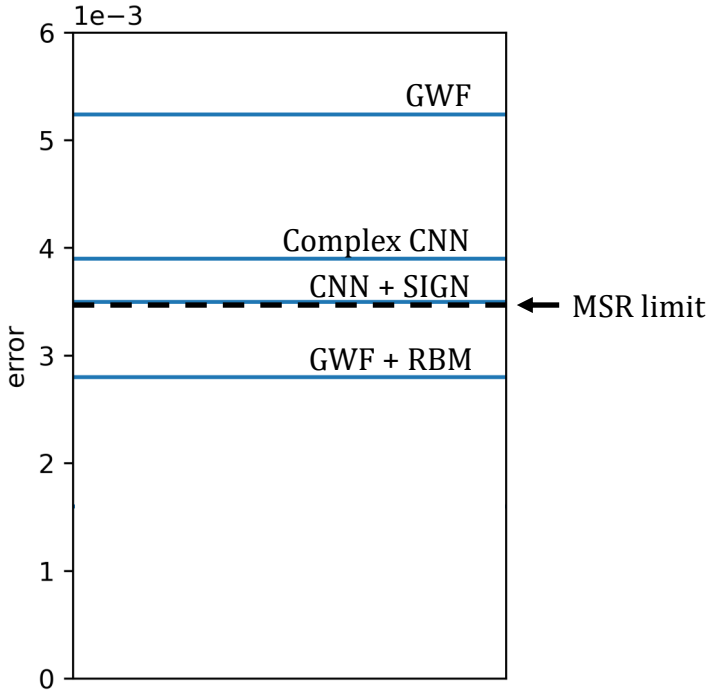


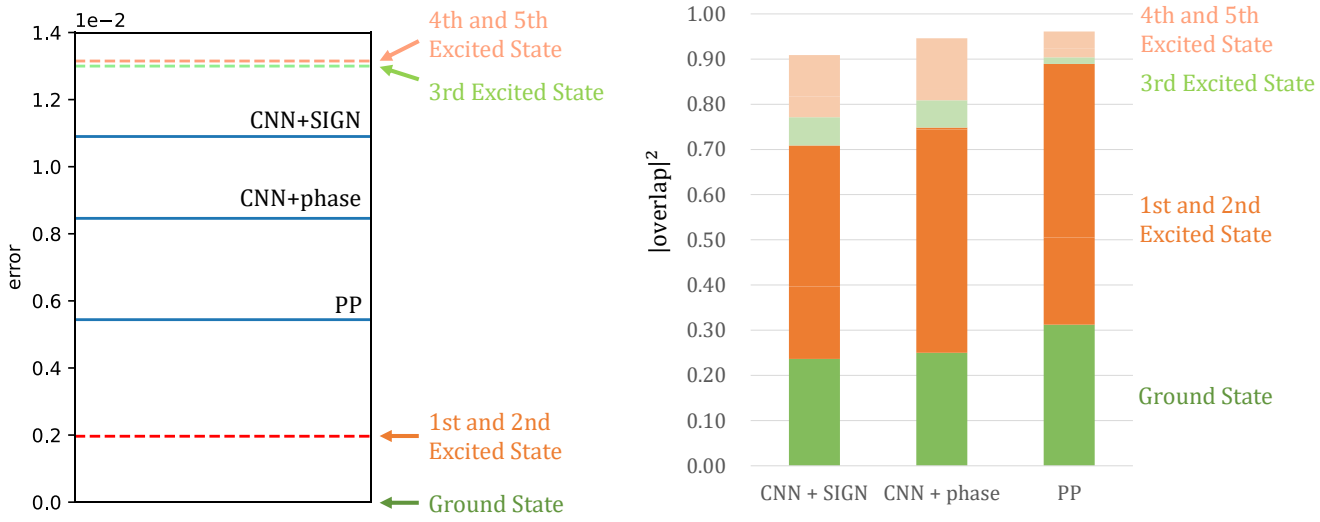
Figure 6.11: Comparison of variational energy error in $J_2 = 0.5$ square 6×6 lattice. The MSR limit is obtained by the neural network result with the sign structure fixed. The GWF, GWF+RBM and complex CNN results come from Ref.[22], Ref.[55] and Ref.[13], respectively.

cause the difficulty for the sign networks to capture the law behind the sign structure. Ref.[65] has shown that it is very hard for the neural networks to generalise the knowledge of the sign structure it obtains during the training to the configurations not trained before in the frustrated region. The irregular sign structure separated from the amplitude part becomes a big challenge for the neural network.

6.5 3D $2 \times 2 \times 2$ Pyrochlore Lattice

The J_1-J_2 model in 3D pyrochlore lattice is much more complicated than in 2D square lattice. Different from 2D square lattice where MSR can be a good approximation for the sign structures in the frustrated region, there is no available approximate sign structure for 3D pyrochlore lattice and the neural network has to learn a good sign structure by itself. The severe sign problem will be a big difficulty if traditional network structures like complex-valued RBM or CNN are directly applied to this strongly frustrated model.

We consider pyrochlore lattice with $2 \times 2 \times 2$ unit cells. As each unit cell contains 4 sites, the total number of sites is 32. As usual neural network methods do not work in this model, we compare ES with gradient-based phase network (CNN+phase) [61] as well as PP. In two neural network methods the translation symmetry is directly implemented in the CNN architecture. In PP we use symmetry projection to impose translation symmetry. The relative error of variational energy and the overlap with lowest eigenstates are shown in Fig.6.12. The performance of ES is not as good as other methods, but it is still a feasible method to gain reasonable results in such a complex model. The difference between the sign network trained by ES and the phase network trained by SR probably comes from some training details.



