Bachelor's Programme in Science and Technology

# Solving the Quantum Heisenberg model with neural-network quantum states

Subham Das

Aalto University

BACHELOR'S THESIS

Aalto University
**BACHELOR'S THESIS** 2024

# Solving the quantum Heisenberg model with neural-network quantum states

**Subham Das**

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Technology.
Otaniemi, 27 Nov 2024

| | |
|---|---|
| Supervisor: | Prateek Chawla |
| Advisor: | Jose Lado |

**Aalto University**
**School of Science**
**Bachelor's Programme in Science and Technology**

**Aalto University**

**Abstract**

**Aalto University, P.O. Box 11000, FI-00076 Aalto** www.aalto.fi

**Author**
Subham Das

**Title**
Solving the Quantum Heisenberg model with neural-network quantum states

**School** School of Science

**Degree programme** Bachelor’s Programme in Science and Technology

**Major** Quantum Technology **Code** SCI3103

**Supervisor** Jose Lado

**Advisor** Prateek Chawla

**Level** Bachelor’s thesis **Date** 27 Nov 2024 **Pages** 38 **Language** English

## Abstract

Solving quantum many-body systems is a fundamental challenge in physics. Traditional computational methods often struggle to solve them due to the exponential growth of the system’s Hilbert space. In recent times, Neural-Network Quantum States (NNQS) has emerged as a promising tool to approximate quantum wave functions based on machine learning methodologies. Using machine learning enables efficient computation of high-dimensional quantum systems which are otherwise unfeasible to compute. This work investigates two foundational models in quantum mechanics: One dimensional transverse field Ising and Heisenberg model, and predicts the expected energies and other properties with the help of machine learning.

To implement the NNQS, predefined neural network functions from the NetKet library is used. The methodology uses Monte Carlo sampling techniques along with gradient descent optimizer to find the approximate solution. Using these methods we predict the ground state energy, magnetization and magnetic susceptibility in transverse and longitudinal fields for each of the Ising system and the Heisenberg system.

Later, the Barren Plateau is also investigated. In a Barren plateau, gradient based optimization methods struggle to find the optimal solution. In this work, this was a significant problem to reach the approximate solution.

Lastly, the results are attached after which we summarize our findings and compare the methodologies and give a final reflection on the work.

**Keywords** Quantum many-body problems, Neural networks, Neural-Network Quantum States, Ising model, Heisenberg model, Machine learning, Barren plateau

**urn** https://aaltodoc.aalto.fi

# Contents

# List of Figures

# Code Listings

# 1. Introduction

Quantum many body problems are among the central challenges in modern physics, and are particularly important for understanding the behaviors and properties of materials and systems governed by quantum mechanics. These problems become increasingly difficult to solve as the dimensionality of the system grows exponentially with its size, leading to a vast Hilbert space. Traditional computational methods such as exact diagonalization or density matrix renormalization group methods are effective for finding solutions of smaller system size. However, computational feasibility diminishes rapidly when the system size is increased.

Machine learning has recently emerged as a tool for tackling many scientific problems, offering new perspectives(patterns) and better computational advantages to traditional methods. In the context of quantum many-body physics, machine learning has introduced the concept of variational wavefunctions (1) also known as neural-network quantum states(NNQS). NNQS uses artificial neural networks to approximate wavefunctions of quantum systems. NNQS can be used to simulate and solve such high-dimensional quantum systems easier and effective. Unlike traditional methods, NNQS are not limited to system size and can be used to solve ground states and other properties of a system.

In this thesis, we explore the application of NNQS to solve for two models - the Ising model and Heisenberg model, two very important models in quantum mechanics and statistical physics. Both of these models are useful for understanding phase transitions, ferromagnetic systems, quantum magnetism and spin interactions. Our primary objective here is to demonstrate the capability of neural network based methods for approximating the ground states and computing observables of the two systems using the NetKet library. The NetKet library integrates machine learning techniques with Monte Carlo simulations to implement NNQS.

We also examine some of the challenges encountered while employing this method, particularly the "Barren plateau" problem and ways to mitigate it. Barren Plateau is a known optimization problem which hinders the training of the neural network in high-dimensional parameter spaces. This Thesis demonstrates the effectiveness of using NNQS to evaluate quantum systems and presents the challenges associated with this approach.

# 2. Background

This section introduces the foundational topics and methods relevant for implementing the NNQS to solve for quantum systems. It begins with the Ising and Heisenberg models, which serve as a representation of spin systems in quantum mechanics. Following this, it explores computational strategies like the Lanczos method and their limitations. Finally, the discussion transitions to the emerging role of neural networks, particularly Neural Network Quantum States (NNQS), in approximating quantum solutions. These concepts provide a framework for understanding the challenges and innovations addressed in this thesis.

## 2.1 Ising Model

The Ising model is a statistical model that can be used to represent ferromagnetic and thermodynamic systems. It models spin-$\frac{1}{2}$ particles such as electrons, which possess a magnetic moment due to their quantum mechanical spin.

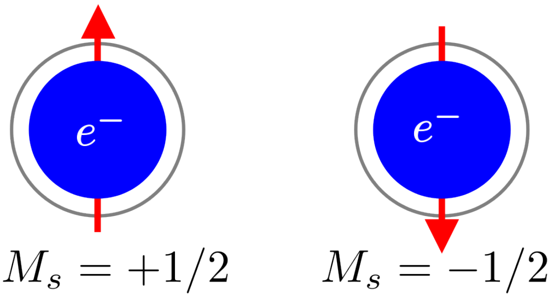

**Figure 2.1.** Spin-$\frac{1}{2}$ particles

Figure 2.1 describes such spin particles. The configurations of spins on a lattice determines the net magnetic moment, with aligned spins resulting in a overall ferromagnetic property and non-aligned spins resulting in anti-ferromagnetic property. In a ferromagnetic material, the interaction term contributes a negative term in the hamiltonian of the system when the spins are aligned, thus minimizing the energy of the system. Similarly in an anti-ferromagnetic system, the interaction term favors anti-alignment of the spins.

For example in case of electrons, (i) electrons repel each other and (ii) the Pauli exclusion rule which states that two electrons cannot occupy the same quantum state simultaneously. Thus when spins align, the spatial wavefunctions of the electrons are antisymmetric, reducing overlap in the electron cloud. This leads to less Coulomb repulsion between the electrons leading to a ferromagnetic state. Similarly in the case of anti-ferromagnetic material, the spatial wavefunctions of the anti-parallel spins can have more overlap of electron cloud because (ii) does not hold here leading to more Coulomb repulsion, however the interaction term between neighboring spins dominates and thus the system favors anti-ferromagnetism.

Let the spin be quantified as $\sigma_i$ where $\sigma_i = \pm 1$ (+1 for spin up and -1 for spin down) and $i$ represents the position of the atom in the lattice. If a pair of adjacent atoms have aligned(same) spin they are in a lower energy state than if they were in non-aligned(opposite) spin state. In order to remain stable, such a lattice tends to an aligned system where most spins at the positions are aligned. This results in a net macroscopic magnetic dipole of the lattice system since the system tends to aligned spin pairs instead of non-aligned spin pairs. Since increasing the temperature of the system directly correlates to increasing the kinetic energy of the atoms in the system which correlates to higher probability of the spin aligned pairs to become non aligned, $T_C$ is this critical temperature where the majority aligned state of the system breaks and the macroscopic magnetic dipole moment is lost and the system becomes disorganized. This transition of the Ising model from organized to disorganized is a phase transition (2).

In the special case of one dimensional lattice, Ising calculated and concluded that no phase transition occurred to ferromagnetic ordered state at any temperature (3). For our purposes we are only interested in one dimensional versions of the Ising model.

