

Quantum Phase Recognition using Quantum Tensor Networks

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in
Computational Natural Sciences
by Research*

by

Shweta Sahoo
20171006

shweta.sahoo@research.iiit.ac.in



International Institute of Information Technology
Hyderabad - 500 032, INDIA
APRIL 2023

Copyright © SHWETA SAHOO, 2023
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Quantum Phase Recognition using Quantum Tensor Networks” by Shweta Sahoo, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Harjinder Singh

To my family and friends

Acknowledgments

I want to express my deepest gratitude to my advisor Prof. Harjinder Singh for his guidance, support and understanding through some tough times in my personal life and during my research tenure under him. Without him constantly pushing me and his incredible patience, I would not have completed this thesis on time.

I am immensely grateful to Utkarsh, my friend, my mentor and my co-author, for introducing me to the wonderful and exciting world of quantum computing and for helping me formulate the problem statement, helping me fix bugs in my code, helping me edit my thesis and paper and for always having my back.

I want to thank my fellow lab mates Adrian Alva, Animesh Sinha and Utkarsh again for engaging in many lively discussions on topics ranging from quantum computing to the effects of capitalism on the world.

I would like to extend my thanks to all my professors and peers in the Center for Computational Natural Sciences and Bioinformatics who taught me all the skills required for my research.

My IIIT journey would not be complete without my friends Saraansh Tandon, Zahabiya Doctor, and Arvind Bhaskar who have always encouraged me and have stood by my side like rocks.

Finally, I would like to thank Jeje, Maa, Aja, Aai, Papa, Mama, Daddy, Mummy, Bhai, Rose, and all my uncles, aunts and cousins for their immense love, support and blessings which made me the person I am today.

Abstract

The current generation of quantum devices fall in the noisy intermediate scale quantum (NISQ) era and are afflicted with qubit decoherence issues, scalability problems, circuit trainability and much more. Keeping these issues in mind, quantum-classical hybrid algorithms have been developed that break the problem into tasks for both classical and quantum processors. The classical processor is leveraged for computational tasks like arithmetic operations and optimizations which are easier for classical computers to handle, while computational tasks where quantum processors give better results like solving systems of linear equations come in the "quantum part" of the algorithm. A special class of quantum-classical hybrid algorithms is the variational quantum algorithms (VQAs) which has been used extensively to solve quantum chemistry problems, particle physics problems, optimization problems and much more as they provide a general framework for solving problems.

Classical machine learning has recently facilitated many advances in solving problems related to many-body physical systems. Given the intrinsic quantum nature of these problems, it is natural to speculate that quantum-enhanced machine learning will enable us to unveil even greater details than what we currently have. With this motivation, this thesis examines a quantum machine learning approach based on shallow variational ansatz inspired by tensor networks for supervised learning tasks.

We first start with an introductory chapter explaining the focus of our works and our contributions. We also discuss the applications of quantum computing in natural sciences. Then we dive into the necessary background of our works in detail. This includes sections on the basics of quantum computing, the NISQ era, variational quantum algorithms, tensor network ansatz and a literature review of the use of tensor networks in quantum computing. We introduce the circuits used by us and explain the tasks tackled in this work. We first look at a preliminary classical image classification task on the Fashion-MNIST dataset and study the effect of repeating tensor network layers on the ansatz's expressibility and performance. Finally, we take on the problem of quantum phase recognition for the Transverse Ising and Heisenberg spin models in one and two dimensions, where we were able to reach $\geq 98\%$ test-set accuracies with both multi-scale entanglement renormalization ansatz (MERA) and tree tensor network (TTN) inspired parametrized quantum circuits. Finally, we analyze our results and end with a discussion on the scope of future works.

Contents

Chapter	Page
1 Introduction	1
1.1 Motivation	1
1.2 Research Focus and Contribution	3
1.3 Outline	4
1.4 Quantum computing in Natural Sciences	5
2 Background	6
2.1 Quantum Computing	6
2.1.1 Qubits and the Bloch sphere	6
2.1.2 Measurement	7
2.1.3 Pure and Mixed States	8
2.1.4 Quantum Evolution	9
2.1.5 Quantum Circuits and Quantum gates	9
2.1.6 The Noisy Intermediate-Scale Quantum era	11
2.1.7 Variational Quantum Algorithms	12
2.1.7.1 Parameterized Quantum Circuit	13
2.1.7.2 Data Embedding	13
2.1.7.3 Objective Function	15
2.1.7.4 Optimization	15
2.2 Tensor networks	16
2.2.1 Notation	16
2.2.1.1 The Tree Tensor Network	19
2.2.1.2 The Matrix Product State	20
2.2.1.3 The Multi-scale Entanglement Renormalization Ansatz	20
2.2.2 Tensor networks in quantum computing	21
2.3 Spin Systems	21
3 Machine Learning using Quantum Tensor Networks	23
3.1 Circuit Architecture	25
3.2 Image Classification	27
3.2.1 Dataset	27
3.2.2 Encoding Strategy	27
3.2.3 Optimization and Hyperparameters	28
3.2.4 Results	28
3.3 Quantum Phase Recognition	30

3.3.1	Models and Data generation	30
3.3.1.1	1-D Transverse-field Ising Model	30
3.3.1.2	1-D XXZ Heisenberg Model	30
3.3.1.3	2-D Transverse-field Ising Model	31
3.3.1.4	2-D XXZ Heisenberg Model	31
3.3.2	Training	31
3.3.3	Results	33
4	Conclusions	34

List of Figures

Figure	Page
1.1 Moore’s Law	1
1.2 Google’s Sycamore quantum computer.	2
1.3 Overlap between ansatz and solution states.	4
2.1 Bloch Sphere.	7
2.2 The measurement operator.	8
2.3 Pure and mixed states.	8
2.4 A four qubit quantum circuit.	9
2.5 Some commonly used quantum gates	10
2.6 The NISQ Era.	11
2.7 The general workflow of a VQA.	12
2.8 Barren Plateaus.	14
2.9 Cost function dependent barren plateaus.	15
2.10 Tensors of various ranks.	17
2.11 Some general tensor contractions	17
2.12 A tensor network with rank 0.	18
2.13 Splitting a node with rank 2.	18
2.14 Splitting a node with rank 3.	19
2.15 Classical TTN tensor network and its circuit realization	19
2.16 Classical MPS tensor network and its circuit realization	20
2.17 Classical MERA tensor network and its circuit realization	20
2.18 A 2-D lattice of an eight spin system in paramagnetic phase	22
3.1 Variational workflow using MERA TN	24
3.2 Unitary blocks and their analysis: The possible choices of unitary blocks for building variational ansätze are (a) $V(\vec{\theta})$, which can represent any element from $SU(4)$ group, and (b) $U(\vec{\theta})$, which is a two-qubit entangling unitary. For comparing the effectiveness of built TTN and MERA tensor network ansätze, we perform (a) expressibility analysis based on the Jensen-Shannon divergence of fidelity distributions of generated parameterized states with that of Haar states (lower the better), and (b) entangling power analysis based on the Meyer-Wallach measure (higher the better)	25
3.3 Variational workflow using Tree TN	26
3.4 Performance of the MERA tensor network ansatz on the Fashion-MNIST dataset	28

3.5	Prediction probabilities of phases with MERA based ansatz: (a) for transverse-field Ising model in 1-D case (noiseless simulation), and (b) For transverse-field Ising model in 2-D case (executed on IBMQ Nairobi (<i>ibmq_nairobi</i>), 7-qubit hardware [1])	32
4.1	4 qubit MERA tensor network ansatz used for phase recognition	34
4.2	Tensor network ansätze for four-spin systems: Structures of variational ansätze based on the (a) tree tensor network (TTN) and the (b) multi-scale entanglement renormalization ansatz (MERA) tensor network. (c) Modified structure of MERA tensor network ansatz with changed first unitary block	35

List of Tables

Table		Page
3.1	Pairwise accuracy on the classes of the Fashion-MNIST dataset for one layer of the MERA tensor network	29
3.2	Pairwise accuracy on the classes of the Fashion-MNIST dataset for three layers of the MERA tensor network	29
3.3	Pairwise accuracy on the classes of the Fashion-MNIST dataset for five layers of the MERA tensor network	29
3.4	Performance of the TTN and MERA tensor networks on recognizing correct phases of various XXZ Heisenberg (XXZ-HM) and transverse-field Ising (TFIM) spin systems on one-dimensional (linear) and two-dimensional (rectangular) lattices. For eight spin systems, simulations were performed numerically on a quantum simulator, and results were averaged over five trials. Whereas for the four spins systems, along with similar numerical simulations, experiments were also executed on the IBMQ Nairobi (<i>imbq_nairobi</i>) [1], a seven-qubit quantum hardware, and the best results out of three trials are being reported here.	32

Chapter 1

Introduction

1.1 Motivation

The evolution of modern classical computers from the first generation vacuum tube era to the transistor era and now to the current integrated circuit era has been responsible for great advances in computational capacity. A look at the number of transistors in a dense integrated circuit over the years shows that this number has been doubling every two years (Fig. 1.1). This is referred to as the Moore's law [2]. The Moore's law has set targets for advancements in semiconductor technology for decades now.

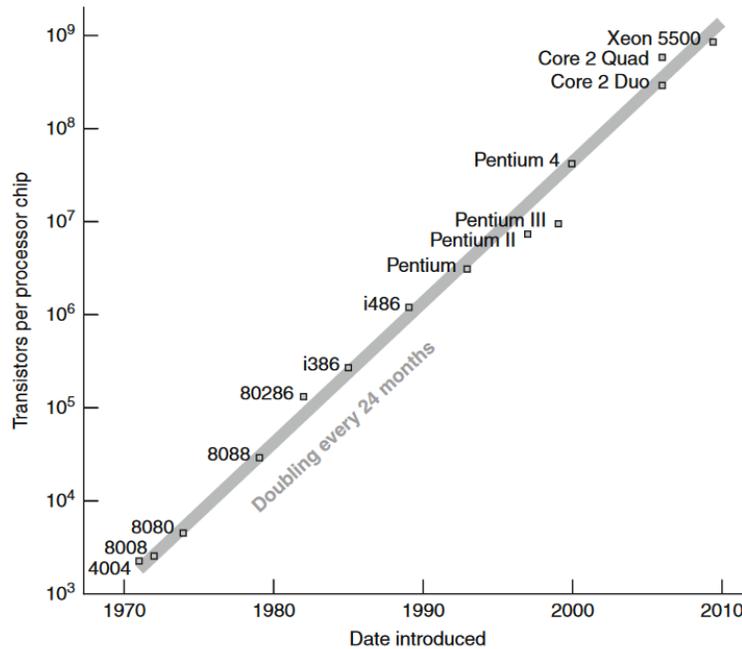


Figure 1.1: The graph shows the doubling of the number of transistors per processor chip every two years in accordance with the Moore's law [2]. The straight line graph in a logarithmic scale points to the exponential nature of this trend.

The design of semiconductor technology is becoming increasingly more complex and is reaching molecular limits. We are reaching a point where transistor size cannot be any smaller. This has contributed towards a slowdown of the Moore's law. However, this does not indicate a complete halt in the progress of semiconductor technology. It is reasonable to expect more efficient and powerful classical computers in the predictable future.

The biggest chink in the armour of even the fastest classical computers is computational tasks such as molecular simulation or finding prime factors of large composite numbers. These problems are exponential in nature and even with the advancements made in classical computing, it is simply not possible for any classical computers in the foreseeable future to solve them.

This is where quantum computers step in. The idea of quantum computing was born in the year 1980 with Paul Benioff's proposal of a quantum mechanical model of the Turing machine [3]. The theory that quantum computers could simulate things that are classically intractable was proposed by Richard Feynman in 1981 [4]. It was in the year 1994 that a quantum algorithm for finding prime factors of an integer was developed by Peter Shor [5]. This is especially important because this is a threat to RSA (Rivest, Shamir, Adleman). RSA is a public-key cryptosystem that uses the fact that factoring the product of two large numbers is hard for classical computers [6]. RSA is the oldest and the most used system for data transmission.

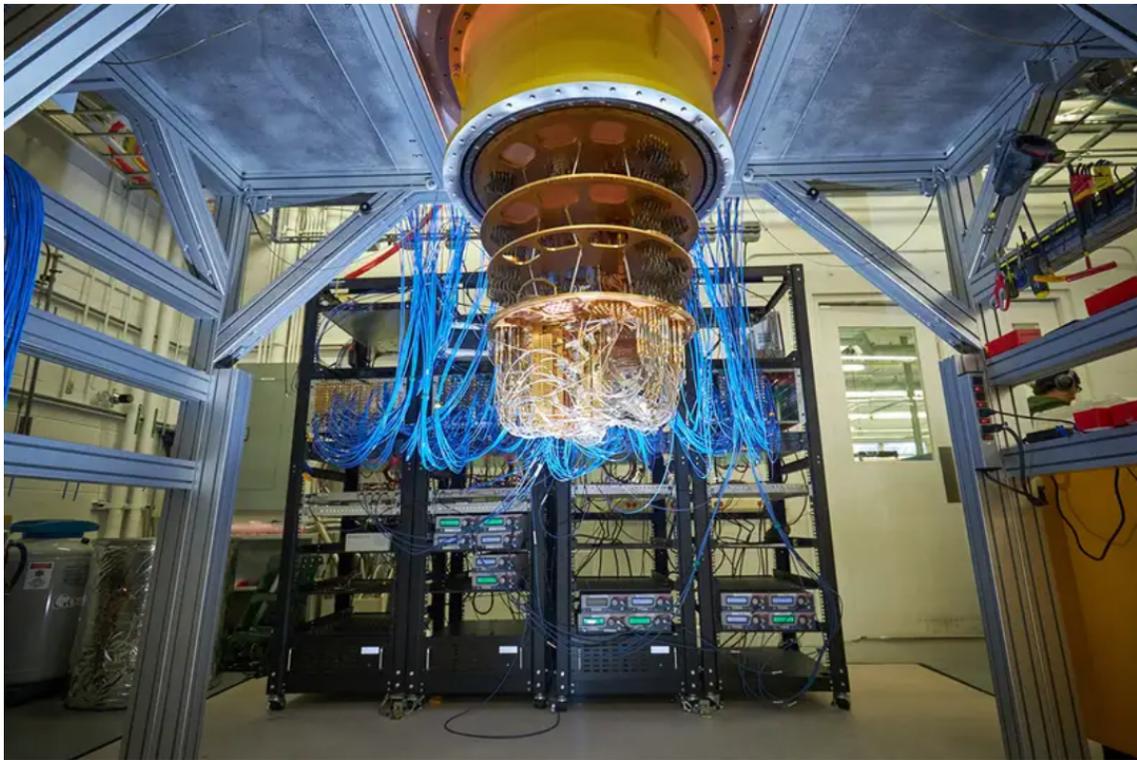


Figure 1.2: Google's Sycamore quantum computer is pictured above [7]. The Sycamore quantum computer has a 53 qubit processor.