(a) Relative error of variational energy

(b) Overlap with lowest 6 eigenstates

Figure 6.12: Various methods in pyrochlore lattice. The eigenstates and their energy values are obtained by ED results provided by Nikita Astrakhantsev. The CNN+phase data is also from Nikita Astrakhantsev.

The overlap shows that in all three methods the ground state and the first two excited states contribute most to the variational wave function. The 3rd to 5th excited states also have a few contributions. However, the existence of some components in higher excited states cause a relatively higher variational energy compared with 1st and 2nd excited states.

An additional S_3 symmetry not used in Fig.6.12 can help to eliminate the contribution of the 1st, 2nd, 4th and 5th excited states. These states are represented by orange colors in Fig.6.12 to distinguish them from other states which will remain after the projection. However, for neural network methods based on ES or SR, the projection will greatly increase the variational energy. On the other hand, the PP variational energy will reduce to a relative error of 0.00316 after the projection. Its squared overlap with the ground state will also increase to 95%. This indicates that the PP variational wave function is closer to the ground state wave function.

Chapter 7

Evolution Strategy for Auxiliary Sign Structures

In the previous chapter, we have seen some achievements as well as limitations when a sign network trained by ES serves as an individual sign part. The limits mainly come from the form of the variational wave function. When the amplitude part and the sign part are totally separated as Eq.(4.10), the training process will become extremely difficult. Consequently, in this chapter we take the sign network as an auxiliary sign structure combined to a main part of the wave function with both the amplitude and the sign parts. The final wave function is

$$\psi(\sigma) = \psi_{\text{main}}(\sigma)\psi_{\text{sgn}}(\sigma). \tag{7.1}$$

7.1 ES for auxiliary sign structure of PP

PP is a typical variational ansatz with both the amplitude and the sign parts, so it is a natural choice to try the combination of PP and a sign network. This will serve as a good starting point of the whole chapter.

In Fig.7.1, we show the performance of a sign network trained by ES (SIGN for short) when combined with PP in 8×8 square lattice at $J_2 = 0.5$. The sign network can improve the energy obtained by PP through modifying its sign part, but this improvement is very small. The result gets better when we introduce an RBM as the correction for the amplitude part. The details of the setting are as follows: only parity symmetry is implemented in PP; the symmetric RBM is a single-layer CNN with 64 channels without bias; SIGN has 2 channels. A result we did not show in Fig.7.1 is that when more symmetries are imposed on PP the energy will get better significantly. In this case, however, the improvement of PP+SIGN will be even smaller.

The improvement when a sign part is imposed on PP is very small and even negligible. The reason is that PP can learn the sign structure very well, especially when PP has more symmetries. An auxiliary sign structure is not much helpful. An evidence is that in Ref.[24] the PP+RBM combination gives a very accurate result even though RBM cannot help to improve the sign structure. Another problem is that PP does not work well with an additional sign part. For instance, the performance of PP in the Heisenberg model will deteriorate significantly if MSR is imposed as an auxiliary sign structure. Consequently, PP is not a good choice if we want to test the performance of the auxiliary sign network. Some other variational ansatz without sufficient ability in learning the sign structure should be introduced to reveal the true ability of the auxiliary sign network.

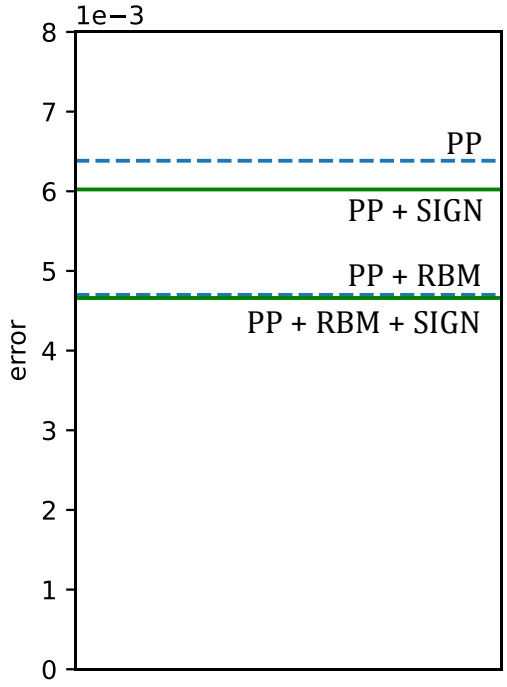


Figure 7.1: Energy improvement when SIGN is combined with PP in 8×8 square lattice at $J_2 = 0.5$. The ground state energy is approximated by the GWF result with 2 Lanczos steps in Ref.[22]. The results without SIGN are represented by blue dotted lines. The results with SIGN are represented by green solid lines.

7.2 Neural Network Output From $\ln \psi$ To ψ

The wave function to take the place of PP is exactly what we have discussed for the whole thesis - the neural network. The main difference between the new network architecture and the network discussed before is the output of the network. We use the network to express the wave function components ψ_σ directly instead of $\ln \psi_\sigma$ widely used in the practices of previous papers. The use of complex parameters in the networks expressing $\ln \psi_\sigma$ proves to be helpless in adapting to a correct sign structure as shown in Fig.6.7. The new choice will allow ψ_σ to change from positive to negative continuously without introducing a complex phase. Using a variational ansatz to express ψ_σ directly is also the choice of some other successful variational wave function forms including PP and MPS.

The aim of using the network to express $\ln \psi_\sigma$ in existing literature is to ensure the stability during the optimisation and the performance in learning the amplitude. However, in order to express a good sign structure, we sacrifice these advantages in exchange for the ability in learning sign structures.