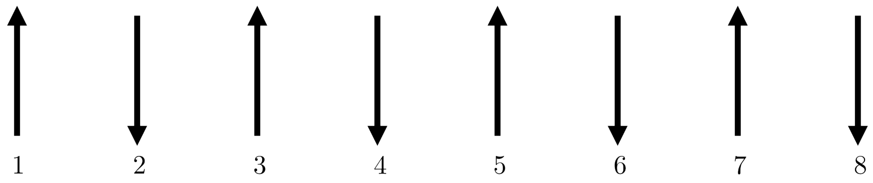

**Figure 2.2.** Alternating Ising lattice arrangement

### 2.1.1 Transverse Field Ising model

One dimensional lattice can be represented as a chain of spins with atoms next to each other in a line and the spin is $\sigma_i^Z = \pm 1$, $i$ is the position in lattice.

The Hamiltonian of such a system is given by:

$$H = -\sum_{\langle i,j\rangle} J_{ij}\sigma_i^z \sigma_Z j - \mu \sum_j h_j \sigma_j^X, \tag{2.1}$$

let $X$ be the set of all points in the lattice and $\alpha_k \in \{-1, +1\}$, $k$ is the site number in the lattice. For two sites $i, j \in X$ there exists a interaction $J_{ij}$. Although usually for the interaction term, only neighboring spins are considered since the interaction between non-neighboring spins diminish significantly with the gap between them. $\mu$ is the magnetic moment and if the site is under the influence of an external magnetic field $h_j$, the second term in the equation comes to existence. The negative interaction term $J$ reflects that the system is anti-ferromagnetic and the system tends to reduce the energy by anti-aligning the neighboring spins.

A transverse field Ising model starts demonstrating quantum mechanical properties in the presence of transverse field $h_j \neq 0$. The magnetic field $h_j$ starts affecting the spin directions of the particles in the system. This leads to a spin particle to enter a state of superposition of both $+1$ and $-1$ spin direction. In the classical version of Ising model, the external field is weak $h_j << J_{ij}$ compared to the neighboring interaction effects and the spin direction of the particles are not influenced much by $h_j$. When $h_j >> J_{ij}$, the spins tend to align according to the influence of the external field and thus shows quantum mechanical properties. This quantum spin property of the particle in the Ising model is an can be represented using a qubit. A qubit state for such spin-$\frac{1}{2}$ particle can be defined as:

$$|\psi\rangle = \alpha\,|{-1}\rangle + \beta\,|{+1}\rangle \tag{2.2}$$

Here $|\psi\rangle$, the single qubit state is a linear combination of $|{-1}\rangle$ and $|{+1}\rangle$ which are the basis vectors for the spins in our model. These basis vectors spans the Hilbert space for the quantum state and can be written down as: $|{-1}\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|{+1}\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. $\alpha$ and $\beta$ are the probability amplitudes for each basis vector. They also obey $|\alpha|^2+|\beta|^2 = 1$ since the total probability will always be equal to 1. The spin can be thought as an operator which when measured provides the eigenvalues $-1$ or $+1$.

Considering an Ising model neglecting the interaction terms of all particles except the neighboring particles. The Hamiltonian of this system is given by:

$$\mathcal{H} = -V\sum_i \sigma_i^{(z)} \otimes \sigma_{i+1}^{(z)} - \Gamma\sum_i \sigma_i^{(x)} \tag{2.3}$$

The first term describes the interaction of two spin particles next to each other, the subscript $i$ refers to the lattice site position. The sum $\Sigma$ is done for all pairs of neighboring spins $i$ and $i{+}1$. The superscripts $x$ and $z$ on $\sigma$ represents the direction of the corresponding particle/ external field. This can be represented by Pauli matrices in $x, y$ and $z$ directions respectively:

$$\sigma^x = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix},$$

$$\sigma^y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix},$$

$$\sigma^z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}.$$

Using Pauli matrices, the spin along any direction $\hat{n} = (n_x, n_y, n_z)$ can be represented as

$$\sigma_{\hat{n}} = n_x\sigma_x + n_y\sigma_y + n_z\sigma_z.$$

$\hat{n}$ is any arbitrary spin direction. $V$ is a constant which has dimensions of energy and $\Gamma$ is coupling strength with the external magnetic field, a coefficient which describes the strength of the external magnetic field on the lattice sites.

## 2.2 Anti-Ferromagnetic Heisenberg Model

In the Heisenberg model, the hamiltonian for the ground state is:

$$H(\sigma) = -\sum_{\langle i,j\rangle} J_{ij}(\sigma_i\sigma_j) - \sum_j h_j\sigma_j^Z, \tag{2.4}$$

Here, $\sigma_i = \sigma_i^x\hat{x} + \sigma_i^y + \sigma_i^z$ since in the Heisenberg mode, the spin-$\frac{1}{2}$ particles can have any arbitrary direction in $x, y$ and $z$, compared to only $\pm z$ direction in Ising model. Where $H$ is the Hamiltonian of the system, $J_{ij}$ is the coupling constant between the $i^{\text{th}}$ and $j^{\text{th}}$ spin particles, with $J > 0$ for ferromagnetic interactions (spins tend to align) and $J < 0$ for antiferromagnetic interactions (spins tend to anti-align). The term $\sigma_i\sigma_j = (\sigma_i \otimes \sigma_j)$ represents the spin interaction operators for the spins at sites $i$ and $j$, respectively. The spin operators are represented by Pauli matrices for spin-$\frac{1}{2}$ particles. This is because the Pauli matrices $\sigma^x, \sigma^y$, and $\sigma^z$ satisfy the same algebraic properties as the spin operators and provide a convenient way to express spin interactions.

The transverse field Ising model can be thought of as a special case of the Heisenberg model where the interaction terms only interact in the $z$ direction, instead of $x, y$ and $z$ directions.

Again, excluding the interaction terms of spins which are non-neighbors 2.4 becomes:

$$\mathcal{H} = J\sum_{0}^{N-1} \vec{\sigma_i} \otimes \vec{\sigma_{i+1}} - h\sum_{0}^{N} \sigma_i^z \tag{2.5}$$

Where $J$ is the constant interaction term between neighboring spins. N is the number of spin particles in the chain. $h$ is the external transverse field constant.

## 2.3 Exact Diagonalization Lanczos Method

The Lanczos algorithm is an iterative method designed to approximate eigenvectors and corresponding eigenvalues of large hermitian matrices. This is particularly useful when these matrices are sparce and is valuable in solving problems where exact method is computationally unfeasible.

The Lanczos algorithm transforms a large Hermitian matrix $A$ into a tridiagonal matrix $T$ through a series of orthogonal transformations (4). A tridiagonal matrix is a special type of matrix where all the elements except the main diagonal and the diagonals above and below the main diagonals are $0$. For $n \times n$ square matrix $T$:

$$T = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ c_1 & a_2 & b_2 & \cdots & 0 \\ 0 & c_2 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & b_{n-1} \\ 0 & 0 & 0 & c_{n-1} & a_n \end{bmatrix}$$

Here, $a_i, b_i$ and $c_i$ represents the elements on main diagonal,upper diagonal and lower diagonal respectively. After the tridiagonal matrix $T$ is obtained, the eigenvalues of $T$ are used as approximations for the original matrix $A$ and the eigenvectors of $A$ can be reconstructed from eigenvectors of $T$.

Using the Lanczos algorithm, it becomes easier to find a subset of eigenvalues and eigenvectors from the initial large matrix from the tridiagonal matrix and it allows to get the ground states and low-lying states without actually diagonalizing the initial matrix. However, there are some challenges with this algorithm including: numerical instability due to lack of orthogonality among the generated vectors and false eigenvalues (5).