The most exciting development in the field of quantum computing came in 1998 when the first two qubit quantum computer was created [8]. Quantum computers use the quantum principles of superposition and entanglement in their computations. They use quantum bits or 'qubits' instead of the normal bits used in classical computing.

The current era of quantum computing faces enormous challenges. This era is called the noisy intermediate-scale quantum (NISQ) era. In this work, we focus on the trainability aspect of variational quantum algorithms that leverage the available NISQ devices.

1.2 Research Focus and Contribution

This current era of quantum computing is as exasperating as it is exciting. The exasperation comes from the available technology that faces problems like qubit decoherence, scalability, circuit depth, circuit trainability etc. However, it is also an incredibly exciting time to be a researcher in this field as we are on the path towards fault tolerant quantum devices. Moreover, many algorithms have been developed to leverage the existing classical and quantum technologies to perform complex computational tasks. This set of quantum-classical hybrid algorithms, called variational quantum algorithms (VQAs), have been the subject of focus of this thesis. More specifically, we focus on the trainability of tensor network inspired ansatz for image classification and quantum phase recognition tasks.

We have tried to answer the following questions in our work:

1. Can the performance of the ansatz be increased?

A major component of the quantum variational workflow is the parameterized circuit or the ansatz. The performance of the algorithm is greatly dependent on it. We look at two main tensor network inspired ansätze and study unitary blocks based on metrics like expressibility and entangling capability of the ansatz to find the best unitary blocks that will evolve the input state to our target state. We also see the effect on expressibility and entangling capacity of the ansatz on increasing the layers of such circuits with hierarchical structures and using different unitary blocks.

2. How does our ansatz perform on more complex tasks?

We study the performance of our chosen unitaries arranged in a tensor network circuit on complex tasks. We have extended the previous works done in literature by attempting to perform quantum phase recognition on 2D spin systems with eight spins. These 2D systems are generally more challenging to deal with than the spin systems on 1D lattice that have been studied till now.

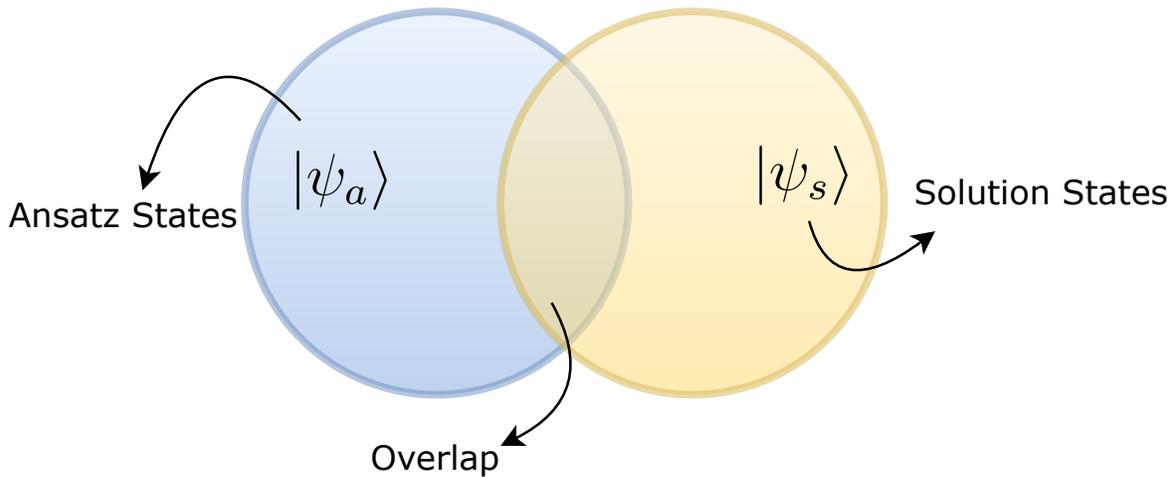


Figure 1.3: Our main goal is to maximize the overlap between the solution states and the states that the ansatz evolves the input states to, which we refer to as ansatz states here.

1.3 Outline

This thesis has been broadly divided into 4 main chapters:

- Ch. 1** In this introductory chapter we give a brief background on the history of quantum computing and the challenges faced when using NISQ devices. We discuss the focus of our works and our contributions. We end the chapter with a brief overview of the applications of quantum computing in natural sciences.
- Ch. 2** This chapter focuses on topics used in our works. We start with a basic run through of quantum computing topics, covering qubits, their representations, quantum measurement, pure and mixed states, quantum evolution, quantum circuits and quantum gates. We then move on to a discussion on the NISQ era and variational quantum algorithms. Then a quick overview of tensors, their notations, tensor networks and types of tensor networks is given. Finally, a brief introduction to spin systems is given.
- Ch. 3** We start off this chapter with a description of the circuit architecture used by us. We then go on to introduce the image classification and phase recognition tasks done by us. We explain the different experimental setups and the datasets used. Finally, we report the results obtained by us.
- Ch. 4** In this chapter, we analyze the results of the image classification task on the Fashion-MNIST dataset and phase recognition tasks on the various spin models in detail and provide our conclusions.

1.4 Quantum computing in Natural Sciences

Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

— Richard Feynman

Researchers are actively working towards leveraging near term quantum devices to provide a quantum advantage for natural science problems. Certain problems like that of drug discovery and protein folding require the efficient simulation of many-body systems to understand and predict complex structures, functions, interactions with other compounds, and much more. Let us take an example of designing a drug molecule with 50 atoms that can be built using say 10 unique atoms. For any given geometric configuration, the number of molecules that need to be studied is 10^{50} . Now, adding in various possible structural conformations increases the number of possible candidates to be studied. Clearly, this problem is exponential in nature and even with the approximate methods, it is bound to become classically intractable for the present computational methods. These systems are inherently quantum mechanical in nature and simulating quantum many body systems using classical computers is computationally expensive. Thus, such problems provide promising applications of quantum computers [9].

Quantum computers can be used to prepare and evolve complex, highly entangled quantum states. This makes them a better contender at simulating many-body quantum systems. The current generation of quantum hardware, referred to as NISQ devices, should be viewed as stepping stones towards advancements in quantum technology. Hybrid form of algorithms called variational quantum algorithms implemented on NISQ devices have been used extensively in the field of quantum chemistry, particle physics, condensed matter physics, and life sciences.

For example, the variational quantum eigensolver (VQE) [10] is an algorithm belonging to the class of VQAs that is used to find the ground state of a given molecular Hamiltonian. The subspace search VQE (SSVQE) [11] is used to find the excited states of a molecular Hamiltonian. Similarly, in the field of material sciences, quantum computing is currently being used for calculating various properties of different materials as it allows for better modeling and efficient simulations, leading to results that are closer to the actual experimental results. In particle physics, the ground state energies of light nuclei have been computed and research is being done to simulate sub-atomic many body physics on quantum processors [12]. Quantum computing is also being used in life sciences to study the protein folding problem [13] and for drug design and drug discovery [14].

In our work, we have performed image classification on images as a preliminary task to analyze the learning ability of tensor network ansatz presented by us. We have then performed quantum phase recognition tasks on one-dimension and two-dimension spin systems defined by the Transverse Ising and Heisenberg models of interaction using our tensor network ansatz. We show that our proposed ansätze were capable of learning phase labels and transition which are critical for identifying important characteristics of quantum many-body systems.

Chapter 2

Background

This chapter provides a brief overview of the concepts used in our studies. It is divided into three main sections: quantum computing, tensor networks and spin systems.

The section on quantum computing starts with a very basic introduction of the fundamentals of quantum computing. A discussion on qubits and their representation using the Bloch sphere, quantum measurements, the different kinds of states a qubit can be found in, and quantum evolution of the state of qubits is done. This is followed by an outline of quantum circuits and various quantum gates available for computation. Finally, a short discussion on the current NISQ systems and a class of hybrid classical-quantum algorithms called variational quantum algorithms (VQE) is done.

The second section is on tensor networks and covers two main topics, First, tensor network notation and some of the types of tensor networks and second, tensor networks in quantum computing.

The third and final section of this chapter is on Spin systems. The concepts of exchange interaction, 1D and 2D spin lattices and quantum transitions are discussed here.

2.1 Quantum Computing

2.1.1 Qubits and the Bloch sphere

Computation requires manipulation of information that needs to be stored and utilized in various ways. In the classical framework, information is stored in bits which can have the discrete, deterministic values 0 or 1. In the quantum framework, quantum bits or qubits are the vectors of information. A qubit is essentially a description of a two level quantum system. Mathematically, a qubit is represented as a superposition of states $|\Psi\rangle$:

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.1)$$

$|\Psi\rangle$ is a normalized vector in the Hilbert space \mathbb{C}^2 with $|0\rangle$ and $|1\rangle$ as the orthonormal basis of this two dimensional vector space. Hence, $|\alpha|^2 + |\beta|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ are known as the computational basis states [15].

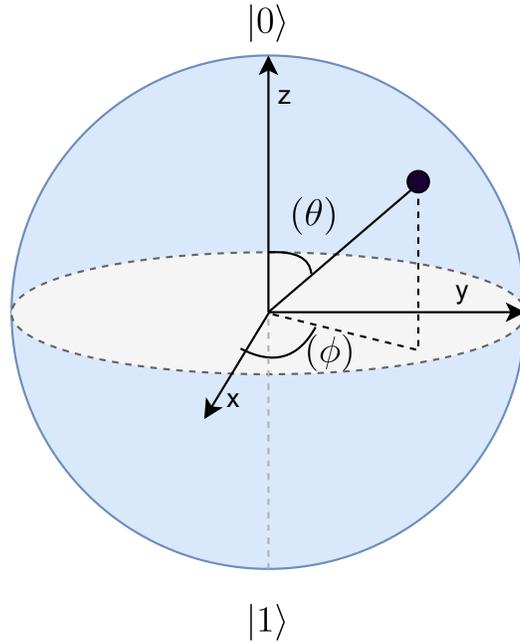


Figure 2.1: Representing a qubit's state as a vector in the bloch sphere

The state of a qubit can be represented geometrically as a vector in a 3 dimensional sphere called the Bloch sphere as shown in Fig. 2.1. The bloch sphere is a useful tool for the visualization of the state of a single qubit. Operations on the state of a qubit can be described using the Bloch sphere as well.

The qubit's state can be rewritten using the elevation angle θ and the azimuthal angle ϕ as:

$$|\Psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \quad (2.2)$$

2.1.2 Measurement

While a qubit does exist in a continuum of states between $|0\rangle$ and $|1\rangle$ as shown geometrically on the bloch sphere, which can have an infinite number of points on its surface, it would be incorrect to presume that a qubit can store an infinite amount of information. This is due to its behavior when a measurement is done. A measurement collapses the state of the qubit.

Measurements are done using measurement operators (M_m) (Fig. 2.2) acting on the qubit's state space. Measurement of a qubit in the computational basis yields the states $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$.

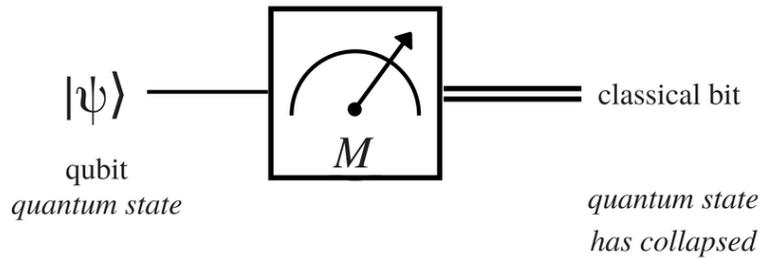


Figure 2.2: Measurement operator acting a qubit [16]. After a measurement, a qubit’s state collapses, yielding either the $|0\rangle$ or the $|1\rangle$ state. This resultant information can be stored in a classical bit as 0 or 1 as shown.

2.1.3 Pure and Mixed States

When the state of a qubit that can be represented as a single state vector $|\psi\rangle$, it is called a pure state. In cases where the state of a qubit is a statistical ensemble of state vectors with some probability distribution, we say that the qubit is in a mixed state.

Geometrically, a pure state is represented by points on the surface of the bloch sphere while a mixed state is represented by points inside the bloch sphere as shown in Fig. 2.3.