We will use a CNN architecture as the main part of the wave function $\psi_{\text{main}}(\sigma)$. To enhance the expressive power of the network, we introduce a configuration product layer, as shown in Fig.7.2. This layer serves as the last layer of a CNN to generate the network output which is taken as ψ_σ. Different elements in the same channel are multiplied together to convert every channel to a number and then these numbers are summed to give a scalar output. This layer will not change the network output if the input of CNN is translated, so the translation symmetry of CNN is maintained.

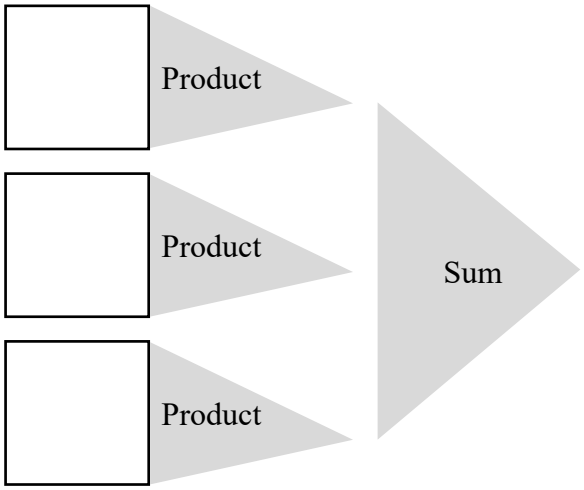


Figure 7.2: Configuration product layer

The configuration product layer is inspired by PP, which, as explaned in Eq.(3.74), can be viewed as a sum over the products of pair entanglement coefficients. In configuration product layer we view the different elements in the channels as the entanglement coefficients of the whole configuration σ, and apply the same manipulation as the pair product to these coefficients. Since the configuration product manipulation serves as the core structure of the network, we will call such a network the configuration product network (CPN).

The architecture of the whole CPN for the square lattice is shown in Fig.7.3. There are two convolutional layers with a kernel size as large as the whole lattice and one configuration product layer. The first convolutional layer has a ReLU activation function. The second one has no activation, and also has no bias term. For the pyrochlore lattice, the structure is roughly the same, but the channel number following the input layer is changed to 100 to compensate for the relatively smaller lattice size. The number of parameters for the 6×6 square lattice and the $2 \times 2 \times 2$ pyrochlore lattice is 59080 and 83300, respectively. This is a very large number compared with usual networks used as NQS, which usually has about 3000 to 10000 parameters. In spite of so many parameters, the choice of using ψ as the network output proves to be helpful in accelerating the training process. In our practice, the training of CPN is about 10 times faster than the previous networks that output $\ln \psi$.

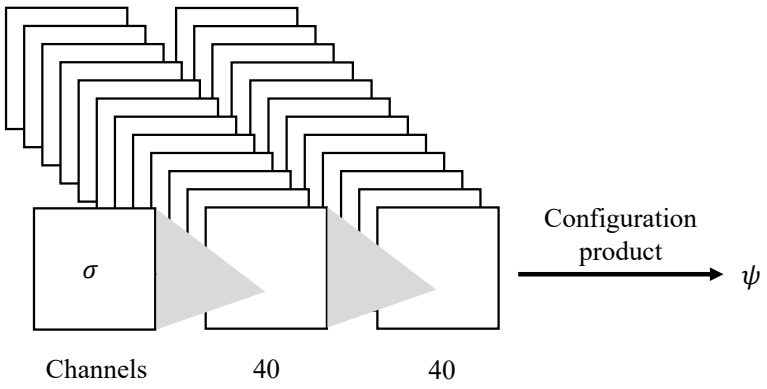


Figure 7.3: Architecture of the configuration product network. Not all channels are shown to simplify the diagram.

The sign network trained by ES will serve as a background sign structure for CPN, so our

new training strategy is to train the sign network individually with all amplitudes equal, and then fix the sign network and optimise CPN. This strategy will not cause a stop at the local minimum shown in Fig.6.4 because both the amplitude and the sign parts can be tuned in the configuration product network.

Finally, we also introduce RBM as an auxiliary amplitude part for CPN. After the training of CPN converges, we add RBM into the ansatz and train CPN and RBM together. This will give a slight improvement to the variational energy obtained by CPN.

7.3 2D Square Lattice

We will use the 2D square lattice to illustrate the main properties of the configuration product network (CPN) and its performance when combined with a sign network (SIGN). The result in the square 6×6 Heisenberg model ($J_2 = 0$) is shown in Fig.7.4. Here we use a sign network trained by ES as an auxiliary sign structure. Alternatively, one can also impose MSR directly. The rotation symmetry is used in all results. The complex CNN and CNN+SIGN result is the same as what we showed in Fig.6.10. In addition to the CPN architecture introduced above, we also use a slightly different CPN architecture with an activation function $\text{ShiftedReLU}(x) = \text{ReLU}(x) + 1$ before the configuration product layer to force the network to express a positive amplitude. This new architecture will be helpful for investigating the ability of CPN in learning amplitudes. From this figure we can find two significant characteristics of the configuration product network (CPN). First, the performance of CPN in learning amplitudes is not as good as those networks expressing $\ln\psi$. This is not a big problem because GWF and PP also have relatively worse performance in learning amplitudes compared with CNN, but they can learn the frustrated sign structures as we want. Second, although CPN is capable of learning some sign structures and lower the error of the variational energy to about 10^{-2}, a good auxiliary sign structure is still very important to make its result competitive.