## 2.4 Neural Networks

An artificial neural network is a computational model inspired by the neural structures found in brains. This artificial neural network(ANN) is capable of functioning similar to a biological neural network. In biological systems, neurons process and transmit information through complex networks that enable sophisticated behaviors and learning. ANNs aim to replicate some of these functions in a simplified, computational form, allowing machines to perform tasks such as pattern recognition, classification, and decision-making among others. An ANN consists of interconnected units called nodes or artificial neurons, which are organized into layers. Typically, these layers in a network includes an input layer, one or more hidden layers, and an output layer. The inter-nodal connections are represented by edges which models the functioning of synapses in a biological system and are represented by weights. Each of the nodes receives a set of signals from each of the sending nodes from the previous layer, applies a set of weights (adjustable parameter) to these inputs which is then processed by a function depending on variable conditions and passes on the output signal to the next set of nodes. Usually the network pattern and the activation function used depends on the type of problem the neural network the problem is catered for. This activation function introduces non-linearity into the model, allowing the network to approximate complex patterns that are not linearly separable. Without an activation function, the neural network would essentially perform linear transformations and this activation function on each node is essential for the whole neural network to model complex patterns.

Allowing a large number of nodes to come together and interconnect to form a connecting pattern with each other results in the formation of a neural network. The network depicted in the figure is just one of the many possible variations of networks that can be used.

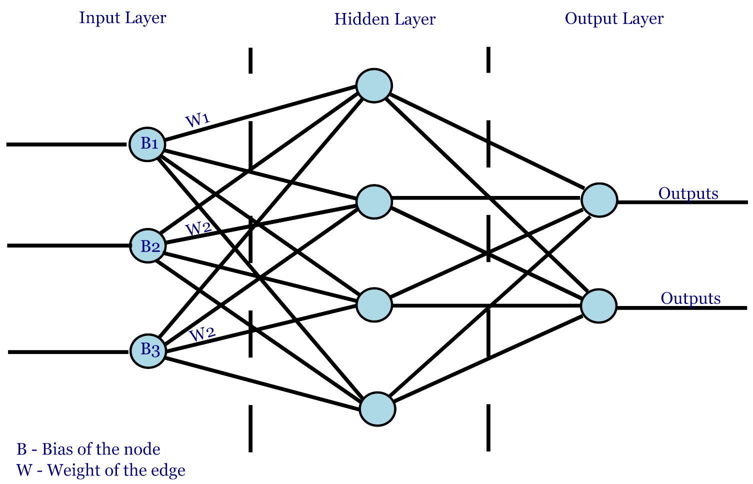

**Figure 2.3.** Artificial neural network architecture

Each connection has an associated weight that determines the strength and direction of the influence one node has on an another node. Learning typically occurs through an iterative process, where the network tries to minimize the error between the actual output and predicted outputs by iteratively adjusting the weights and biases of the network based on the gradient of the loss function (6). In estimating for Ising and Heisenberg model, this process is done by gradient descent algorithms which iteratively tries to find a lower ground state energy for the system.

By modifying the structure, depth, and activation functions of a neural network, researchers can tailor the ANN to a broad range of tasks in different fields such as image and speech recognition, natural language processing, and scientific modeling. The function replicating the biological behavior of neurons such as thresholding, non-linearity etc is replaced in the ANN by an activation function which calculates the output of a node based on each of the input and their weights.

## 2.5 Neural Network Quantum States

The Quantum Neural Network States (NNQS) represents an emerging intersection between quantum mechanics and machine learning. Traditional methods of solving for the states of quantum systems, such as exact-diagonalization method becomes computationally infeasible due to the exponential growth in the Hilbert state with each new particle in the system. In contrast once the neural network is trained, predicting the solution using NNQS is computationally less demanding and provides a potential alternative. Essentially, NNQS are neural networks designed to estimate the wavefunctions of quantum systems which in theory is computationally better and faster than using traditional methods to compute the wavefunctions.

### 2.5.1 NNQS Architecture

Neural Network Quantum States use neural network architectures, such as Restricted Boltzmann Machines (RBMs) and feed-forward networks, to encode the wavefunction of quantum systems. In our case, we have a system with $N = 20$ spin-$\frac{1}{2}$ particles. A complete representation of all the states in such a system requires $O(2^N) = O(2^{20})$ coefficients, which is proportional exponentially to the system size of N particles. Thus using traditional methods like the exact diagonalization method becomes inefficient.

Now using neural network we can tackle this problem. First in order to represent a quantum state with neural network, we start out with a variational ansatz, where the wavefunction $|\Psi(\Omega)\rangle$ depends on parameters $\Omega$. Here, $\Omega$ represents the weights and biases of the neural network which can be tuned during the training phase and then fixed at the end to provide the best estimate of ground state. The aim here is to adjust $\Omega$ such that the network's output approximates the system's ground state.

$$E = \frac{\langle \Psi(\Omega)|H|\Psi(\Omega)\rangle}{\langle \Psi(\Omega)|\Psi(\Omega)\rangle}$$

This 2.5.1 is the loss function which the optimizer tries to find a lower ground state to iteratively. In order to train our neural network we use standard machine learning procedures. We use exact diagonalization technique, in particular the Lanczos method to compute the exact ground state energy and based on this data, train the network (7). After the training is done the weights and biases are fixed and now the neural network can be used to estimate the ground state.

Neural networks can represent the expectation values of the quantum states more efficiently, circumventing the exponential complexity by capturing essential correlations with fewer parameters. This efficiency allows NNQS to approximate wavefunctions of high-dimensional quantum systems without directly calculating the full state space.

### 2.5.2 Barren Plateau

Using the gradient descent optimization techniques which is used here to estimate the ground state, the optimizer usually gets stuck in a plateau like environment in the parameter space this is known as the Barren Plateau. This usually occurs when the parameter space is very large. Machine learning algorithms translate an optimization problem to a cost function and try to minimize the cost repeatedly by each iteration of training. When we are working with a large number of parameters the search space for the algorithm is very complex and may have local minimas or extended plateaus which can cause the algorithm to search aimlessly for the optimal solution in such environment which is a hard problem to tackle. These types of occurrences are generally known as barren plateaus in variational quantum algorithms.
Many quantum algorithms start their search randomly which increases chances of encountering a barren plateau. When the slope for the cost function appears flat, the algorithm does not know where to explore to go deeper to find the solution, thus appearing stuck. Barren plateau is an obstacle for using machine learning to study such systems since the advantage in using NNQS is lost when stuck in this state. Random initialization of parameters, large number of parameters, large circuit depth and measurement noise are all factors which can contribute to reaching a barren plateau state (8).

In order to minimize chances of Barren plateau, some techniques such as careful parameter initialization instead of random and training the quantum circuit layer by layer instead of optimizing all the parameters thereby making the parameter space smaller can be used (8).

# 3. Method

In this study, employing a combination of computational and machine learning techniques is used to solve quantum many-body problems. Specifically targeting the Ising and Heisenberg models. The approach involves using NNQS to estimate the ground state of the model and comparing it against the traditional Lanczos method as a benchmark. This section details the methods implemented to define the system and design the neural network and evaluate the performance of the neural network. Initially, the hamiltonians for each Ising and Heisenberg models are established. Subsequently, NNQS architecture is made tailored for each of the model utilizing tools from the NetKet library (9). Then the optimization process using gradient descent is implemented for both the models and finally the output results are benchmarked against the exact ground state obtained from Lanczos method.