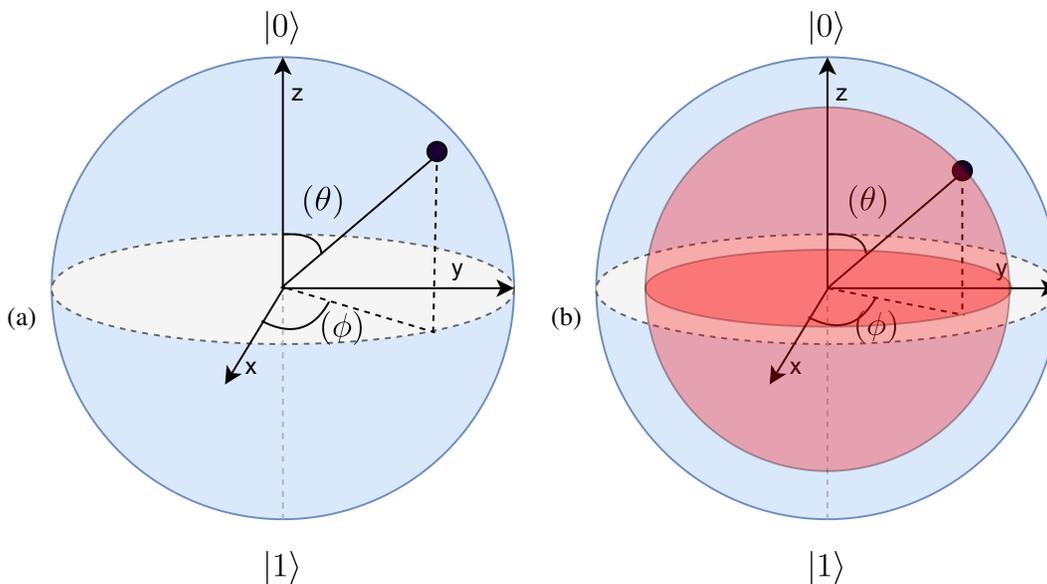


Figure 2.3: (a) A pure state in represented by a unit vector in the bloch sphere. (b) A mixed state is represented by a vector of length < 1 in the bloch sphere

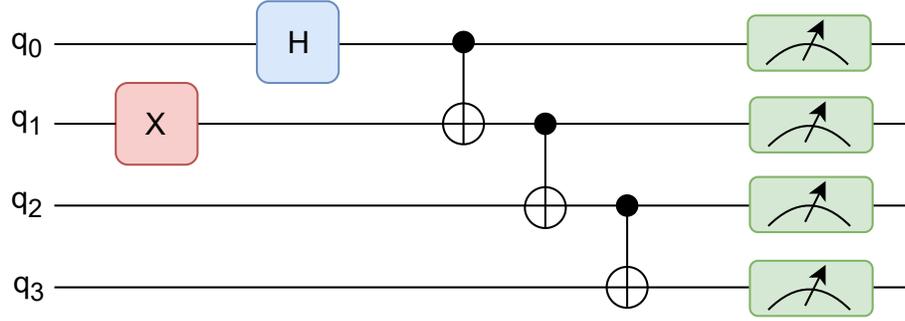


Figure 2.4: A four qubit quantum circuit with the NOT gate acting on q_1 , the Hadamard gate acting on q_0 and the CNOT gate acting on : q_1 as target and q_0 as control, q_2 as target and q_1 as control and q_3 as target and q_2 as control

The density matrix formulation is used for describing qubits in a mixed state. For a system in a statistical ensemble of state vectors $|\psi_1\rangle, |\psi_2\rangle, \dots |\psi_n\rangle$ with the respective probability distribution $p_1, p_2, \dots p_n$, the density matrix is given by:

$$\rho \equiv \sum_{i=1}^n p_i |\psi_i\rangle \langle \psi_i| \quad (2.3)$$

2.1.4 Quantum Evolution

Quantum evolution refers to the change in the state of the qubit with respect to time. The evolution of a quantum system from a pure state $|\psi\rangle$ at time t to a pure state $|\psi'\rangle$ at time t' is given by:

$$|\psi'\rangle = U |\psi\rangle \quad (2.4)$$

Where U is a unitary operator and is dependent on the times t and t' .

For mixed states, the density matrix evolves as:

$$\rho = \sum_{i=1}^n p_i |\psi_i\rangle \langle \psi_i| \xrightarrow{U} \sum_{i=1}^n p_i U |\psi_i\rangle \langle \psi_i| U^\dagger = U \rho U^\dagger \quad (2.5)$$

This implies that if the system was initially in state $|\psi\rangle$ at time t with probability p_i , then the unitary operator U evolves it such that at time t' , it will be in state $U |\psi_i\rangle$ with the same probability.

2.1.5 Quantum Circuits and Quantum gates

Quantum circuits are made up of qubits, which are often referred to as wires, and quantum gates that act on the state of these wires and evolve them (Fig. 2.4). Quantum gates are represented using unitary matrices U , that is $UU^\dagger = I$. These quantum gates can act on single qubits, like the NOT gate (X) or they can act on multiple qubits like the CNOT gate (Fig. 2.5).

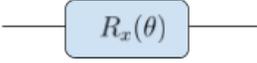
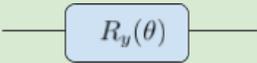
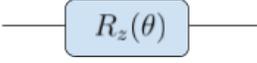
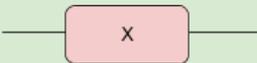
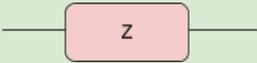
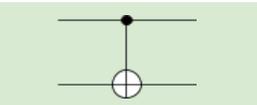
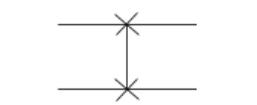
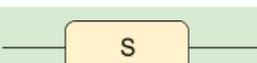
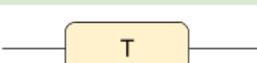
Operator	Symbol	Matrix
Rotation about x-axis ($\hat{R}_x(\theta)$)		$\begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$
Rotation about y-axis ($\hat{R}_y(\theta)$)		$\begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$
Rotation about z-axis ($\hat{R}_z(\theta)$)		$\begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$
Pauli X ($\hat{\sigma}_x$)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli Y ($\hat{\sigma}_y$)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli Z ($\hat{\sigma}_z$)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (\hat{H})		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
S		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
T		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$

Figure 2.5: Some commonly used quantum gates

Rotation gates $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ rotate the state vector $|\psi\rangle$ of the qubit about the x , y and z axes respectively in the bloch sphere by an angle θ and are particularly useful. They have been used a lot in the computations done in our studies. The matrix representations of these gates are given by:

$$\hat{R}_x(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad \hat{R}_y(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad \hat{R}_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad (2.6)$$

Rotations by π about the x , y and z axes are represented using the Pauli operators and are denoted using σ_x , σ_y , σ_z . The matrix representation of the Pauli operators is given below:

$$\hat{\sigma}_x(\theta) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \hat{\sigma}_y(\theta) = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \hat{\sigma}_z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.7)$$

2.1.6 The Noisy Intermediate-Scale Quantum era

As discussed earlier, qubits are able to represent complex entangled quantum systems and form the basis of quantum computing. Advances in the field of quantum computing have been done in an effort to reach quantum supremacy. Quantum supremacy refers to an era where no classical computer that exists now or in the predictable future can solve computational tasks that a quantum computer can.

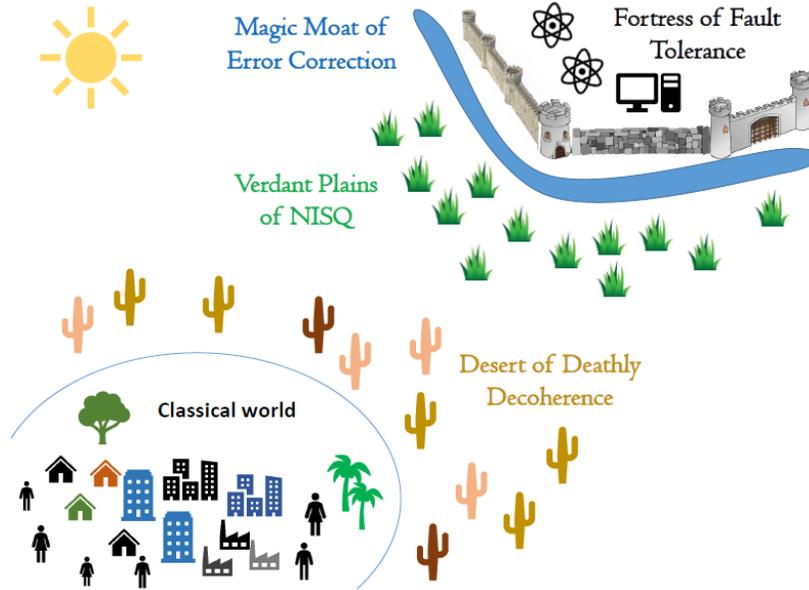


Figure 2.6: The NISQ era lacks robust error correcting codes that can deal with all the decoherence issues. There is still a long time to go till we reach the stage of fault tolerant quantum devices [17].

While quantum computers are inherently more complex than classical computers, it does not mean that they are more powerful. Quantum computers have been shown to perform better than classical computers in some instances. The most famous example of this is that no efficient classical algorithm exists for finding the prime factors of large composite integers [18].

The current quantum devices fall in the noisy intermediate-scale quantum (NISQ) era. This era is characterized by noisy devices with around 50 to a few hundred qubits, which is more than what the existing supercomputers can simulate by brute force. Even though the number of qubits we can simulate is important, we need to also focus on the quality of qubits. The performance of NISQ devices is severely affected due to quantum decoherence [18]. We do expect to see robust, fault tolerant systems in the future with advances in error correcting codes and better qubit quality (Fig. 2.6).

The NISQ devices should be perceived as a stepping stone to do more advanced and powerful fault tolerant quantum technologies. A new powerful set of algorithms called variational quantum algorithms have been developed to work on NISQ devices. These algorithms are the subject of discussion of the next subsection.

2.1.7 Variational Quantum Algorithms

As discussed earlier, the NISQ era is plagued by limited qubits, shorter circuit depth, decoherence issues and much more. A special class of classical-quantum hybrid algorithms called variational quantum algorithms, or VQAs for short, have been proposed to leverage the current NISQ devices in an effort towards achieving quantum advantage [19].

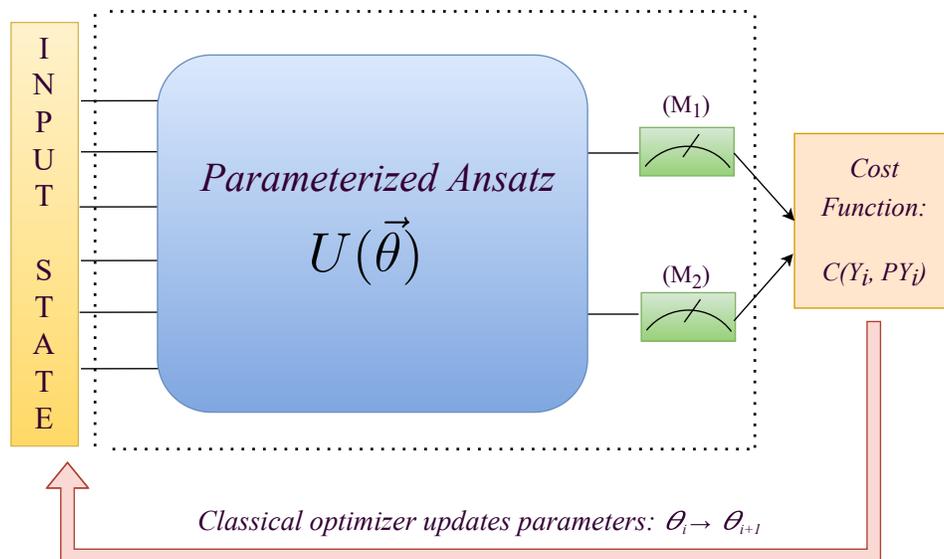


Figure 2.7: The general workflow of a VQA is broken up into a quantum section, consisting of the state preparation and the quantum circuit, and the classical section, consisting of the cost function evaluation and optimization[20].

A typical VQA algorithm consists of an objective function, a parameterized quantum circuit, a data encoding strategy and an optimization scheme. Formulating quantum algorithms to solve problems is hard and not intuitive. A huge advantage of the variational quantum algorithms is that they provide a general framework for solving problems. Moreover, they leverage the best of both classical and quantum computing. The use of classical computing for optimization helps keep the circuit depth short which reduces issues caused by noise.

In this section, we will discuss in detail about the objective function of VQAs, the parameterized circuit, the data encoding strategies and optimization.

2.1.7.1 Parameterized Quantum Circuit

The parameterized quantum circuit (PQC) or the ansatz is a quantum circuit which takes in certain parameters that are optimized using the variational quantum algorithm. The structure of the ansatz is very important and studies are being done on the trainability, efficiency and accuracy of various ansätze.

Aside from qubit decoherence and low circuit depth issues, there are other problems that affect the trainability of the ansatz and are ansatz specific. One such problem is the barren plateau problem (Fig. 2.8). It is a trainability problem that occurs when the cost function landscape turns flat as the algorithm is run, that is the gradient becomes zero. This makes reaching the global minima difficult. It has been shown that the likelihood of the presence of barren plateaus increases with increasing expressibility of the circuit and increasing depth [21].

The choice of ansatz varies from problem to problem. Ansätze that are problem specific are known as physically motivated ansatz, as seen in the variational quantum eigensolver (VQE) which is used to find the ground state energy of a molecular Hamiltonian. Ansätze that make use of the existing hardware are known as hardware efficient ansätze.

The parameterized ansatz can be denoted by a single unitary $U(\theta)$ that is applied to the qubit states that are inputted, where θ represents the parameters of the circuit. $U(\theta)$ can be split into a product of M unitaries applied sequentially as:

$$U(\theta) = U_M(\theta_M)U_{(M-1)}(\theta_{(M-1)})\dots U_1(\theta_1) \quad (2.8)$$

2.1.7.2 Data Embedding

Applications of VQAs have used both classical and quantum data. Since our quantum devices expect to work on quantum data, classical data needs to be represented as quantum states in the Hilbert space. This process is known as quantum embedding. A quantum feature map is used to embed classical data into a quantum state. This is an important step in the algorithm as the choice of embedding strategy affects the performance of the algorithm.