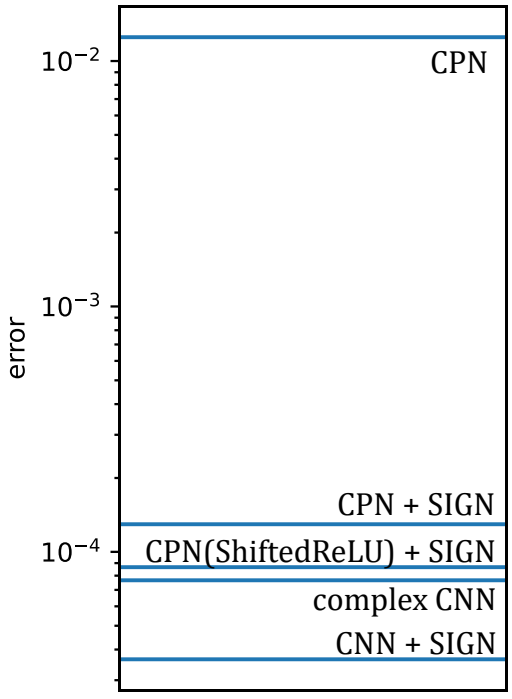


Figure 7.4: Error of variational energy in square 6×6 Heisenberg model in log scale.

The reason behind the limits of CPN in learning some trivial sign structures like MSR is that the usual neural network architecture is only good at expressing continuous and smooth functions. For many quantum systems we encounter, nevertheless, the wave function can abruptly change from positive to negative even only two spins are exchanged. This characteristic is very different from the usual applications of neural networks like computer vision (CV) where we seldom see an obvious difference when two pictures are just different by a few pixels. The network architecture should be very carefully designed to express the violently oscillating sign structures like MSR. The design of the sign network we use in this thesis is a good example of such a network able to express these sign structures.

The significance of the configuration product network becomes obvious in the frustrated situation at $J_2 = 0.5$ as shown in Fig.7.5. The MSR limit and the GWF+RBM result is the same as the data in Fig.6.11. The variational energy of CPN is much better than the limit energy we can obtain with MSR, which means that the network learns the frustrated sign structure successfully. This is also the first pure neural network method capable of learning frustrated sign structures in square J_1-J_2 model in the studies of NQS. In addition, an RBM can be used as an auxiliary amplitude part to further reduce the energy. A result we did not show in Fig.7.5 is that the error of CPN will increase dramatically to about 0.1 if an auxiliary sign structure is not used. This result indicates the necessity of the additional sign network.

Here we need to explain why CPN, inspired by PP, gets worse variational energy compared with PP. Although PP only considers pair products, it has much more terms to sum up in the wave function compared with CPN because all possible pair product combinations are considered. Its perfect structure as a pfaffian makes it possible to go through all the calculations efficiently. This is usually impossible for a neural network, so the terms used in CPN is much less. This is the main reason that reduces the expressive power of CPN.

Although the performance of CPN is not as good as PP, it has shown that the neural network methods can capture frustrated sign structures without the help of GWF or PP. With this result, we can state with no doubt that the artificial neural networks can be used in solving frustrated sign structures. This result exhibits a possible direction for the future development of NQS.

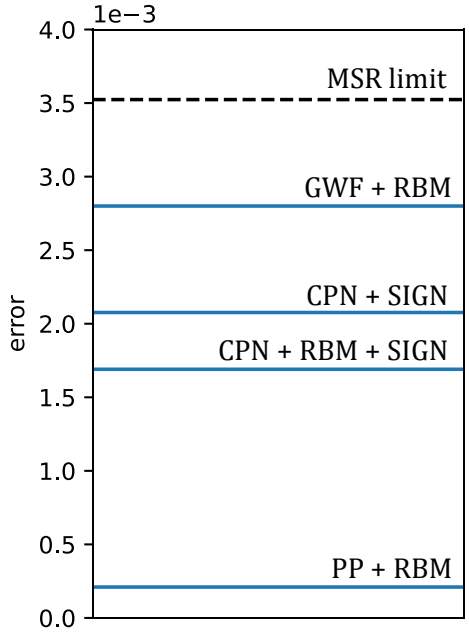


Figure 7.5: Error of variational energy in square 6×6 J_1-J_2 model at $J_2 = 0.5$.