## 3.1 Defining Hamiltonian

### 3.1.1 Ising model

$$\mathcal{H} = -V \sum_{0}^{N-1} \sigma_i^{(z)} \otimes \sigma_{i+1}^{(z)} - \Gamma \sum_{0}^{N} \sigma_i^{(x)} \tag{3.1}$$

For obtaining energy and magnetization plots we vary the value of $\Gamma$ (transverse magnetic field) from 0 to 2 with increment of 0.1 and record the corresponding data for each $\Gamma$. We define a Hilbert space with $N = 20$ spin-$\frac{1}{2}$ particles.

### 3.1.2 Heisenberg model

Similarly, for the Hamiltonian for the Heisenberg model, we vary the value of $h$ from 0 to 2 with increment of 0.1 and record the corresponding data for each $h$. The Hilbert space in this case is also defined with $N = 20$ spin-$\frac{1}{2}$ particles.

$$Heisenberg : \mathcal{H} = J \sum_{0}^{N-1} \vec{\sigma_i} \otimes \vec{\sigma_{i+1}} - h \sum_{0}^{N} \sigma_i^z \tag{3.2}$$

## 3.2 Defining the NNQS

To define the neural network structures, implementations from the NETKET library and Flax package have been used.

### 3.2.1 Ising model

**Code Listing 3.1.** NNQS Ising

```
class FFN(nn.Module):
    alpha: int = 1

    @nn.compact
    def __call__(self, x):
        dense = nn.Dense(features=self.alpha * x.shape[-1])
        y = dense(x)
        y = nn.relu(y)
        return jnp.sum(y, axis=-1)
```

For the Ising model, a feed-forward neural network is implemented as the NNQS. The network consists of a dense layer at the beginning which transforms the input parameters, followed by a ReLU activation function. The output of the activation is summed in the final sum layer, which provides a scalar magnitude representing the NNQS. This is a relatively simple architecture which is useful for its computational efficiency and accuracy for approximating the ground states in this problem.

### 3.2.2 Heisenberg Model

**Code Listing 3.2.** NNQS Heisenberg

```
    @nn.compact
    def __call__(self, x):
        x = nn.Dense(features=2*x.shape[-1], param_dtype=np.complex128, kernel_init=nn.
                                                      initializers.normal(stddev=0.1),
                                                      bias_init=nn.initializers.normal(
                                                      stddev=0.1))(x)
        x = nk.nn.activation.log_cosh(x)
        x = nn.Dense(features=x.shape[-1], param_dtype=np.complex128, kernel_init=nn.
                                                      initializers.normal(stddev=0.1),
                                                      bias_init=nn.initializers.normal(
                                                      stddev=0.1))(x)
        x = nk.nn.activation.log_cosh(x)
        return jax.numpy.sum(x, axis=-1)
```

Similarly for the case of Heisenberg model, an feed forward structure is implemented. The model begins by a dense layer that doubles the dimensionality of the input parameters and allows the parameters to be complex. The biases and weights (kernel in the code) in our NNQS is initialized based on a normal distribution. The activation function used is $log_{cosh}$, which introduces non-linearity in the system. The transformed output is passed through another pair of dense and activation layers, however this time the dimensionality remains the same for the dense layer. Finally the output is summed to give out a scalar magnitude which represents the NNQS for the Heisenberg model. This structure is more complex than the one used for Ising model, this is due to Heisenberg model having increased complexity.

### 3.3 Sampling the data

The sampling process allows the NNQS to approximate the quantum wavefunction by selecting the configurations based on probability distributions from the current NNQS.

**Code Listing 3.3.** NNQS Sampler

```
        ffnn = Model()
        sampler =  nk.sampler.MetropolisLocal(hilbert=hi)
```

For sampling in the feed forward structure, the Metropolis-Hastings algorithm is used. This algorithm is a Monte Carlo method which samples from the Hilbert space to create our variational state for training. This sampler can generate a set of spin configurations from the Hilbert space according to the probability distribution of the current wave function, these samples are then used to estimate the expectation value for operators like in our case, Hamiltonian for energy and magnetization.

In our $N = 20$ spin system, our sampler locally flips the spin of one particle at a time and checks if the new wave function is acceptable based on the probability of the wave function generated by the feed forward model. The variational state is generated after $2048$ such sampling and checks.

## 3.4 Optimization

The optimizer aims to find the optimal weights and biases for the NNQS such that it can predict the wavefunction for the given Hamiltonian. In this case, the Stochastic Gradient Descent optimization strategy and Stochastic Reconfiguration have been used. The optimizer updates the weights and biases of the NNQS iteratively after every pass through, this is done by trying to minimize the expectation value of Hamiltonian in 2.5.1.

**Code Listing 3.4.** NNQS Optimizer

```
op = nk.optimizer.Sgd(learning_rate=0.05)
sr = nk.optimizer.SR(diag_shift=0.1,holomorphic=True)
vs = nk.vqs.MCState(sa, ffnn, n_samples=2048)
gs = nk.VMC(hamiltonian=H,
            optimizer=op,
            preconditioner=sr,
            variational_state=vs)
gs.run(n_iter=300)
```

In SGD, the parameters $\Omega$ are iteratively updated in the direction of negative gradient of the energy, which helps to find the ground state.The learning rate 0.1 signifies the step size on the gradient to update the next parameter.

Stochastic Reconfiguration(SR) refines the optimization by taking into account the whole energy landscape in the parameter space.To improve convergence, it adjusts the parameter update by avoiding steps which are very different from each other. It updates the parameters after accounting for correlations between the parameters, thus scaling the gradients suitably. This whole process is repeated for 300 iterations, and at the end the final weights and biases for the $H$ is determined.

These strategies are however ineffective against Barren Plateau, since the optimization loop gets stuck in the local minima and considers it to be the lowest expectation value.

## 3.5 Evaluation

The ground state energy is calculated as the expectation value in eq 2.5.1. The magnetization is calculated as the expectation value of the $\sigma^z$ operator from the variational state. Both of these values are calculated for a range of transverse field from 0.0 to 2.0 with 0.1 step intervals. The Lanczos method as described in section 3.5, provides with a highly accurate estimate for ground state energy value.

**Code Listing 3.5.** Lanczos method for benchmark

```
evals = nk.exact.lanczos_ed(H, compute_eigenvectors=False)

exact_groundstate_energypersite = evals[0]/hi.size
```

The ground state energy is divided by number of sites to obtain exact energy per site. This is used as a benchmark to compare the ground state energy per site predicted by the NNQS to the exact ground state energy per site. By comparing for different transverse field strength, a reliable method is established to benchmark the NNQS outputs.

# 4. Results and Observations

This section evaluates and explains the observations and results obtained from estimating the Ising and Heisenberg models using NNQS.

## 4.1 Barren Plateau

It is observed that Barren plateau is a quite common challenge faced by the optimizer in training the NNQS for both the models. However in the case of Heisenberg model, encountering a Barren plateau is much more likely with the NNQS defined in 3.2. This is because with the increased dimensions of the hilbert space leading to increased complexity in the parameter space that ultimately leads to more chances of the optimizer gradient function to get stuck in a barren plateau in the parameter space.