The basis embedding is used in places where the classical data is in the form of binary strings. This embedding involves associating each data point with the computational basis of the space. Hence, in

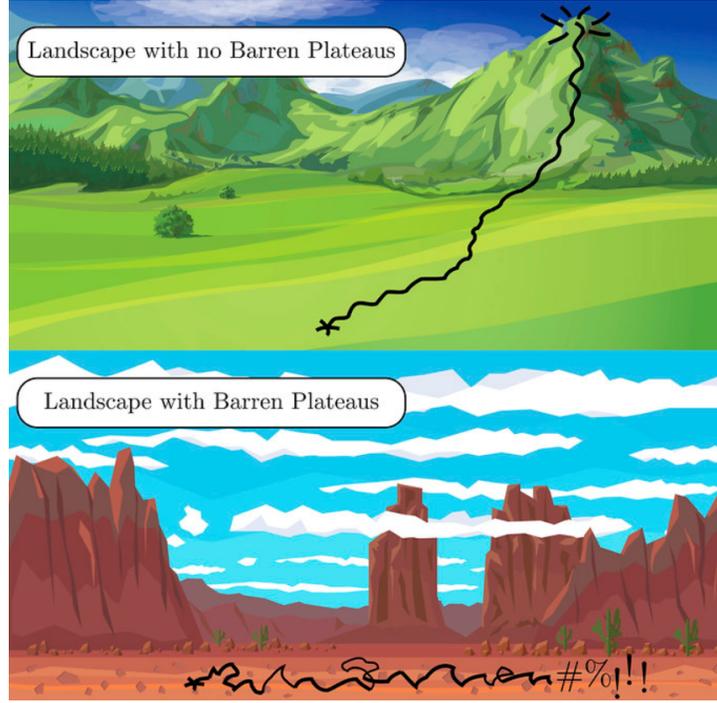


Figure 2.8: Mathematically, the phenomenon of barren plateaus can be explained as the cost function partial derivatives on an average becoming exponentially flat with the system size. In this case, reaching the global minima becomes very difficult, affecting the training of the ansatz [22]

this embedding the classical binary string essentially corresponds to its respective quantum state in the Hilbert space.

For a dataset $D = (x_1, x_2, \dots, x_p)$ where each data point x_i is an n -dimensional vector, the basis embedding requires N qubits such that $N \geq n$.

The dataset D is given by:

$$|D\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^p |x_i\rangle \quad (2.9)$$

which is just the superposition of the computational basis states with each state having equal probability.

The amplitude embedding encodes each data point into the amplitudes of the quantum state. The data $D = (x_1, x_2, \dots, x_p)$ is first normalized and each data point x_i is encoded into the amplitudes of a 2^n dimensional quantum state $|\psi_{x_i}\rangle$:

$$|\psi_{x_i}\rangle = \sum_{j=1}^N x_{ij} |j\rangle \quad (2.10)$$

where $N = 2^n$, x_{ij} is the j^{th} element of x_i and $|j\rangle$ is the j^{th} computational basis state. We need $n \geq \log_2(pN)$ qubits to encode dataset D .

There are many other types of embedding, like angle embedding, hamiltonian embedding etc. We have used the amplitude encoding in our works.

2.1.7.3 Objective Function

The objective function or the cost function is a function that needs to be optimized to get the solution to our problem. The cost function is defined on measurements taken on the qubit state and is given by the general formula:

$$C(\theta) = f(\rho_k, O_k, U(\theta)) \quad (2.11)$$

where f is the cost function, ρ is the set of inputs, O_k is the set of observables, and $U(\theta)$ is our parametrized circuit.

The cost function chosen is very important as it generates a cost function landscape whose global minima corresponds to the solution of our problem. It is seen that the phenomenon of barren plateaus is dependent on the choice of cost function used.

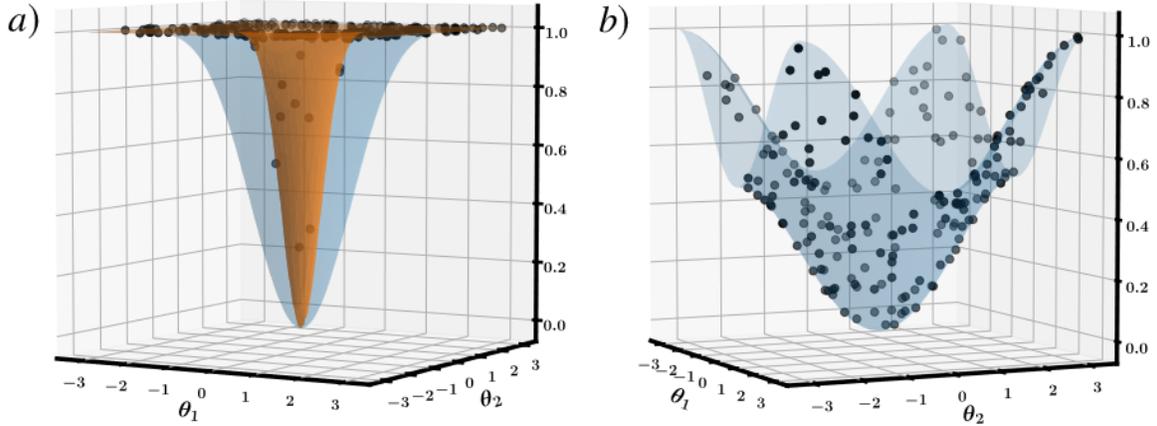


Figure 2.9: The cost function landscape with (a) a global cost function and (b) a local cost function. The usage of a global cost function causes the formation of barren plateaus in the landscape [23].

Studies have shown that the landscape becomes flatter with increase in number of qubits even when the circuit depth is less. This happens because the cost function is trying to compare two states that live in exponentially large hilbert spaces and the information is in the overlap. This overlap on average is going to be very small. It was found that the usage of a local cost function instead of a global one eliminates the barren plateaus in the loss landscape as shown in Fig. 2.9 [21].

2.1.7.4 Optimization

This is the normal classical optimization step where the parameters θ are updated according to the cost function.

$$\theta_{i+1} \leftarrow \theta_i - g_{opt}(\theta_i) \quad (2.12)$$

2.2 Tensor networks

Tensor networks are essentially approximations of very large tensors using smaller, easier to handle tensors [24]. They are widely used to simulate strongly entangled quantum systems. The usage of tensor networks has seen developments in quantum information theory, condensed matter physics and machine learning among other avenues. Due to the ability of tensor networks to describe quantum states it is useful to check their performance on quantum phase recognition tasks. In this section we will be discussing how tensor networks are represented, the types of tensor networks and finally we will be discussing the usage of tensor networks in quantum computing.

2.2.1 Notation

A tensor with rank n in m dimension is a mathematical object with n indices and m^n components and obeys certain transformation rules. Hence, the number of indices that the tensor has is its rank and the number of values that each index can take is the dimension of the tensor. For example, a tensor that can be represented as a scalar has a rank 0, a vector has a rank 1 and a matrix has rank 2 [25].

In linear algebra, tensors are represented using bases of the space and its respective components. The indices of the components vary in whether they are written as a superscript (for contravariant tensors) or in the subscript (for covariant tensors). Moreover, this notation also includes the usage of the Einstein convention, which is just a method of expressing summations in a short form by using repeated indices for the indices that are summed over. This way of expressing tensors becomes a bit difficult to handle when dealing with tensor networks.

The development of the graphical notation of tensors in 1971 by Roger Penrose made dealing with large tensors and tensor networks more intuitive and easier. In the graphical notation, the tensors themselves are denoted by solid shapes called nodes with their indices denoted by edges or lines jutting out from them as shown in Fig. 2.10. Therefore, the rank of the tensor is just the number of open edges. The dimension of the tensor cannot be gleaned from this graphical notation.

An important concept in the graphical notation of tensors is that of edge contraction and node splitting. Tensors are contracted by summing over their common index. In the graphical notation, tensor contraction is shown by joining the common index of the tensors as shown in Fig 2.11. This is equivalent to a matrix multiplication:

$$C_{ij} = \sum_k A_{ik} B_{kj} \quad (2.13)$$

Connecting the indices of the same tensor gives the trace of the tensor as shown in Fig. 2.11.

We see some more edge contractions in Fig. 2.11. As can be seen, in this notation, we can omit the names of the tensors and the indices. This leads to a cleaner, more coherent diagram. Moreover, as is evident, even for complicated contractions, the rank of the resulting tensor can be deduced at a glance

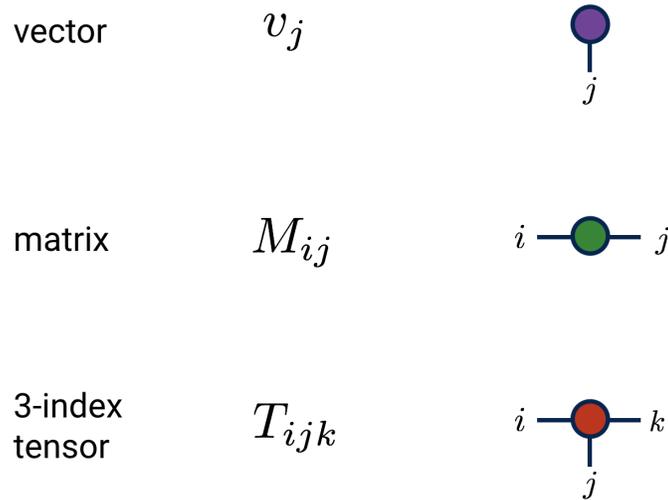


Figure 2.10: Tensors of rank one (represented as a vector) , two (represented as a matrix) and three [26]

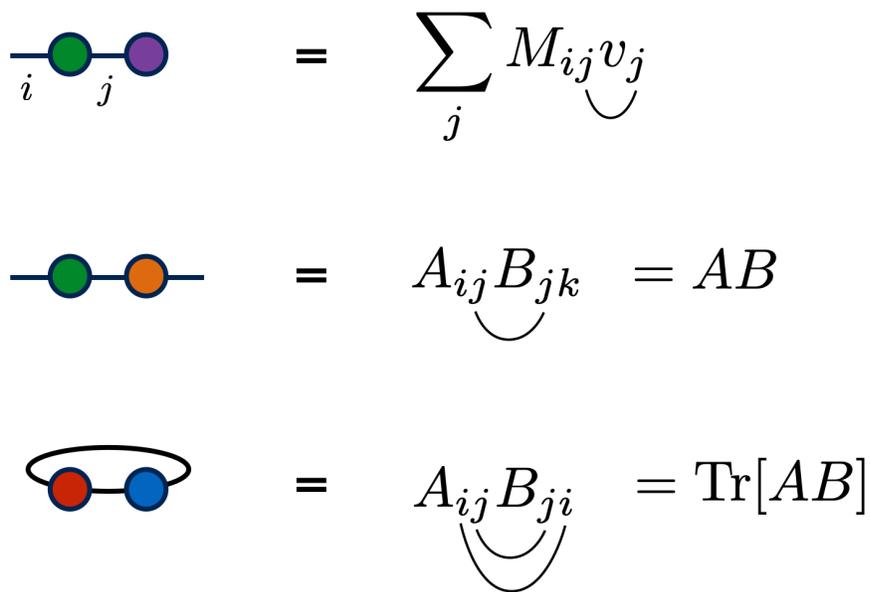


Figure 2.11: The figure above shows some general tensor contractions [26]

by counting the number of open edges. So for example in Fig. 2.12, since there are no open edges, we know, without doing any calculations, that the resultant tensor has to be a scalar.

To denote contravariant or covariant indices, one can simply use directed edges.

The second important concept that we need to talk about is node splitting. It is a method by which a tensor node is split into 2 tensor nodes. Fig. 2.13 shows the splitting of a rank 2 tensor. This is done by first doing an Singular Value Decomposition of the rank 2 tensor. This gives rise to three rank 3 tensor

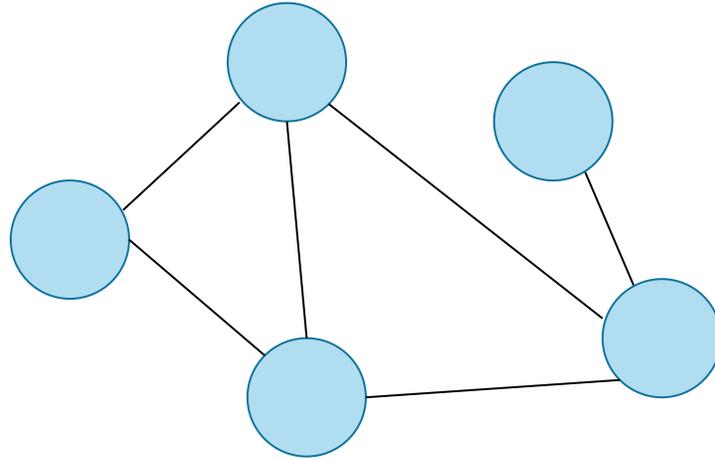


Figure 2.12: The figure shows a tensor network with rank zero which involves some complex contractions.

nodes that are joined together. These can be represented by matrices U , D and V as shown in Fig.2.13. From the resulting matrices, the diagonal matrix D is split into D_1 and D_2 by taking the square root of the elements of the diagonal. Now the edge between U and D_1 is contracted. Similarly the edge between D_2 and V is contracted. So now our initial tensor node has been split into two nodes. Note that even after splitting the node, the rank of the resultant node remains the same.

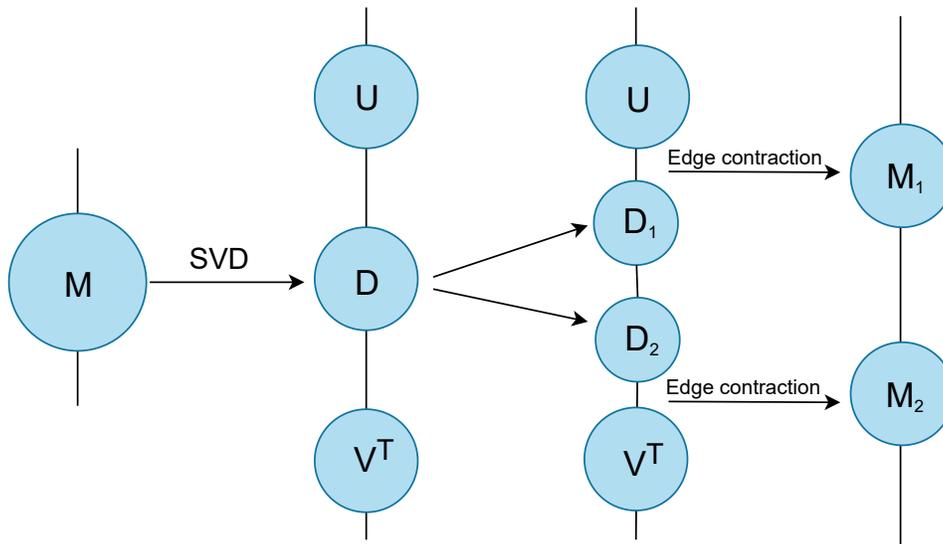


Figure 2.13: The figure shows a node of rank two splitting using Singular Value Decomposition (SVD).