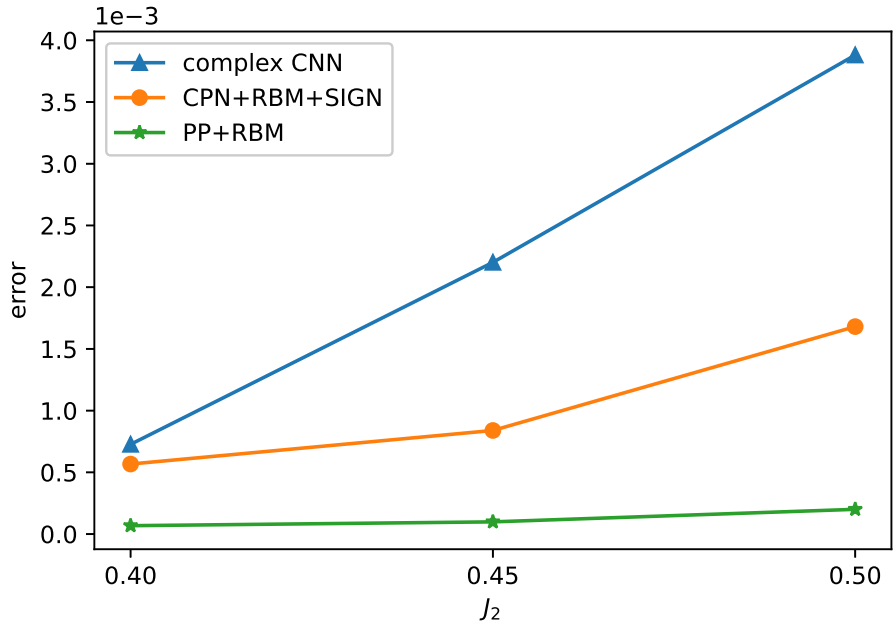


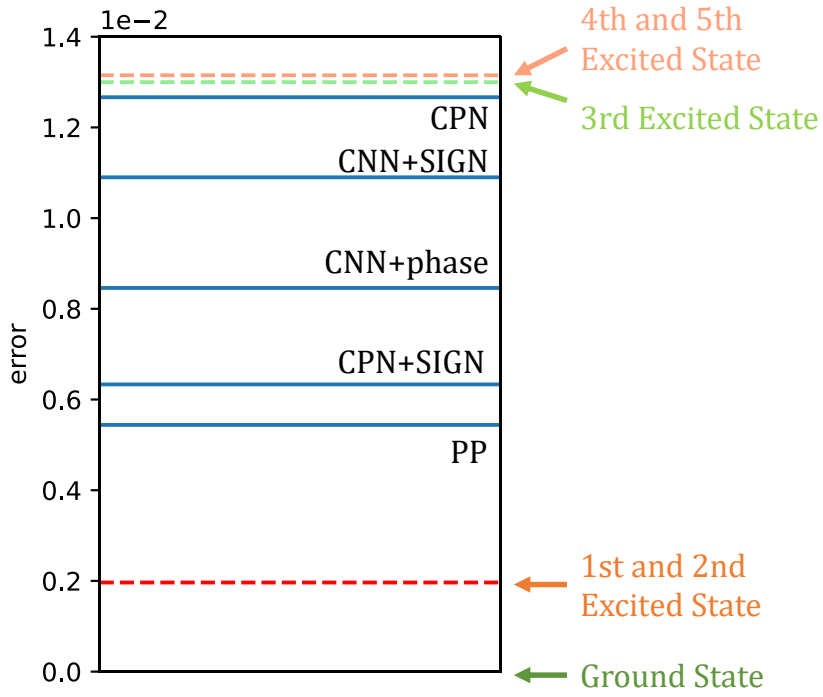
Figure 7.6: Relative error of variational energy for different J_2 values in square 6×6 J_1-J_2 model. The exact ground energy data is from Ref.[21]. For $J_2 = 0.45$ the ground state energy is not available, so we use the DMRG result from Ref.[23]. PP+RBM result is from Ref.[24].

In Fig.7.6 we show the relative error of the variational energy across different J_2 values in the frustrated region of the square 6×6 J_1-J_2 model with J_1 fixed to 1. The result of 3 different methods show that the variational energy of our method is significantly better than the complex-valued CNN. This is actually the best neural network result in square J_1-J_2 model until now. However, it is still not as good as PP. The distance between NQS and traditional methods in frustrated J_1-J_2 models still exists.

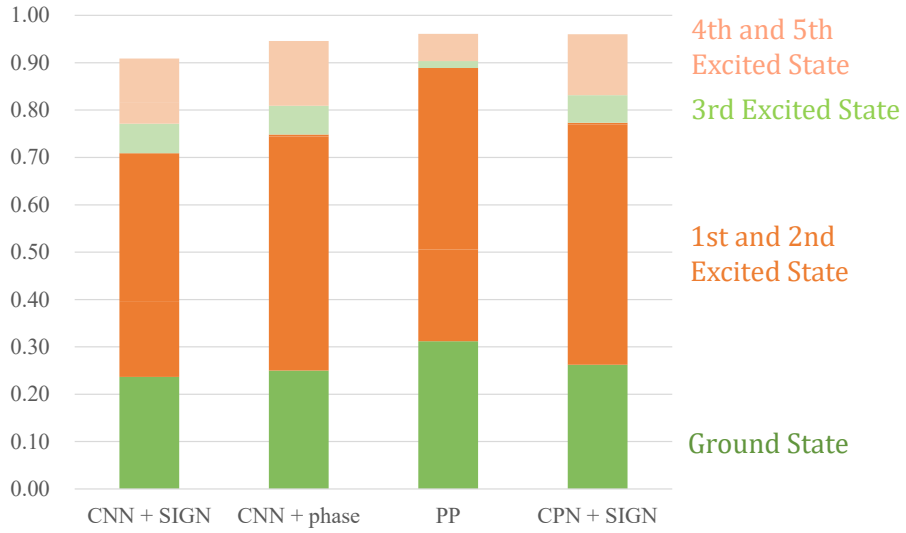
7.4 3D Pyrochlore Lattice

The application of ES is very trivial in the square lattice, because the sign network trained by ES only serves as the MSR which can be easily imposed directly. In the 3D pyrochlore lattice, nevertheless, there does not exist such a good sign structure like MSR for us to begin with. An auxiliary sign network will be necessary in this case.

In Fig.7.7, we add the new result from the configuration product network to Fig.6.12 where we show the variational energy with the sign network serving as an independent sign structure. This time we do not use an auxiliary RBM because its effect is negligible in our numerical tests. The combination of CPN+SIGN gets a result much better than CPN itself. This means that the auxiliary sign part trained by ES serves as a very important part of the variational wave function. The non-trivial contributions from ES proves that ES is very helpful in giving some primary sign structures which can dramatically improve the performance of neural networks in learning quantum sign structures. The improvement in this complicated model shows the potential of our method in solving cutting-edge problems of the quantum many-body systems.



(a) Relative error of variational energy



(b) Overlap with lowest 6 eigenstates

Figure 7.7: Various methods in $2 \times 2 \times 2$ pyrochlore lattice at $J_2 = 0$. All results except for CPN and CPN+SIGN are the same as Fig.6.12.