Some examples below when the NNQS encounters a Barren plateau and when it does not:

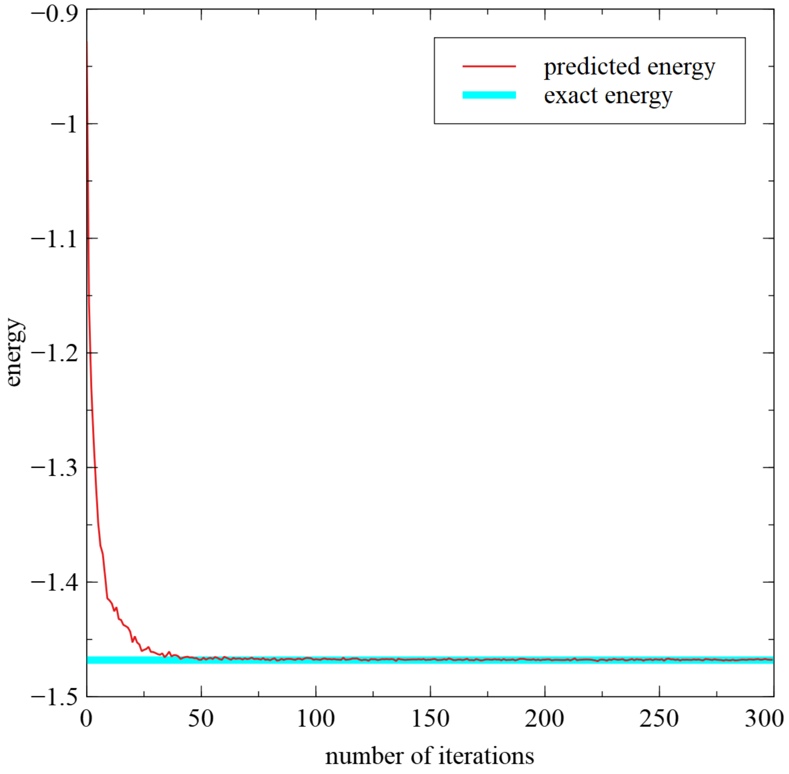

**Figure 4.1.** Converging Ising model plot for $\Gamma = 1.26$

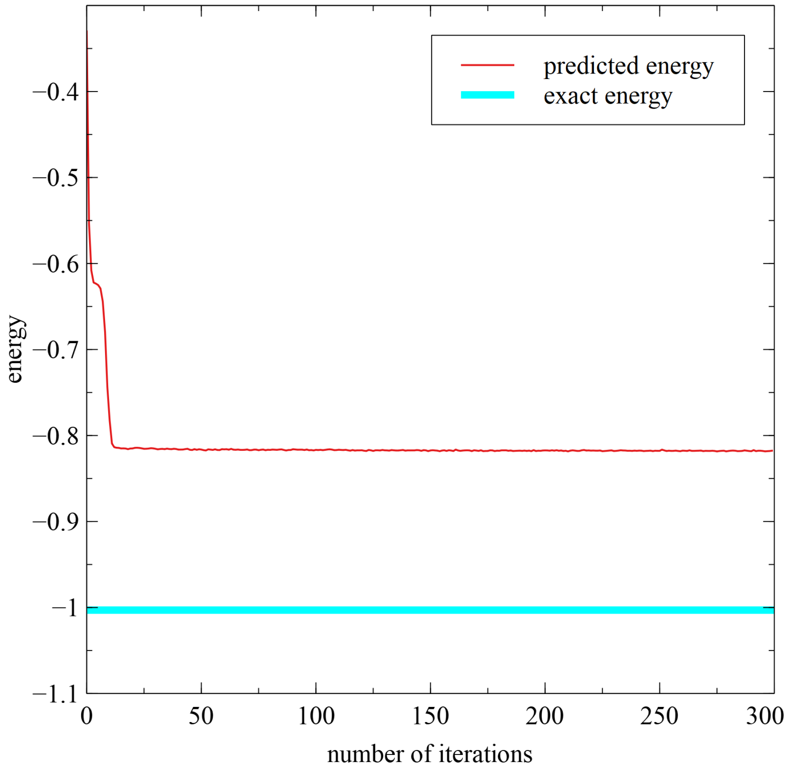

**Figure 4.2.** Ising model plot showing Barren plateau for $\Gamma = 0.11$

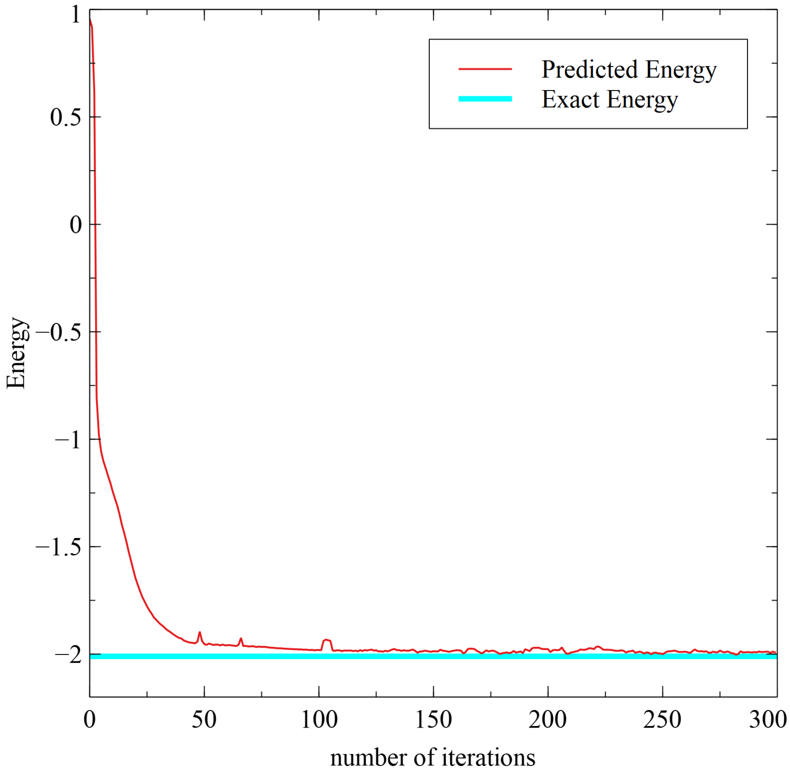

**Figure 4.3.** Converging Heisenberg model plot for $\Gamma = 2.0$

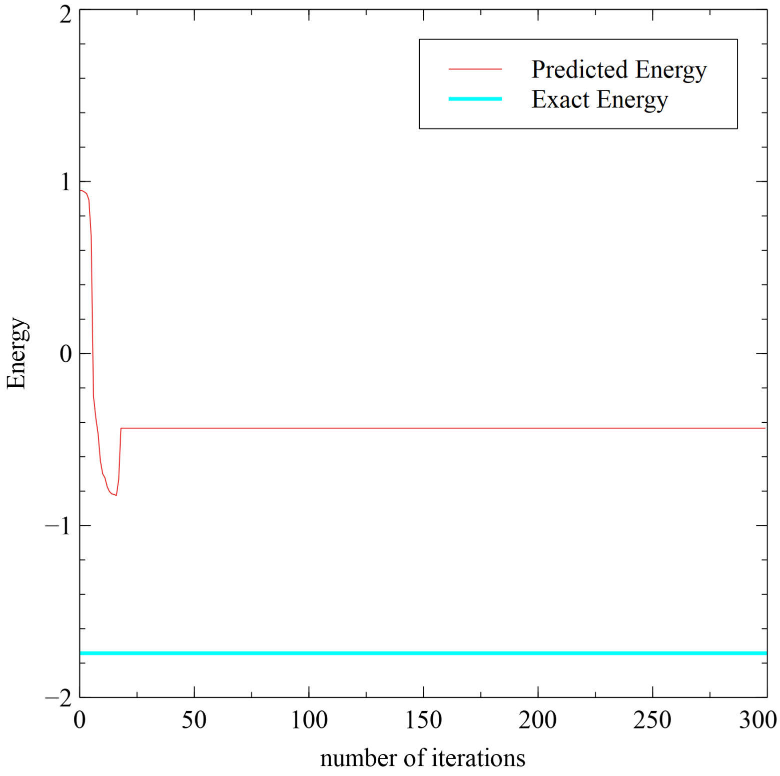

**Figure 4.4.** Heisenberg model plot showing Barren plateau for $\Gamma = 0.4$

The plots 4.1 and 4.2 shows the difference in converging when a Barren plateau is encountered while solving for Ising model.