Tensor nodes with rank greater than 2 can also be split as shown in Fig. 2.14. This example uses a tensor of rank 3. In this case, this tensor is first converted to a rank 2 tensor by flattening one index. The node is then split normally.

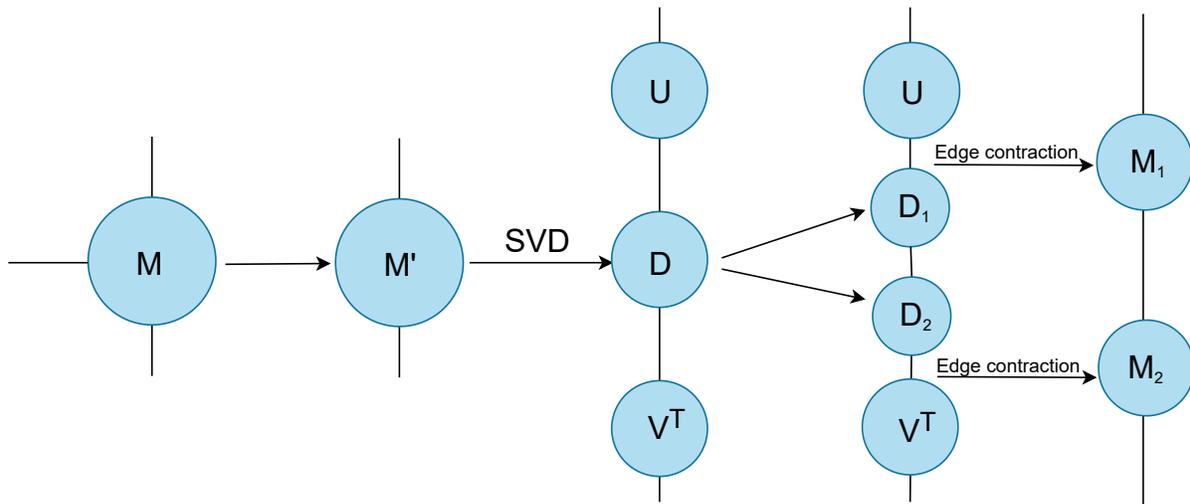


Figure 2.14: The figure shows a node of rank three splitting. In this case an index of the node is first flattened and the node is hence converted to a rank two tensor and is then split using SVD (Singular Value Decomposition) as before.

Certain tensor networks can be realized as quantum circuits and have been used as ansätze in variational algorithms. Three such tensor networks are:

2.2.1.1 The Tree Tensor Network

The tree tensor network (TTN) is characterized by a tree like structure (Fig. 2.15). Note that it is not necessary for the tree tensor network to have only two children per node. It can have multiple sub trees as children. For example, if there are three sub trees as children of the node, this corresponds to a three qubit unitary block in the ansatz [27].

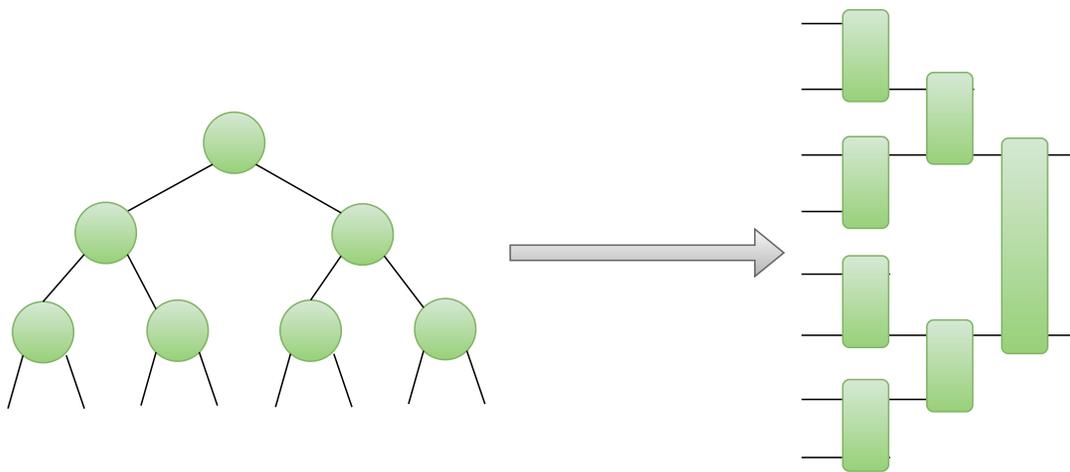


Figure 2.15: Classical TTN tensor network and its circuit realization

2.2.1.2 The Matrix Product State

The matrix product state (MPS) is a network of Rank 3 tensors as shown in Fig 2.16. The matrix product state tensor network is a special case of a tree tensor network. This tensor network has been studied extensively and many efficient algorithms have been developed using this [26].

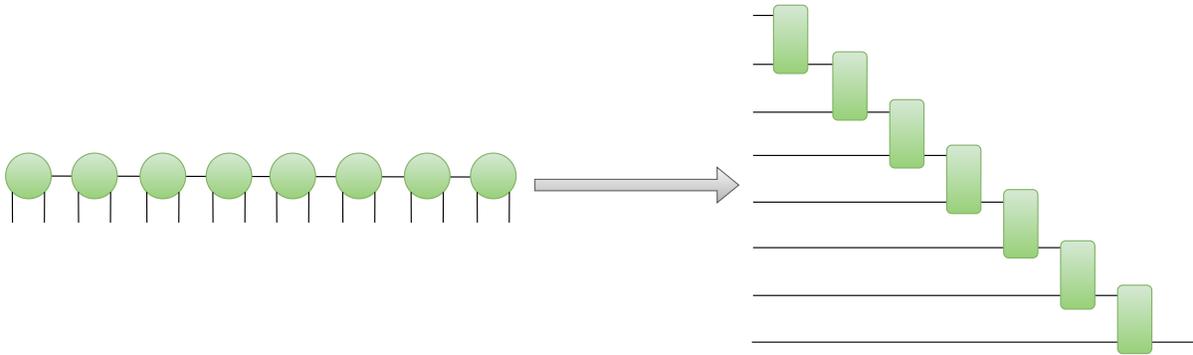


Figure 2.16: Classical MPS tensor network and its circuit realization

2.2.1.3 The Multi-scale Entanglement Renormalization Ansatz

The multi-scale entanglement renormalization ansatz (MERA) tensor networks produce highly entangled states. Their structure and corresponding quantum circuit [27] have been illustrated in Fig 2.17. The MERA tensor network ansatz has shown promising results in various computational tasks, which is the subject of discussion of the next section.

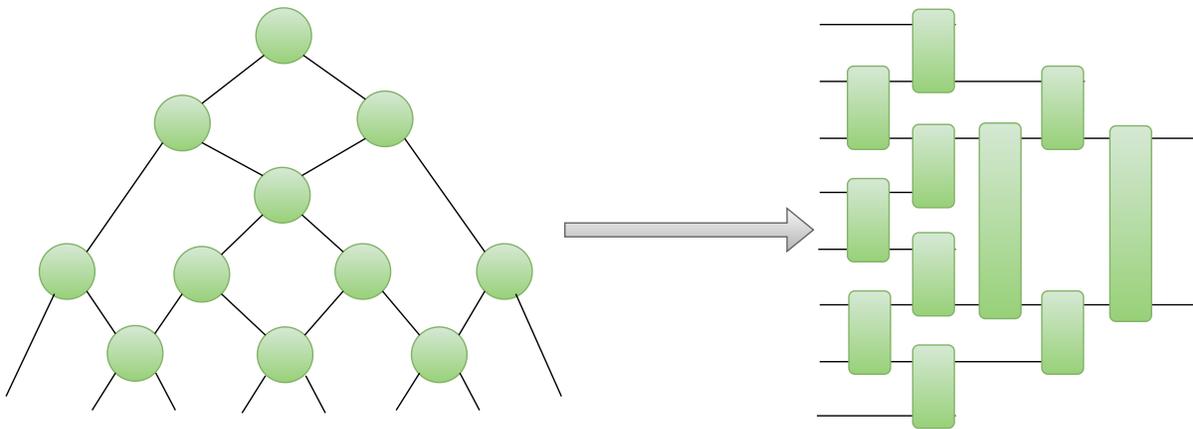


Figure 2.17: Classical MERA tensor network and its circuit realization

2.2.2 Tensor networks in quantum computing

As discussed before, tree tensor networks (TTN) [28] and multi-scale entanglement renormalization ansatz (MERA) [29] can be constructed using a quantum circuit [30, 31].

Recently, the use of quantum circuits based on tensor networks have been explored in the domain of machine learning for both generative [32] and discriminative tasks [30, 33, 34]. The bond dimension of a tensor network is the dimension of the index connecting smaller tensors together. The bond dimension, D , of a tensor network that has been realized using a quantum circuit is 2^v , where v qubits connect different subtrees. The main motivation behind using tensor network inspired quantum circuits is the increase in expressibility of the ansatz with increasing bond dimension to the point that for a sufficiently huge bond dimension, the entire state space can be covered. Dealing with such systems is computationally expensive in the classical scenario [30]. Moreover, it is highly likely that in the case of quantum data such as the wavefunction of a system, usage of classical methods will be intractable due to the exponential increase of information that needs to be encoded and computed with the increase in number of particles.

Circuits with a hierarchical structure like that of a MERA or TTN tensor network have been used in the classification of images like those in the MNIST dataset [33, 34]. A hybrid classical-quantum MPS-VQC has been used in image classification tasks with the MPS being the feature extractor for the images. While the MPS tensor network in this case is classical in nature, it can be replaced with an equivalent quantum circuit paving the way for the usage of quantum tensor networks as feature extractors [35]. Various tensor network ansätze have also been used for Quantum phase recognition tasks for the 1-D Heisenberg [34] and Transverse Ising models [36] with good results.

2.3 Spin Systems

The study of spin systems is important in order to understand the magnetic properties of a system at a macroscopic level. This is because the magnetic moment of an atom has contributions from the electron spins. The alignment of many such spins on a macroscopic scale defines the magnetic properties of the system. This alignment of spins is driven by the exchange interaction between the atoms. Exchange interaction is a short range, powerful interaction that occurs due to the electrical forces between electrons in the atoms [37, 38]. In our works we will deal with spin systems where the atomic dipoles are depicted by points on a 1-D and 2-D square lattice. The exchange interaction between atoms is limited to nearest neighbors and is given by the general formula:

$$\varepsilon = \alpha \mathbf{S}_1 \cdot \mathbf{S}_2 + \beta S_1^Z S_2^Z$$

\mathbf{S}_1 and S_2 are the spins of the two neighboring atoms in question (bold style represents the corresponding angular momentum vector). We will be considering variants of the special case of $\alpha = 0$ and $\beta = J$ which is the Ising model of interaction and $\alpha = J$ and $\beta = 0$ which is the Heisenberg model of

interaction [37]. Spin systems undergo quantum phase transitions. A quantum phase transition is a point of non-analyticity in the energy graph of the ground state of the Hamiltonian of the system caused due to quantum fluctuations at 0 K [39]. Since many complex models can be approximated as spin systems, quantum phase recognition allows us to derive and understand the properties of such systems.

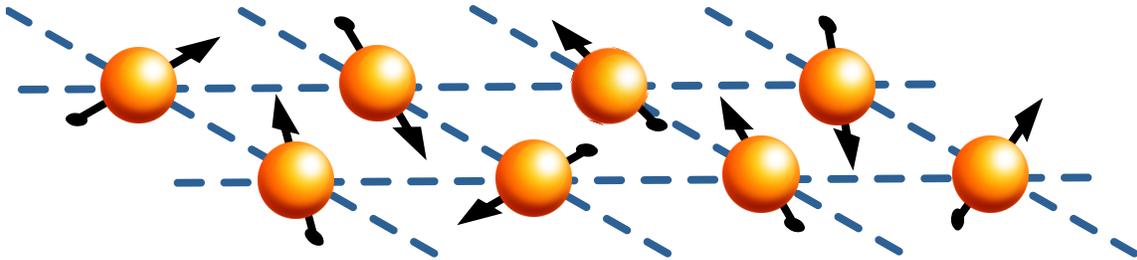


Figure 2.18: A 2-D lattice of an eight spin system in paramagnetic phase

Chapter 3

Machine Learning using Quantum Tensor Networks

Machine learning (ML) offers tools and techniques to learn and predict patterns that emerge in data. One crucial avenue of this pattern recognition task is classification, which involves predicting class labels for the input data and has found applications in speech recognition [40], biometric identification [41], object classification [42], disease identification [43] and many more. These applications result from immense leaps classical ML algorithms have made in dealing with various challenging datasets. Lately, based on these successes, ML algorithms have been used for problems related to many-body physical systems, such as recognizing phases of matter [44, 45]. Even though they have shown more promising results for studying relevant and useful many-body systems than the best contemporary classical algorithms, they still do not alleviate the sign problem [46], which usually emerges in such calculations and causes an exponential slowdown.

More recently, there has been an ongoing effort to develop quantum-enhanced machine learning algorithms that leverage quantum computers to tackle traditional ML problems [47]. These algorithms are typically based on a class of hybrid quantum-classical algorithms called variational quantum algorithms (VQAs), such as variational quantum eigensolver (VQE) [10] and variational quantum linear solvers (VQLS) [48] that have been used to find the ground state of a Hamiltonian and solve systems of linear equations on noisy intermediate-scale quantum (NISQ) hardware [18]. The relatively short depth of the parameterized quantum circuits (PQCs) used in these algorithms makes them an ideal candidate for achieving good results on NISQ devices without error correction codes. [19].