On the other hand, the overlap of CPN+SIGN with the ground state does not improve very much. This indicates that the improvement of the energy compared with CNN+SIGN mainly comes from the reduction of some higher excited components. After symmetry projections, our method still suffers from a significant increase of the variational energy. Some further developments of the neural network methods are still required to obtain a good result after projections in the pyrochlore lattice.

In Fig.7.8, we show the performance of different methods for different J_1 and J_2 values. Similar to Fig.2.11, J_1 and J_2 are tuned by a parameter θ with $J_1 = \cos\theta$ and $J_2 = \sin\theta$.

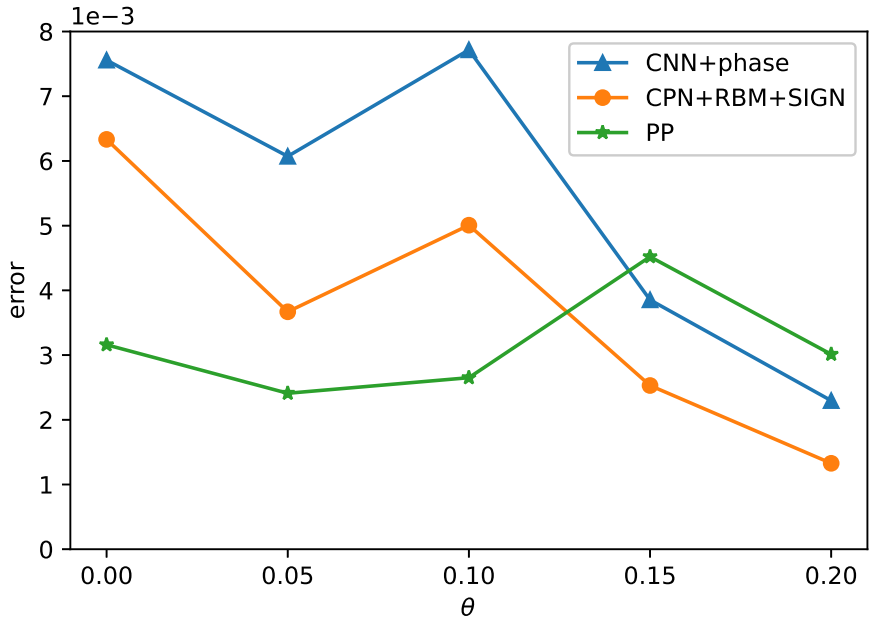


Figure 7.8: Relative energy error of various methods in $2 \times 2 \times 2$ pyrochlore lattice for different θ. The ground state and CNN+phase data is provided by Nikita Astrakhantsev.

Fig.7.8 indicates that our method ourperforms CNN+phase across different θ values. A CPN network coupling the amplitude and the sign is beneficial for approaching the ground state compared with the entire separation of the two parts. The combination of NQS with an auxiliary sign part, a configuration product part and an auxiliary amplitude part may be a good choice for the future studies of neural network methods.

On the other hand, PP still works better in the highly frustrated region around $\theta = 0$, but the neural network methods obtain better variational energy when θ gets to 0.15 and the frustration is alleviated. This coincides with the conclusion of Ref.[12] and Ref.[13] presented in Fig.4.4 and Fig.4.7 that NQS can have better performance compared with traditional numerical methods when the frustration is weak. In the region with a strong frustration, however, the traditional methods are still better. Nevertheless, the current result of our CPN+RBM+SIGN method is satisfactory considering the use of CPN is still in an early stage. Further adjustments of the CPN architecture according to the properties of quantum spin liquids in the future may improve its performance and help to obtain more accurate results in frustrated models.

Chapter 8

Conclusion and Outlook

In this thesis, we study a new optimisation method named evolution strategy in the field of NQS and its application in solving quantum sign structures. We will draw our conclusions in the following aspects.

The first aspect is the choice of the optimisation method in VMC. Different from the usual SR method based on the gradient information of the variational ansatz, ES provides a possible way to optimise non-differentiable wave functions. We show that the methods not relying on the gradients can also be used to training NQS. This characteristic of ES allows us to eliminate the complex degrees of freedom when we introduce a sign network in this thesis. In future researches, ES will also be helpful if some other non-differentiable ansatz are proposed.

The second aspect is the form of the neural network to take as a variational wave function. We have shown that the three different kinds of NQS distinguished by their outputs have totally different performance in learning quantumm sign structures. The complex-valued network taking $\ln \psi$ as its output does not work well in learning unknown sign structures. The real-valued network with a combination of a $\ln \psi$ amplitude part and a sign (or phase) part can learn some primary sign structures like MSR and some frustrated sign structures in very small systems, but in general they fail in larger frustrated systems. The last type is the network that outputs ψ directly and possibly contains an auxiliary sign part. This is the architecture with best performance in learning frustrated sign structures among three network architectures we discussed. Consequently, this thesis gives a hopeful direction for the future development of NQS. A good network architecture may contain an auxiliary sign part trained by ES, a network expressing the main part of the amplitude like CPN and an auxiliary $\ln \psi$ amplitude network like RBM. The sign network trained by ES will be necessary in this architecture to reduce the difficulty of expressing the frustrated sign structures for the main part of the wave function.

Finally, we want to discuss the comparison between NQS and other methods. Many other variational methods like PP have better performance compared to neural networks in frustrated systems, but the development of neural network methods is still necessary. The success of neural networks in learning the non-frustrated systems has shown its potential in the quantum many-body problems. Its complicated architecture allows us to enhance the expressive power and obtain better results through adding more layers or channels in systems without the sign problem. Similar result may also be achieved in the frustrated systems in the future so that we can obtain very accurate properties of large frustrated systems with the help of neural networks. This thesis may indicate a possible way to make NQS competitive in highly frustrated systems.