## 4.2 Ising model

All the obtained data from the VMC run for the Ising model can be interpreted through the plots below which are based on the output data from the python code.

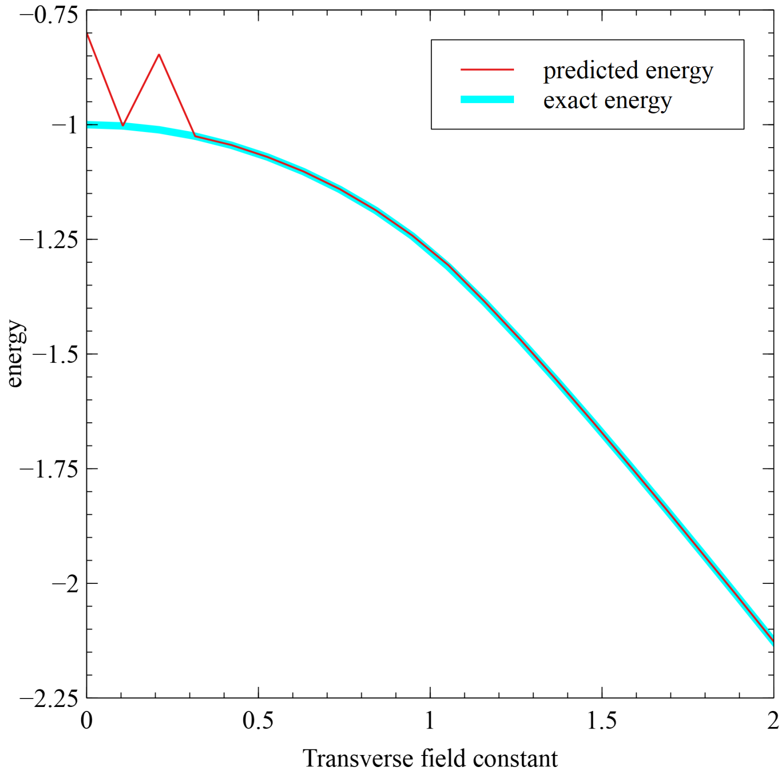

**Figure 4.5.** Energy per site vs Transverse field $\Gamma$

This figure 4.5 demonstrates the effectiveness of NNQS implemented in 3.1 to accurately predict the exact energy. It is also observed that the prediction is unstable around $\Gamma = 0$.

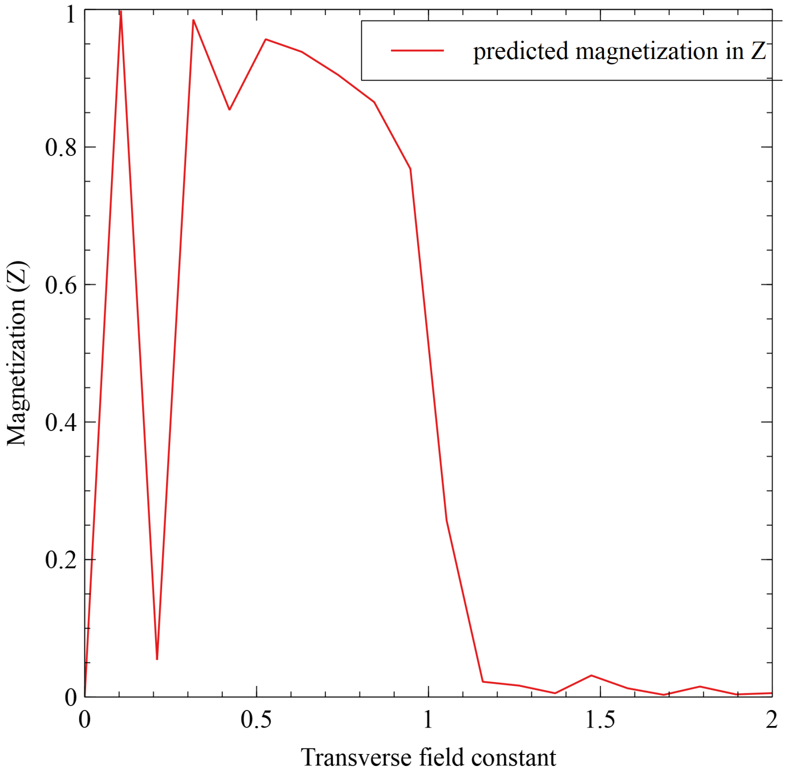

**Figure 4.6.** Magnetization(Z) vs Transverse field $\Gamma$

In Figure 4.10, The model accurately predicts that for low $\Gamma < 1$, the magnetization is highest which is due to the fact when no external magnetic field is applied the system tends to align in the $Z$ direction dominated by the $\sigma^z\sigma^z$ interaction. This also means that in high $\Gamma > 1$, the magnetization approaches 0 due to the external field dominating and aligning the spins in the $X$ direction rather than $Z$ direction. At $\Gamma = 1$, it can be observed that the system undergoes a phase transition in the $Z$ direction from ferromagnetic phase to paramagnetic which matches with the theoretical phase transition at $\Gamma = 1$. However it can also be observed that for $\Gamma < 0.5$, the predictions are unreliable.

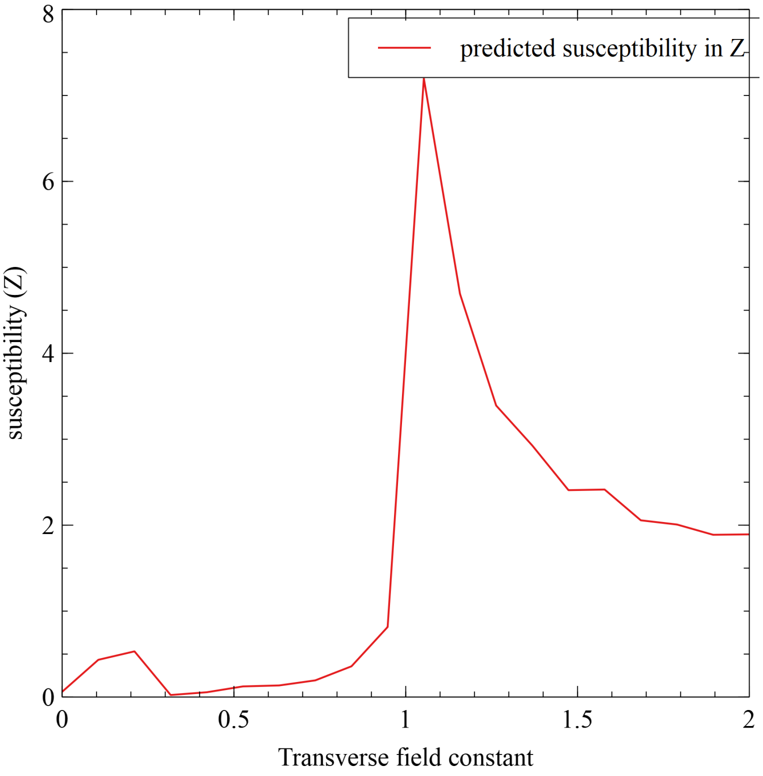

**Figure 4.7.** Susceptibility (Z) vs Transverse field $\Gamma$

Observations:

- The susceptibility in $Z$ reaches sharper peak at $\Gamma \approx 1$. This behavior is characteristic at phase transitions. At this point, small changes in external field can lead to large changes in the $Z$ magnetization, resulting in high susceptibility.