In principle, although PQCs are analogous to classical neural networks structurally, they can exploit additional computational resources due to the presence of quantum mechanical phenomena such as superposition and entanglement [49]. The basic working principle of VQAs is to optimize the parameters of PQC, also referred to as an *ansatz*, using a classical optimization routine to minimize a cost function defined on measurements taken on the qubits present in the *ansatz*. Therefore, the performance of these algorithms is majorly based on the structure of the *ansatz* [50]. Hence, it is crucial to analyze and have some basic insights into the *ansatz* for a particular problem or application to assess and improve their trainability.

In our works, we use a VQE-based algorithm to classify classical and quantum data. For the former, we look at the task of classification of Fashion MNIST dataset [51], whereas, for the latter, we tackle the problem of classification of the quantum phase of 1-D and 2-D transverse-field Ising and XXZ Heisenberg spin system. We employ multi-scale entanglement renormalization ansatz (MERA) and tree tensor network (TTN) states for building the ansatz for the variational routine. Finally, we also use expressibility and entangling capability analysis for choosing the structure of unitary block for these ansätze (Figs. 3.2a and 3.2b) that helps us make use of shorter-depth blocks than the general $SU(4)$ one suggested in [34].

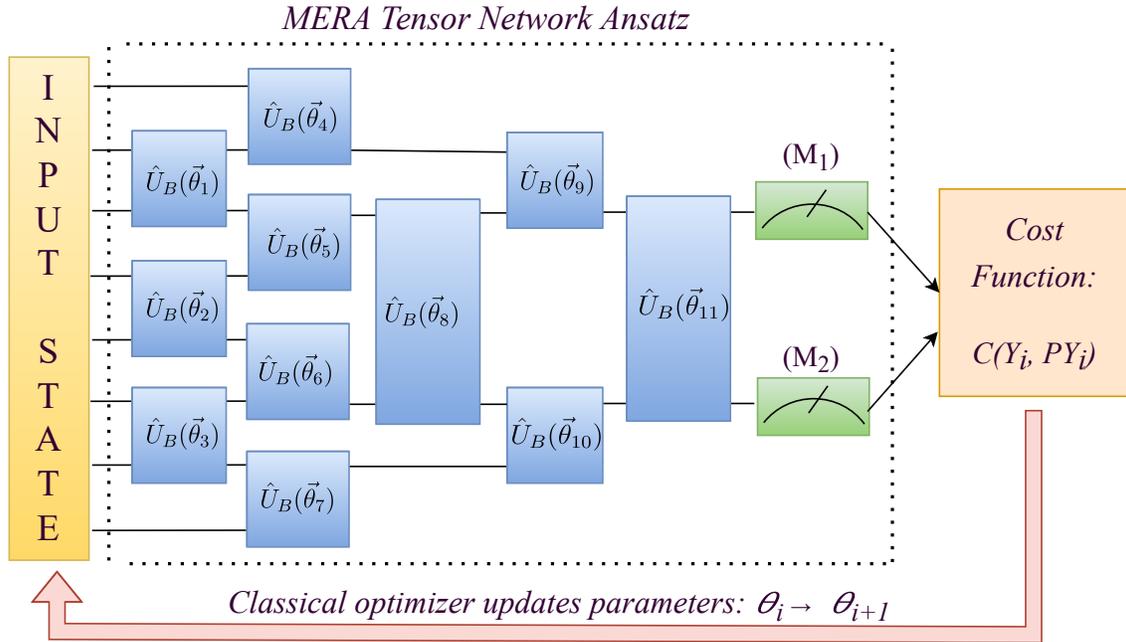


Figure 3.1: Variational workflow using multi-scale entanglement renormalization ansatz (MERA) tensor network

In this chapter, we will describe the circuit architecture of the model used in our studies, the image classification and quantum phase recognition tasks, their experimental setup in detail, various analytical studies and the results obtained by us.

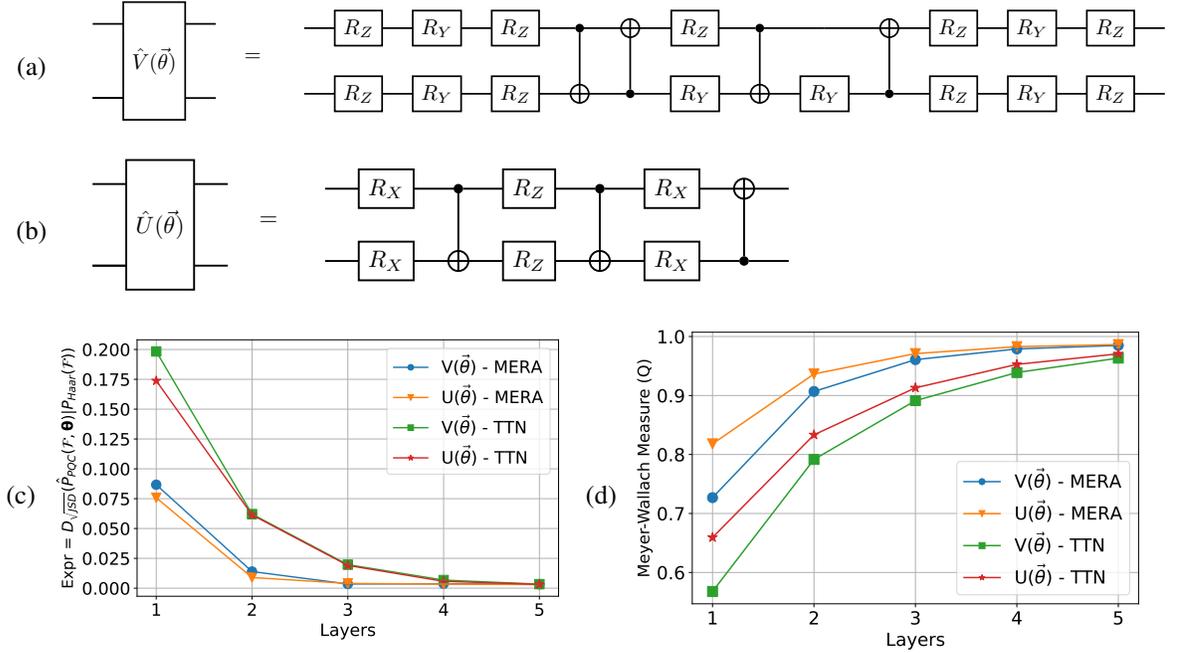


Figure 3.2: **Unitary blocks and their analysis:** The possible choices of unitary blocks for building variational ansätze are (a) $V(\vec{\theta})$, which can represent any element from $SU(4)$ group, and (b) $U(\vec{\theta})$, which is a two-qubit entangling unitary. For comparing the effectiveness of built TTN and MERA tensor network ansätze, we perform (a) expressibility analysis based on the Jensen-Shannon divergence of fidelity distributions of generated parameterized states with that of Haar states (lower the better), and (b) entangling power analysis based on the Meyer-Wallach measure (higher the better)

3.1 Circuit Architecture

Our experiments utilize variational quantum circuits based on the tree tensor network (TTN) [28], and the multi-scale entanglement renormalization ansatz (MERA) [29]. The TTN ansatz has a binary-tree-like structure with unitaries applied to the adjacent nodes, as shown in Fig. 3.3, which depends on the bond dimension D of the tensor network. As mentioned earlier, the bond dimension equals $D = 2^v$, where v is the number of qubits connecting the subtrees [52]. In our case, we have used $v = 1$; therefore, our ansatz has a bond dimension of two. On the other hand, the structure of the MERA tensor network can be explained using that of TTN itself, where it is constructed by adding a set of unitaries to consecutive nodes of the TTN as shown in Fig. 3.1.

The choice of the ansatz $\mathcal{U}(\vec{\theta})$ is a crucial one, which depends on the unitary block (U_B) and their bond dimension (D), and results in varied performance between different ansatz structures. In our case, we compare the performance of MERA- and TTN-based ansatz built using the unitary block $\hat{V}(\vec{\theta})$ and $\hat{U}(\vec{\theta})$ based on metrics of expressibility and entangling capability defined in the qLEET library [50]. In particular, we want our ansatz to be more expressive and capable of generating entanglement. For the former, we compare the divergence between the fidelity distributions for the states generated by Haar

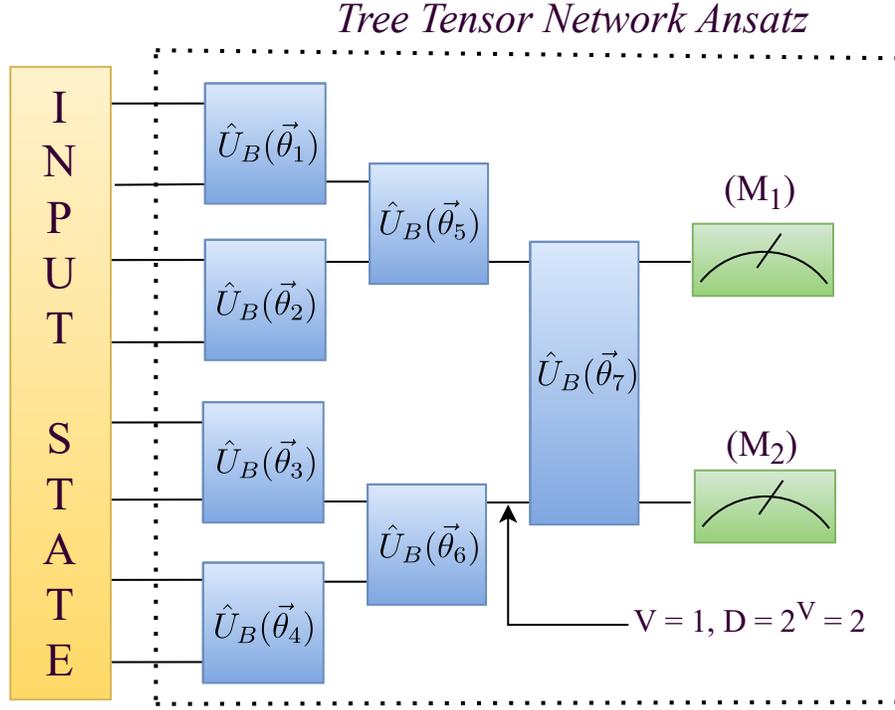


Figure 3.3: Structure of the tree tensor network (TTN) ansatz, with bond dimension $D = 2$, used in our experiments

Random unitaries ($P_{\text{Haar}}(\mathcal{F}, \mathcal{U}_{\text{Haar}})$) and the ansatz ($\hat{P}_{\text{PQC}}(\mathcal{F}, \mathcal{U}(\vec{\theta}))$) using the Jensen-Shannon distance (JSD) [53], where fidelity $\mathcal{F}(\psi_1, \psi_2) = |\langle \psi_1 | \psi_2 \rangle|^2$ is defined as the squared overlap between the states $|\psi_1\rangle, |\psi_2\rangle$ [54] produced by $U_1, U_2 \in \mathcal{U}(\vec{\theta})$ (or $\mathcal{U}_{\text{Haar}}$). We use this to define *expressivity* ($\text{Expr} \in [0, 1]$) of the ansätze as given below:

$$\text{Expr} = D_{\sqrt{\text{JSD}}}(\hat{P}_{\text{PQC}}(\mathcal{F}, \mathcal{U}(\vec{\theta})) | P_{\text{Haar}}(\mathcal{F}, \mathcal{U}_{\text{Haar}})), \quad (3.1)$$

The smaller this distance, i.e., divergence, the closer ansatz is to Haar random unitaries and hence more expressive it comes out to be. In contrast, to compare the latter, we use an entanglement measure known as the Mayer-Wallach measure (Q) [55], which quantifies the average entanglement in all the states produced by an ansatz by measuring the average linear entropy over all possible single-qubit subsystems (Eq. 3.2).

$$Q = \frac{2}{|\vec{\theta}|} \sum_{\theta_i \in \vec{\theta}} \left(1 - \frac{1}{n} \sum_{k=1}^n \text{Tr}(\rho_k^2(\theta_i)) \right), \quad 0 \leq Q \leq 1 \quad (3.2)$$

where n is the total number of qubits, $\rho(\vec{\theta}) = |\psi(\vec{\theta})\rangle \langle \psi(\vec{\theta})|$ is the density matrix for the parameterized pure state $|\psi(\theta)\rangle$ and $\rho_k(\vec{\theta})$ is the reduced single-qubit density matrix for the k^{th} qubit after tracing out the rest. For any given ansatz $\mathcal{U}(\vec{\theta})$, the larger the value of Q is, the more capable in general it would be to produce entanglement between qubits, i.e., more entangled states.

We present the structure description of $U(\vec{\theta})$ and $V(\vec{\theta})$ in Figs. 3.2a and 3.2b. The first one is a general element of the $SU(4)$ group, which can be decomposed into four controlled-NOTs and 15 single-qubits rotations [34]. In distinction, the other one is a two-qubit entangler gate comprising three controlled-NOTS and six single-qubit rotations arranged in a layer-wise manner.

In the Figs. 3.2c and 3.2d, we look at expressibility and entangling capability, respectively. As a general trend, we see that MERA-based ansatz is more expressible and generates more entangled states than TTN-based ansatz. Additionally, amongst $U(\vec{\theta})$ and $V(\vec{\theta})$, in both the cases, for single-layered circuits ($L = 1$), the ansatz built using $U(\vec{\theta})$ comes out to be more effective than $V(\vec{\theta})$. Moreover, since the number of variational parameters is lesser in case of the $U(\vec{\theta})$ block as compared to the $V(\vec{\theta})$ block, with a ratio of 2 : 5, it will be easier to optimize the former and hence it will be more scalable for larger systems. Additionally, the circuits with $U(\vec{\theta})$ and $V(\vec{\theta})$ blocks become equally expressible for both MERA- and TTN- based ansatz for multiple layers ($L > 1$). However, the $U(\vec{\theta})$ block produces more entangled states than the $V(\vec{\theta})$ block. Therefore, based on these observations, we have used tensor network ansatz based on the $U_B = U(\vec{\theta})$.