References

[1]D. Ceperley and B. Alder, "Quantum monte carlo", Science **231**, 555–560 (1986).

[2]M. Troyer and U.-J. Wiese, "Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations", Phys. Rev. Lett. **94**, 170201 (2005).

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

[4]W. L. McMillan, "Ground state of liquid he4", Phys. Rev. **138**, A442–A451 (1965).

[5]F. Becca and S. Sorella, "Variational monte carlo", in *Quantum monte carlo approaches for correlated systems* (Cambridge University Press, 2017), pp. 103–130.

[6]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", nature **521**, 436–444 (2015).

[7]C. B. Csáji, "Approximation with artificial neural networks", MA thesis (Eötvös Loránd University, 2001).

[8]D. Wu, L. Wang, and P. Zhang, "Solving statistical mechanics using variational autoregressive networks", Phys. Rev. Lett. **122**, 080602 (2019).

[9]J. Carrasquilla and R. G. Melko, "Machine learning phases of matter", Nature Physics **13**, 431–434 (2017).

[10]E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, "Learning phase transitions by confusion", Nature Physics **13**, 435–439 (2017).

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

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

[13]K. Choo, T. Neupert, and G. Carleo, "Two-dimensional frustrated $J1-J2$ model studied with neural network quantum states", Phys. Rev. B **100**, 125124 (2019).

[14]I. Rechenberg, "Evolution strategy: nature's way of optimization", in Optimization: methods and applications, possibilities and limitations, edited by H. W. Bergmann (1989), pp. 106–126.

[15]G. G. Batrouni and R. T. Scalettar, "World-line quantum monte carlo algorithm for a one-dimensional bose model", Phys. Rev. B **46**, 9051–9062 (1992).

[16]A. W. Sandvik, "Stochastic series expansion method with operator-loop update", Phys. Rev. B **59**, R14157–R14160 (1999).

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

[18]P. W. Anderson, "The resonating valence bond state in la2cuo4 and superconductivity", science **235**, 1196–1198 (1987).

[19]N. Read and S. Sachdev, "Spin-peierls, valence-bond solid, and néel ground states of low-dimensional quantum antiferromagnets", Physical Review B **42**, 4568 (1990).

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

[21]H.J. Schulz, T.A.L. Ziman, and D. Poilblanc, "Magnetic order and disorder in the frustrated quantum heisenberg antiferromagnet in two dimensions", J. Phys. I France **6**, 675–703 (1996).

[22]W.-J. Hu, F. Becca, A. Parola, and S. Sorella, "Direct evidence for a gapless $Z2$ spin liquid by frustrating néel antiferromagnetism", Phys. Rev. B **88**, 060402 (2013).

[23]S.-S. Gong, W. Zhu, D. N. Sheng, O. I. Motrunich, and M. P. A. Fisher, "Plaquette ordered phase and quantum phase diagram in the spin-$\frac{1}{2}$ $J1{-}J2$ square heisenberg model", Phys. Rev. Lett. **113**, 027201 (2014).

[24]Y. Nomura and M. Imada, *Dirac-type nodal spin liquid revealed by machine learning*, 2020.

[25]Y. Iqbal, T. Müller, P. Ghosh, M. J. P. Gingras, H. O. Jeschke, S. Rachel, J. Reuther, and R. Thomale, "Quantum and classical phases of the pyrochlore heisenberg model with competing interactions", Phys. Rev. X **9**, 011005 (2019).

[26]R. Moessner and J. T. Chalker, "Properties of a classical spin liquid: the heisenberg pyrochlore antiferromagnet", Phys. Rev. Lett. **80**, 2929–2932 (1998).

[27]R. Moessner and J. T. Chalker, "Low-temperature properties of classical geometrically frustrated antiferromagnets", Phys. Rev. B **58**, 12049–12062 (1998).

[28]M. Hermele, M. P. A. Fisher, and L. Balents, "Pyrochlore photons: the $u(1)$ spin liquid in a $s = \frac{1}{2}$ three-dimensional frustrated magnet", Phys. Rev. B **69**, 064404 (2004).

[29]B. Canals and C. Lacroix, "Pyrochlore antiferromagnet: a three-dimensional quantum spin liquid", Phys. Rev. Lett. **80**, 2933–2936 (1998).

[30]J. H. Kim and J. H. Han, "Chiral spin states in the pyrochlore heisenberg magnet: fermionic mean-field theory and variational monte carlo calculations", Phys. Rev. B **78**, 180410 (2008).

[31]F. J. Burnell, S. Chakravarty, and S. L. Sondhi, "Monopole flux state on the pyrochlore lattice", Phys. Rev. B **79**, 144432 (2009).

[32]Y. Huang, K. Chen, Y. Deng, N. Prokof'ev, and B. Svistunov, "Spin-ice state of the quantum heisenberg antiferromagnet on the pyrochlore lattice", Phys. Rev. Lett. **116**, 177203 (2016).

[33]P. A. Gagniuc, *Markov chains: from theory to implementation and experimentation* (John Wiley & Sons, 2017).

[34]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", The Journal of Chemical Physics **21**, 1087–1092 (1953).

[35]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).

[36]K. Kreutz-Delgado, *The complex gradient operator and the cr-calculus*, 2009.

[37]S.-C. T. Choi and M. A. Saunders, "Algorithm 937: minres-qlp for symmetric and hermitian linear equations and least-squares problems", ACM Trans. Math. Softw. **40** (2014).

[38]E. P. Münger and M. A. Novotny, "Reweighting in monte carlo and monte carlo renormalization-group studies", Phys. Rev. B **43**, 5773–5783 (1991).