- Low susceptibility at $\Gamma < 1$. This is because the ferromagnetic state of the system is unaffected by the low external magnetic field from the $X$ direction at lower values of $\Gamma$ leading to close to $0$ susceptibility.

- Decrease in susceptibility in $\Gamma > 1$. This is because after $\Gamma$ reaches 1, the system has transitioned to being ferromagnetic in the $Z$ direction to being ferromagnetic in the $X$ direction. Due to this, stronger external magnetic fields in $X$ direction leads to decrease in magnetization in $Z$ direction and susceptibility decreases.

Confidence interval plot for predicted energy vs transverse field:

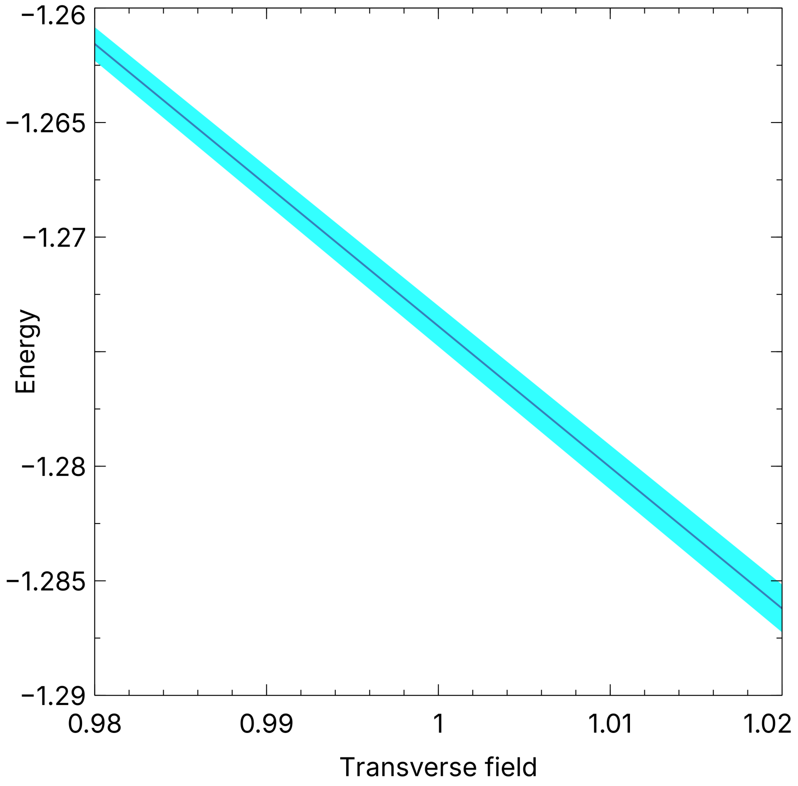

**Figure 4.8.** Confidence interval energy plot at $\Gamma = 1$

The plot 4.8 demonstrates a very accurate ansatz and the $95\%$ confidence interval is small in the order of about $10^{-4}$.

### 4.3 Heisenberg model

Following obtained results for the Heisenberg model:

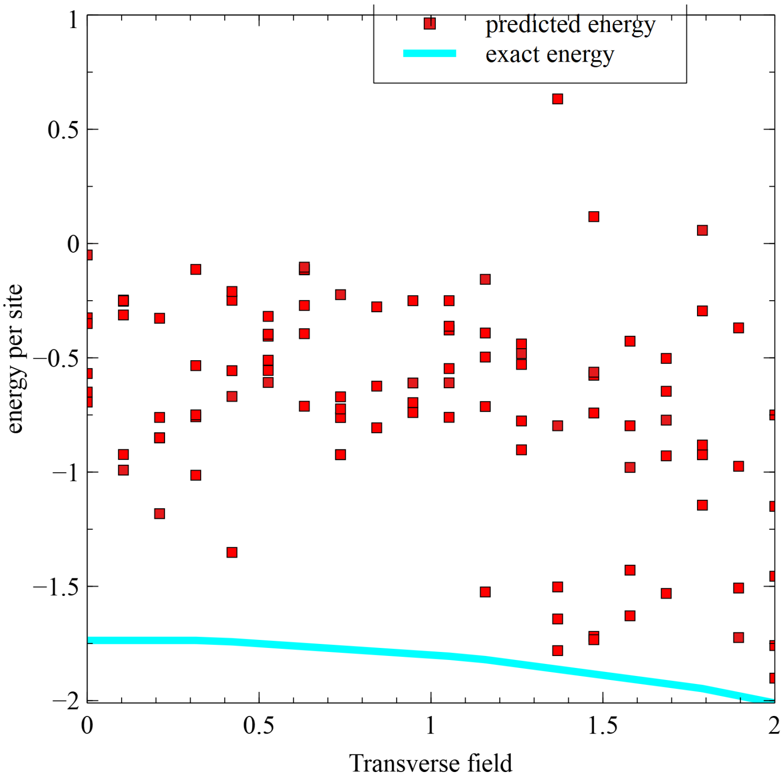

**Figure 4.9.** Energy per site vs Transverse field (h)

Some observations in figure 4.9:

- The variability in red scatter plots is due to the stochastic nature of the VMC, which relies on Monte Carlo sampling methods to optimize the VMC.

- For small $h < 1$, the predictions are closer together than compared to $h > 1$. This shows challenges in accurately predicting the energy for $h > 1$.

- The average moving line of all the red scatter plots resembles the curve of the exact energy plot although the predictions are not accurate.

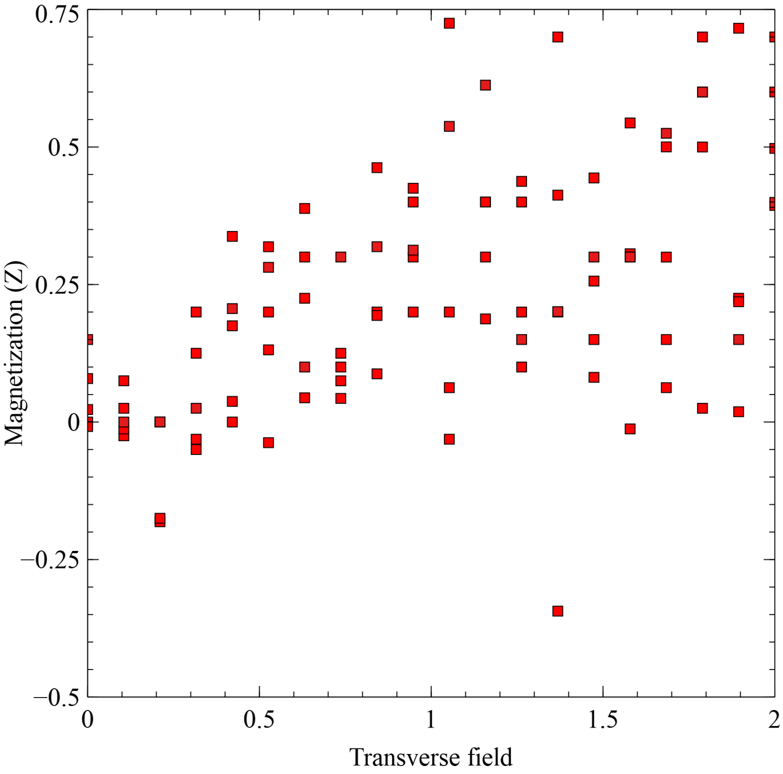

**Figure 4.10.** Magnetization (Z) vs Transverse field (h)

Observations in 4.10:

- From the plot, it can be observed that the Magnetization in $Z$ increases with increasing external field in $Z$ direction. When $h$ is around $0$, the external field does not affect the alignment of the system much and remains mostly unchanged. However with increasing field in $Z$ direction, the spins tend to align towards $Z$ and thus the trend of increase in magnetization in $Z$.