3.2 Image Classification

3.2.1 Dataset

We have conducted the image classification tasks on the Fashion-MNIST dataset [51]. It is a set of 28×28 grayscale images with 60,000 train and 10,000 test samples spread uniformly among 10 classes (tshirt, trousers, pullover, etc.) In our experiments, the train set was split in a ratio of 5 : 1 into balanced train and validation sets. Therefore, our data was split into the train, validation, and test classes in the ratio 5 : 1 : 1, with each class distributed uniformly in each of these sets.

Most of the image classification tasks done using quantum tensor networks use the MNIST dataset [56]. We have chosen the Fashion-MNIST dataset because it is less explored in the quantum machine learning literature and is more complicated than the MNIST dataset, which is essentially solved at this point [57]. Therefore obtaining better accuracies at Fashion-MNIST would represent the effectiveness of the learning models better.

3.2.2 Encoding Strategy

In order to process classical data using a quantum circuit, we first need to embed it in a quantum state. For our experiments, we have used amplitude embedding in which encoding an image of size $N \times M$ will require $\log_2(N \times M)$ qubits. Each image of size 28×28 is first converted to a linear vector of size 1×28^2 . In case the image size is too large to process, it is first resized and then transformed into the image vector. The image vector is then mapped to a state in the Hilbert Space. A variety of feature maps can be used for this purpose. So each image in the Fashion-MNIST dataset was first resized and

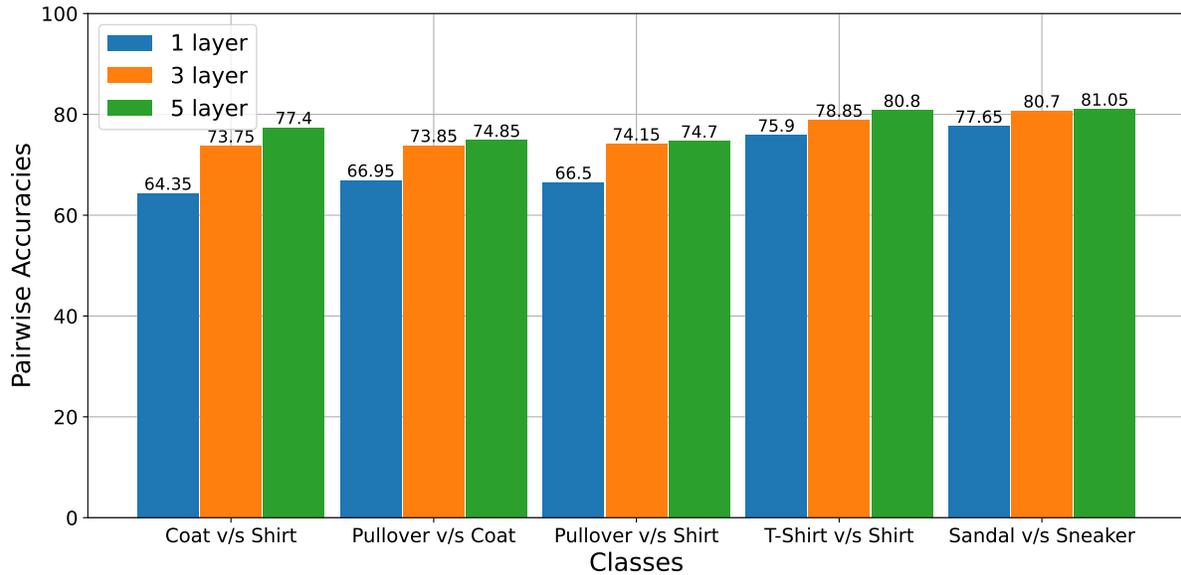


Figure 3.4: Performance of the MERA tensor network ansatz on classes of the Fashion-MNIST dataset with different one, three and five layers of the ansatz

converted to an image vector. The image vector was then normalized and encoded into the amplitudes of an eight qubit quantum state.

3.2.3 Optimization and Hyperparameters

The ADAM optimizer [58] was used to optimize the training process with a learning rate of 0.01. A mini-batch size of 20 was used, and the model was trained over 40 epochs to minimize the cross-entropy loss [59]. Computation of both the loss and pair-wise accuracies was done by using $\langle Z_3 \rangle$ obtained by computational basis measurement of the third qubit.

3.2.4 Results

We used eight qubit ansatz based on both TTN and MERA tensor network states. Amongst them, the latter obtained reasonably better results for all pairs of classes of the Fashion MNIST dataset and therefore, we present only its results here. Predictably, we see better performance on classes that are more unlike each other, like Pullover vs. Ankle boot (99.3%), than classes that are similar to each other, like coat and shirt (64.35%).

Increasing the number of layers to three and five has shown an increase in the pairwise accuracy, especially in our previous case of the coat and shirt labels where the accuracy increases from 64.35%

T-shirt	-									
Trouser	0.953	-								
Pullover	0.894	0.966	-							
Dress	0.8655	0.9155	0.9615	-						
Coat	0.8315	0.944	0.6695	0.8875	-					
Sandal	0.9085	0.977	0.9745	0.9795	0.8925	-				
Shirt	0.759	0.9405	0.665	0.8935	0.6435	0.9725	-			
Sneaker	0.988	0.992	0.9925	0.993	0.995	0.7765	0.994	-		
Bag	0.9155	0.9645	0.939	0.9445	0.8975	0.7895	0.9335	0.9215	-	
Ankle boot	0.9845	0.9815	0.993	0.98	0.98	0.79	0.9845	0.8925	0.9895	-
Layer: 1	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot

Table 3.1: Pairwise accuracy on the classes of the Fashion-MNIST dataset for one layer of the MERA tensor network

T-shirt	-									
Trouser	0.9585	-								
Pullover	0.9375	0.974	-							
Dress	0.8805	0.9445	0.9655	-						
Coat	0.8955	0.959	0.7385	0.891	-					
Sandal	0.982	0.986	0.989	0.9915	0.98	-				
Shirt	0.7875	0.963	0.7415	0.9005	0.7375	0.9825	-			
Sneaker	0.992	0.9945	0.9955	0.9965	0.998	0.807	0.9955	-		
Bag	0.968	0.9775	0.958	0.962	0.973	0.94	0.9475	0.964	-	
Ankle boot	0.989	0.986	0.998	0.9915	0.992	0.799	0.992	0.8975	0.9915	-
Layers: 3	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot

Table 3.2: Pairwise accuracy on the classes of the Fashion-MNIST dataset for three layers of the MERA tensor network

T-shirt	-									
Trouser	0.959	-								
Pullover	0.939	0.9755	-							
Dress	0.8895	0.9515	0.9655	-						
Coat	0.9285	0.964	0.7485	0.8965	-					
Sandal	0.99	0.9915	0.995	0.992	0.9935	-				
Shirt	0.788	0.964	0.747	0.905	0.774	0.9865	-			
Sneaker	0.9925	0.9975	0.999	0.9975	0.9985	0.8105	0.997	-		
Bag	0.9695	0.9775	0.963	0.969	0.974	0.9425	0.955	0.9805	-	
Ankle boot	0.99	0.9885	0.9985	0.9945	0.992	0.8315	0.9965	0.898	0.994	-
Layers: 5	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot

Table 3.3: Pairwise accuracy on the classes of the Fashion-MNIST dataset for five layers of the MERA tensor network

to 73.75% to 77.4% as can be seen in Tables 3.1, 3.2 and 3.3. Such a trend is observed in most classes where single-layered tensor network ansätze did not perform very well.

We observe a bigger difference in accuracy when going from one layer to three layers than when going from three layers to five layers. In fact, in most cases, we see a similar performance in ansätze with three and five layers. Fig. 3.4 shows the pairwise accuracy for certain pairs of classes as the number of layers increases, which is corroborated by the increasing trends of expressibility and entangling power in Figs. 3.2c and 3.2d.

3.3 Quantum Phase Recognition

3.3.1 Models and Data generation

3.3.1.1 1-D Transverse-field Ising Model

The transverse-field Ising model in one dimension is characterized by the following Hamiltonian:

$$\hat{H}(h) = J \sum_{i=1}^n \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z + h \sum_{i=1}^n \hat{\sigma}_i^x, \quad (3.3)$$

where J is the coupling constant, h the external magnetic field and $\hat{\sigma}_i^z$ and $\hat{\sigma}_i^x$ represents the Pauli matrices Z and X acting on the i^{th} spin. We have taken $J = 1$ in our experiments, so when $h < 1$, the nearest-neighbor term dominates. This leads to the spins aligning in either an up or down direction, resulting in a disordered paramagnetic phase. For $h > 1$, the second term dominates, and the spins end up aligning themselves with the external magnetic field leading to an ordered ferromagnetic phase. A phase transition for this system between these two phases occurs at $h = J$ [39].

In our experiments, we have generated 1000 ground states for linear chain systems with four and eight spins using the given Hamiltonian with $J = 1$ and h varying from 0 to $2J$.

3.3.1.2 1-D XXZ Heisenberg Model

The one-dimensional XXZ Heisenberg model is described using the following Hamiltonian:

$$\hat{H}(h) = J \left[\sum_{i=1}^n \hat{\sigma}_i^x \hat{\sigma}_{i+1}^x + \hat{\sigma}_i^y \hat{\sigma}_{i+1}^y + \Delta \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z \right] \quad (3.4)$$

Where J is again a coupling constant, taken as 1 in our experiments, and Δ introduces anisotropy in the interaction along the \hat{z} -direction. It is observed that for $\Delta \rightarrow \infty$, the system is in the antifer-

romagnetic/Néel state, i.e., all spins are alternating spin-up or spin-down. As $\Delta \rightarrow 1$, the spins begin to reorient themselves, and for $-1 < \Delta < 1$, they remain in the $\hat{x} - \hat{y}$ plane, putting the system in a paramagnetic phase [37]. Finally, for $\Delta < -1$, all the spins arrange themselves in the same direction resulting in a ferromagnetic phase. Therefore, for $J = 1$, phase transitions happen clearly at $\Delta = 1$ and $\Delta = -1$ [37, 60].

The 1-D XXZ Heisenberg Model data was generated for linear chain systems with four spins (1×4) and eight spins (1×8) using the given Hamiltonian with $J = 1$ and Δ varying from -2 to 2 for 1000 points.

3.3.1.3 2-D Transverse-field Ising Model

The two-dimensional transverse-field Ising model has the same Hamiltonian as the one-dimensional case. However, a phase transition is observed at $h \approx 3.01J$ [61]. The data is generated for 1000 points for 2-D lattices with four spins (2×2) and eight spins (2×4) using the given Hamiltonian with $J = 1$ and h varying from 0 to $6J$. The phase transition is seen at $h = 3.01J$. Fig. 2.18 shows a 2-D lattice (2×4) of spin systems.

3.3.1.4 2-D XXZ Heisenberg Model

The two-dimensional XXZ Heisenberg model has the same Hamiltonian as the one-dimensional case with a phase transition occurring at $\Delta = 1$ and -1 as well [62]. The data generation process remains the same as in the one-dimensional case.

3.3.2 Training

A quantum circuit was trained using the variational quantum algorithm as shown in Fig 3.1. The data was split randomly into the train, validation, and test sets in the ratio $3 : 1 : 1$, with each set having a balanced distribution of the classes. For all the experiments, measurements were taken on two readout qubits (q_i, q_j). These were the second and third qubits for the four-spin systems and the third and sixth qubits for the eight-spin systems. For both Ising and Heisenberg models, we calculate expectation values $\langle Z_i \rangle$, $\langle Z_j \rangle$ and $\langle Z_i Z_j \rangle$ to compute probabilities of the elements of computational basis corresponding to each class (phase) inspired by the amplitude decoding method introduced in [34]. These probabilities were fed to a softmax function for normalization, whose outputs were used to calculate the cross-entropy loss [59]. A batch size of 8 was used with a learning rate of 0.002 for

Spin Models	Lattice	Tensor Network States	Test Accuracies		
			8 spins (simulator)	4 spins (simulator)	4 spins (IBMQ Nairobi)
XXZ-HM	1-D	MERA	98.6 ± 0.70	98.6 ± 0.38	74.0
XXZ-HM	1-D	TTN	96.5 ± 1.03	98.5 ± 0.32	72.6
TFIM	1-D	MERA	99.8 ± 0.06	98.6 ± 0.08	84.5
TFIM	1-D	TTN	98.3 ± 0.10	99.0 ± 0.03	86.2
XXZ-HM	2-D	MERA	98.5 ± 0.88	98.4 ± 0.27	68.6
XXZ-HM	2-D	TTN	96.3 ± 1.25	98.1 ± 0.31	64.0
TFIM	2-D	MERA	99.8 ± 0.06	98.8 ± 0.18	72.1
TFIM	2-D	TTN	98.0 ± 0.09	99.1 ± 0.13	73.6

Table 3.4: Performance of the TTN and MERA tensor networks on recognizing correct phases of various XXZ Heisenberg (XXZ-HM) and transverse-field Ising (TFIM) spin systems on one-dimensional (linear) and two-dimensional (rectangular) lattices. For eight spin systems, simulations were performed numerically on a quantum simulator, and results were averaged over five trials. Whereas for the four spins systems, along with similar numerical simulations, experiments were also executed on the IBMQ Nairobi (*ibmq_nairobi*) [1], a seven-qubit quantum hardware, and the best results out of three trials are being reported here.