[39]T. Nakamura, N. Hatano, and H. Nishimori, "Reweighting method for quantum monte carlo simulations with the negative-sign problem", Journal of the Physical Society of Japan **61**, 3494–3502 (1992).

[40]N. Shimizu, T. Mizusaki, and K. Kaneko, "Stochastic extension of the lanczos method for nuclear shell-model calculations with variational monte carlo method", Physics Letters B **723**, 251–254 (2013).

[41]G. Baskaran, Z. Zou, and P. Anderson, "The resonating valence bond state and high-tc superconductivity — a mean field theory", Solid State Communications **63**, 973–976 (1987).

[42]D. Tahara and M. Imada, "Variational monte carlo method combined with quantum-number projection and multi-variable optimization", Journal of the Physical Society of Japan **77**, 114701 (2008).

[43]M. Wimmer, "Algorithm 923: efficient numerical computation of the pfaffian for dense and banded skew-symmetric matrices", ACM Trans. Math. Softw. **38** (2012).

[44]X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks", in Proceedings of the fourteenth international conference on artificial intelligence and statistics (2011), pp. 315–323.

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

[46]F. Dangel, F. Kunstner, and P. Hennig, "Backpack: packing more into backprop", in International conference on learning representations (2020).

[47]P. Smolensky, *Information processing in dynamical systems: foundations of harmony theory*, tech. rep. (Colorado Univ at Boulder Dept of Computer Science, 1986).

[48]R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, "Restricted boltzmann machines in quantum physics", Nature Physics **15**, 887–892 (2019).

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

[50]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).

[51]M. Lubasch, J. I. Cirac, and M.-C. Bañuls, "Algorithms for finite projected entangled pair states", Phys. Rev. B **90**, 064425 (2014).

[52]J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, "Equivalence of restricted boltzmann machines and tensor network states", Phys. Rev. B **97**, 085104 (2018).

[53]X. Gao and L.-M. Duan, "Efficient representation of quantum many-body states with deep neural networks", Nature Communications **8**, 662 (2017).

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

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

[56]V. Dumoulin and F. Visin, *A guide to convolution arithmetic for deep learning*, 2016.

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

[58]N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks", in , Vol. 70, Proceedings of Machine Learning Research (2017), pp. 1771–1779.

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

[60]Z. Cai and J. Liu, "Approximating quantum many-body wave functions using artificial neural networks", Phys. Rev. B **97**, 035116 (2018).

[61]A. Szabó and C. Castelnovo, *Neural network wave functions and the sign problem*, 2020.

[62]N. Hansen, *The cma evolution strategy: a tutorial*, 2016.

[63]N. Hansen, Y. Akimoto, and P. Baudis, *CMA-ES/pycma on Github*, 2019.

[64]G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, and et al., "Netket: a machine learning toolkit for many-body quantum systems", SoftwareX **10**, 100311 (2019).

[65]T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, "Generalization properties of neural network approximations to frustrated magnet ground states", Nature Communications **11**, 1593 (2020).

Appendix A

Variational Energy in Square J_1-J_2 Models

Variational energy in square 6×6 J_1-J_2 model

J_2	0.0	0.4	0.45	0.5	0.55
ED	-0.678872	-0.529745		-0.503810	-0.495178
QMC	-0.678873(4)				
GWF		-0.52715(1)	-0.51364(1)	-0.50117(1)	-0.48992(1)
GWF (p=2)		-0.52957(1)	-0.51558(1)	-0.50357(1)	-0.49399(1)
DMRG		-0.529744	-0.515655	-0.503805	-0.495167
complex CNN	-0.67882(1)	-0.52936(1)	-0.51452(1)	-0.50185(1)	-0.49067(2)
GWF + RBM	-0.67874			-0.5024	
PP + RBM		-0.529709(5)	-0.515604(4)	-0.503704(5)	-0.494800(8)
CNN + SIGN	-0.67884			-0.50203	
CPN + RBM + SIGN	-0.678784(5)	-0.5294	-0.515222(7)	-0.50298(1)	

Variational energy in square 8×8 J_1-J_2 model

J_2	0.0	0.4	0.45	0.5	0.55
QMC	-0.673487(4)				
GWF		-0.52302(1)	-0.50930(1)	-0.49656(1)	-0.48487(1)
GWF (p=2)		-0.52539(1)	-0.51125(1)	-0.49886(1)	-0.48841(2)
DMRG		-0.525196	-0.510740	-0.498175	-0.488160
PP + RBM		-0.525492(4)	-0.511117(4)	-0.498460(6)	-0.48781(1)

Variational energy in square 10×10 J_1-J_2 model

J_2	0.0	0.4	0.45	0.5	0.55
QMC	−0.671549(4)				
GWF		−0.52188(1)	−0.50811(1)	−0.49521(1)	−0.48335(1)
GWF (p=2)		−0.5240(1)	−0.51001(1)	−0.49755(1)	−0.48693(3)
DMRG		−0.522391	−0.507976	−0.495530	−0.485434
complex CNN	−0.67135(1)	−0.52371(1)	−0.50905(1)	−0.49516(1)	−0.48277(1)
GWF + RBM	−0.67111(2)			−0.49575(3)	
CNN + phase	−0.671275(5)			−0.494757(12)	

The sources of data are as follows: ED [21], QMC [49], GWF and GWF(p=2) [22], DMRG [23], complex CNN [13], GWF+RBM [55], PP+RBM [24] and CNN+phase [61]. CNN+SIGN and CPN+RBM+SIGN are from the numerical results of this thesis.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

NEURAL NETWORK EVOLUTION STRATEGY FOR SOLVING QUANTUM SIGN STRUCTURES

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
CHEN	AO

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 15-09-2020	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.