- The variability of the scatter plots shows that the NNQS is not predicting the energies consistently and accurately.

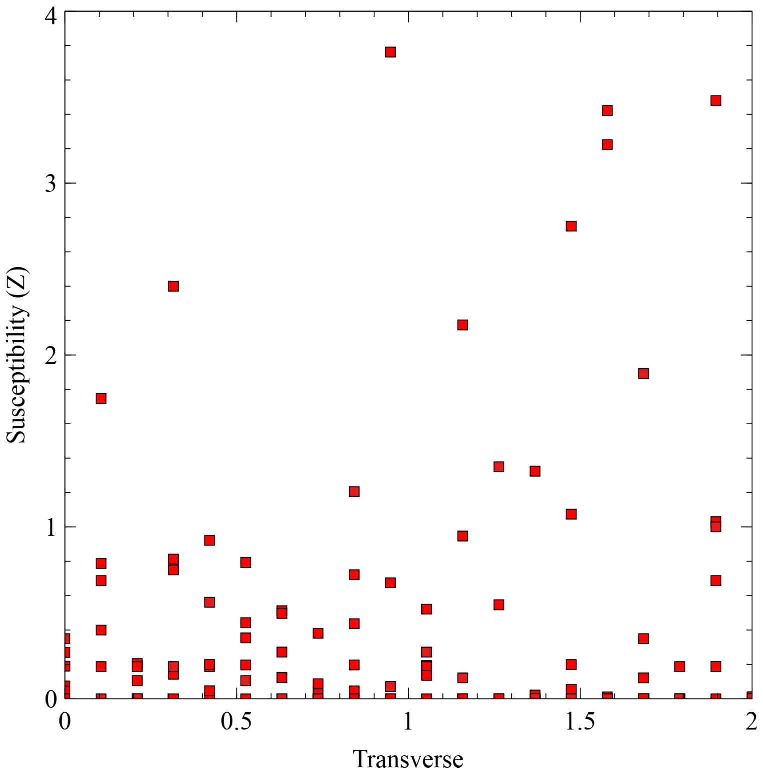

**Figure 4.11.** Susceptibility (Z) vs Transverse field (h)

Observations in plot 4.11:

- From the plot, a general trend of increasing susceptibility can be observed as the external transverse field in increased. However the plot consists of erroneous data points which makes the trend not clear.

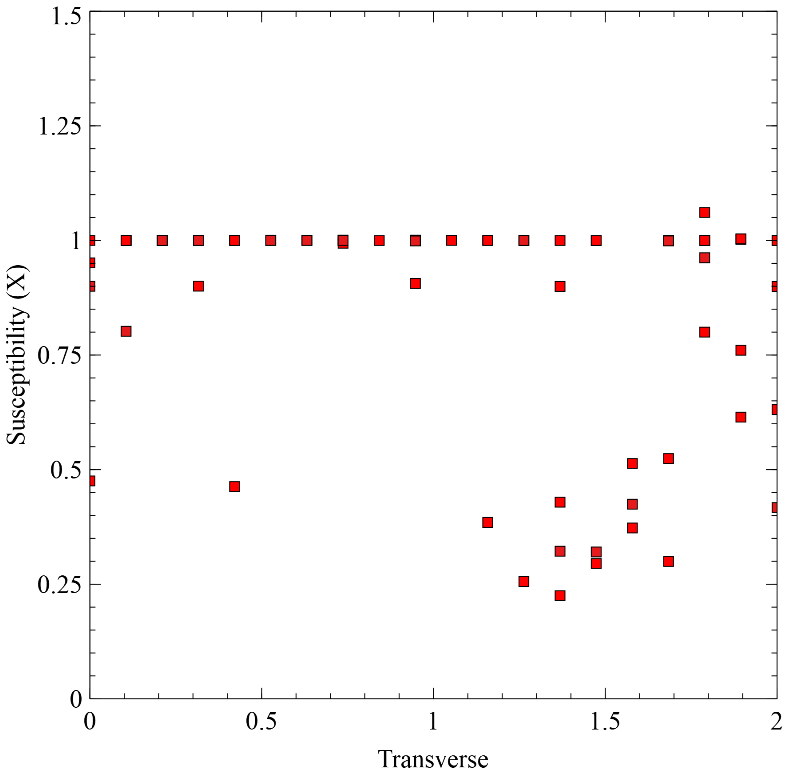

**Figure 4.12.** Susceptibility (X) vs Transverse field (h)

Observations in plot 4.12:

- A general trend can be observed here, which shows that the susceptibility in $X$ direction decreases with increasing magnetic field in $Z$ direction.

- Since the external field is applied in $Z$ direction and with stronger transverse fields, the system tends to align in $Z$ direction. This implies that the system does not prefer alignment in other directions ($X$ and $Y$) and thus leading to reduced susceptibility in $X$ with increasing transverse field in $Z$.

- It can also be observed that with weaker transverse field ($h < 1$), the susceptibility does not change since it does not affect the alignment much.

# 5. Conclusion

write as paragraphs.

### Achieved Effective Approximations of Ground State

- The NNQS method demonstrated promising capability to predict the ground state energies, magnetization and susceptibility for both the Ising and Heisenberg model.

- Comparisons with exact energy showed that NNQS performs well to predict the observables. This is true specially for Ising model which does not have a large parameter space like the Heisenberg model.

### Model-Specific Conclusions

- **Ising Model:** The NNQS successfully predicted accurate results with detailed trends and phase transitions. There were occasional inaccuracies, but most of the result is very accurate.

- **Heisenberg Model:** NNQS successfully predicted the general trends for the model, however it struggled with its exponentially large parameter space and encountered barren plateaus at almost every run and thus the predictions were quite varied.

### Challenges Faced

- **Barren Plateau Problem:** Particularly pronounced in predicting observables for Heisenberg model.

- **Unreliable Predictions at Low Field Strength:** Predictions were inconsistent at external field strengths near zero. This is likely caused by the sampler and optimizer used.

To counter these challenges, initialization techniques can be used for barren plateau. Adding more layers to the neural network such as convolutional layers to help solve for complex parameter space and predict better. Improved sampling techniques can also be employed to reduce variability and get accurate predictions.

Overall, this study demonstrated the capabilities of using NNQS to solve quantum many body physics while highlighting the shortcomings and possible future improvements

# Bibliography

[1] G. Carleo and M. Troyer, “Solving the Quantum Many-Body Problem with Artificial Neural Networks,” 6 2016.

[2] A. Kabelac, “One-and two-dimensional Ising model Statistical Physics Seminar by Prof. Wolschin,” tech. rep., 2021.

[3] E. Ising, “Beitrag zur Theorie des Ferromagnetismus,” *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[4] C. Lanczos, “An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators 1,” Tech. Rep. 4, 1950.

[5] D. Hernández-Lobato, “The Lanczos Algorithm,” tech. rep., 2013.

[6] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*. MIT Press, 2016.

[7] Z.-A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, “Quantum Neural Network States: A Brief Review of Methods and Applications,” 8 2018.

[8] “barren plateue1,”

[9] Filippo Vicentini (EPFL-CQSL), “Ground-State: Ising model âĂŤ NetKet,” 11 2021.