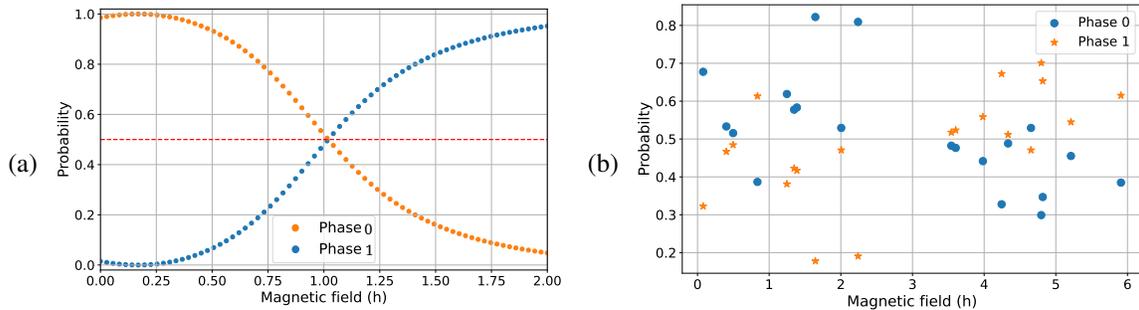


Figure 3.5: **Prediction probabilities of phases with MERA based ansatz:** (a) for transverse-field Ising model in 1-D case (noiseless simulation), and (b) For transverse-field Ising model in 2-D case (executed on IBMQ Nairobi (*ibmq_nairobi*), 7-qubit hardware [1])

the MERA-based ansatz and 0.0008 for the TTN-based ansatz. The ADAM optimizer [58] was used to optimize the training process over 2000 iterations.

3.3.3 Results

We see that both MERA- and TTN-based ansätze perform well for both four-spin and eight-spin systems, with overall better performance for the transverse-field Ising models than the Heisenberg models (Table 3.4). Moreover, their performance in the 1-D cases is better than in the more complicated 2-D case. In Fig. 3.5a, we show the results outputted by our model for the transverse-field Ising model with eight spins on a linear (chain) lattice. When the probability of a phase is more than 50%, we assign our output the label corresponding to that phase. We see that our model is more confident when the value of h is further from the point of phase transition, i.e., when h is 1. Similarly, Fig. 3.5b shows the values of our model’s outputs when the models were trained and executed on the IBMQ Nairobi (*ibmq_nairobi*) [1], which is a 7-qubit superconducting quantum hardware for the transverse-field Ising model with four spins on a square (2×2) lattice.

Chapter 4

Conclusions

In this thesis, we have studied the performance of quantum tensor networks for image classification tasks and quantum phase recognition tasks of spin systems.

We have extended the previous works done in this domain in the following two ways. First, we have presented a strategy based on metrics like expressibility and entangling capability of the parameterized circuits to choose a well-suited block structure for the tensor-network inspired ansätze. Such analysis was corroborated by the results obtained for the image classification task, where we were able to increase the performance of our classifiers by increasing the number of layers of the circuits. Second, for the quantum phase recognition task, we have attempted to study spin systems on 2-D lattices with the tensor-network ansatz, which are generally more challenging than those on 1-D spin lattices that have been studied in the literature until now [34, 36].

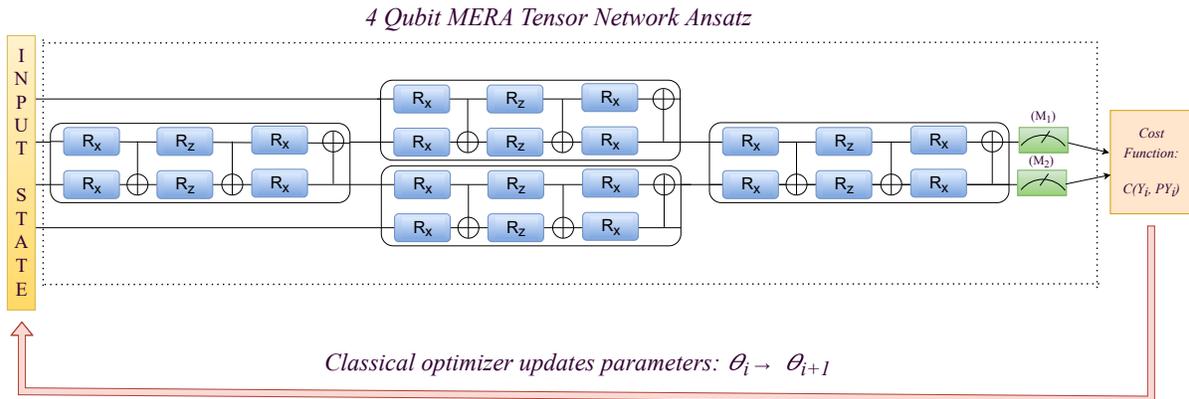


Figure 4.1: 4 qubit MERA tensor network ansatz used for phase recognition

In the image classification task, the pairwise accuracies between the different classes of the Fashion-MNIST dataset were calculated for 1, 3, and 5 layers of the MERA tensor network ansatz. The results

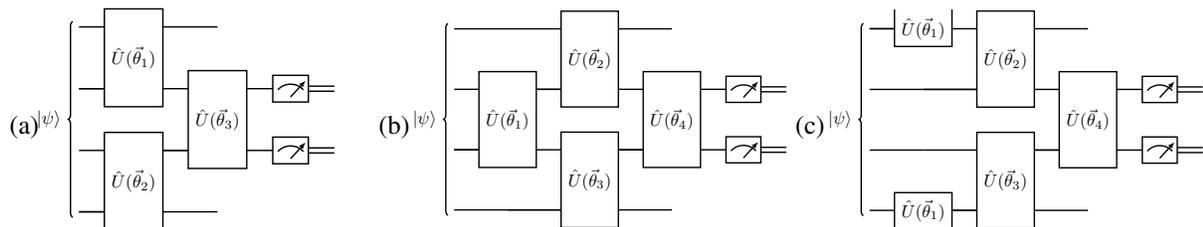


Figure 4.2: **Tensor network ansätze for four-spin systems:** Structures of variational ansätze based on the (a) tree tensor network (TTN) and the (b) multi-scale entanglement renormalization ansatz (MERA) tensor network. (c) Modified structure of MERA tensor network ansatz with changed first unitary block

are shown in Tables 3.1, 3.2 and 3.3. We see a clear increase in accuracy when the number of layers is increased, especially when we go from a single layer to three layers. The performance of the ansatz with five layers is slightly more than when three layers are used. This is corroborated by Fig. 3.2c, where we see a marked increase in the expressibility of our circuit when the number of layers is increased from one to three but not a lot of increase when going from three layers to five. In pairs of classes where one layer of the ansatz performed poorly, like coat vs. shirt or sandals vs. sneakers, we see an appreciable increase in accuracy with an increase in layers. Possibly, this happens because the layered structure allows correlation to be distributed more effectively among the qubits allowing the system to evolve to states that were not previously possible. More explicitly, the ansatz becomes more expressible with each layer, and its overall entangling power also gets enhanced. This can be easily seen in the results of the entangling power analysis as shown in Fig. 3.2d, where entanglement measures for both TTN and MERA follow a similar exponential trend of improvement with each layer before plateauing down.

In the quantum phase recognition tasks, we first use a VQE-based variational routine with a hardware-efficient ansatz to prepare these systems in the ground state of the Hamiltonian for each spin system instance. This enables us to take care of the sign problem by employing the hybrid quantum-classical routine. Furthermore, the TTN and MERA tensor network ansatz results indicate their effectiveness at solving many-body physics problems. We see that it was much easier for tensor-network-based ansätze to classify phases for the transverse-field Ising model, which has simpler interaction terms than the XXZ Heisenberg models. This is in agreement with the previous results obtained for these two models [36]. Moreover, for the both models, the results for systems on one-dimensional linear lattices were better than the results for two-dimensional rectangular lattices. This is again due to fewer interacting terms, as seen in the previous observation. Among the two tensor-network-based ansätze, we find MERA-based ansatz to be overall superior in performance for such tasks than the TTN one, except for the case with the

Ising model with four spins. While we can still attribute the MERA-based ansatz's better performance to it being more expressible and generating more entanglement in the states it evolves, the exception tells us that the order in which the correlation gets distributed also matters, especially when the circuit is shallow. In this particular case, it appears from the ansätze structures presented in Fig. 4.2 that the correlation involving the first and fourth qubits need to be spread before the second and third qubits are entangled. Modifying the MERA ansatz structure by changing the first unitary block to act on the first and fourth qubits as shown in Fig. 4.2c results in a significant improvement in performance to 99.2 ± 0.05 and 99.3 ± 0.09 for 1-D and 2-D Ising models, respectively. Finally, we also executed our classifiers on the actual quantum hardware, IBMQ Nairobi, for classifying phases of four spin systems for both 1-D and 2-D cases. We see that even though there's a decreased performance due to the noise present on the device, it was still able to classify the phases decently (Fig. 3.5b). We speculate that this performance can be further improved by employing specific error mitigation techniques like those available in Mitiq [63].

Overall, our studies have shown promising results in both tasks, and we conclude that tensor-network-inspired ansatz is an ideal candidate for quantum-enhanced learning of both quantum and classical data. For quantum data, further studies need to be done on tasks such as the phase recognition task on larger, more complicated systems, like systems with 16 or 24 spins, to see how scalable our current model is, which is something we are currently pursuing. On the other hand, for the classical data, more specifically, for the image classification tasks, more work is required to study higher resolution images that would require much better encoding strategies, which is another area of our interest.

Bibliography

- [1] IBM Quantum, <https://quantum-computing.ibm.com/> (2021).
- [2] D. Padua, A. Ghoting, J. A. Gunnels, M. S. Squillante, J. Meseguer, J. H. Cownie, D. Roweth, S. V. Adve, H. J. Boehm, S. A. McKee, R. W. Wisniewski, G. Karypis, A. D. Malony, S. Gottlieb, R. Riesen, A. B. Maccabe, G. Bilardi, A. Pietracaprina, A. Kejariwal, A. Nicolau, C. Lengauer, J. L. Gustafson, W. Gropp, J.-P. Prost, D. Padua, G. Lowney, P. Amestoy, A. Buttari, I. Duff, A. Guermouche, J.-Y. L'Excellent, B. Uçar, R. H. Halstead, M. Nemirovsky, P. Amestoy, A. Buttari, I. Duff, A. Guermouche, J.-Y. L'Excellent, B. Uçar, and S. Pakin, *Encyclopedia of Parallel Computing* (Springer US, 2011) pp. 1177–1184.
- [3] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by Turing machines*, *Journal of Statistical Physics* **22**, 563 (1980).
- [4] R. P. Feynman, *Simulating physics with computers*, *International Journal of Theoretical Physics* **21**, 467 (1982).
- [5] P. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124 (1994).
- [6] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, *Communications of the ACM* **21**, 120 (1978).
- [7] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, and et al., *Quantum supremacy using a programmable superconducting processor*, *Nature* **574**, 505 (2019).
- [8] I. L. Chuang, N. Gershenfeld, and M. Kubinec, *Experimental implementation of fast quantum searching*, *Physical Review Letters* **80**, 3408 (1998).

- [9] D. F. Flother, C. Moose, D. I. Tavernelli, H. Fraser, and V. Pureswaran, *Exploring quantum computing use cases for Life Sciences* (2019), available at <https://www.ibm.com/thought-leadership/institute-business-value/report/quantum-life-sciences#>.
- [10] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, *A variational eigenvalue solver on a photonic quantum processor*, *Nature Communications* **5**, [10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213) (2014).
- [11] K. M. Nakanishi, K. Mitarai, and K. Fujii, *Subspace-search variational quantum eigensolver for excited states*, *Physical Review Research* **1**, [10.1103/physrevresearch.1.033062](https://doi.org/10.1103/physrevresearch.1.033062) (2019).
- [12] H.-H. Lu, N. Klco, J. M. Lukens, T. D. Morris, A. Bansal, A. Ekstrom, G. Hagen, T. Papenbrock, A. M. Weiner, M. J. Savage, and P. Lougovski, *Subatomic many-body physics simulations on a quantum frequency processor*, *Conference on Lasers and Electro-Optics* [10.1364/cleo_qels.2019.fth3a.6](https://doi.org/10.1364/cleo_qels.2019.fth3a.6) (2019).
- [13] A. Robert, P. K. Barkoutsos, S. Woerner, and I. Tavernelli, *Resource-efficient quantum algorithm for protein folding*, *npj Quantum Information* **7**, [10.1038/s41534-021-00368-4](https://doi.org/10.1038/s41534-021-00368-4) (2021).
- [14] Y. Cao, J. Romero, and A. Aspuru-Guzik, *Potential of quantum computing for Drug Discovery*, *IBM Journal of Research and Development* **62**, [10.1147/jrd.2018.2888987](https://doi.org/10.1147/jrd.2018.2888987) (2018).
- [15] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2012).
- [16] A. Pathak, *Elements of Quantum Computation and Quantum Communication*, 1st ed. (Taylor & Francis, Inc., USA, 2013).
- [17] E. Munro, *Quantum computing: Near- and far-term opportunities* (2018), available at https://medium.com/@quantum_wa/quantum-computing-near-and-far-term-opportunities-f8ffa83cc0c9.
- [18] J. Preskill, *Quantum Computing in the NISQ era and beyond*, *Quantum* **2**, 79 (2018).

- [19] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, *Variational Quantum Algorithms*, arXiv e-prints (2020), [arXiv:2012.09265 \[quant-ph\]](https://arxiv.org/abs/2012.09265) .
- [20] A. Macaluso, L. Clissa, S. Lodi, and C. Sartori, in *Lecture Notes in Computer Science* (Springer International Publishing, 2020) pp. 591–604.
- [21] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, Equivalence of quantum barren plateaus to cost concentration and narrow gorges (2021), [arXiv:2104.05868 \[quant-ph\]](https://arxiv.org/abs/2104.05868) .
- [22] C. Poling, [Solving 'barren plateaus' is the key to Quantum Machine Learning](https://www.hpcwire.com/2021/03/24/lanl-researchers-tackle-the-barren-plateau-in-quantum-computing/) (2021), available at <https://www.hpcwire.com/2021/03/24/lanl-researchers-tackle-the-barren-plateau-in-quantum-computing/>.
- [23] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, *Cost function dependent barren plateaus in shallow parametrized quantum circuits*, Nature Communications **12**, [10.1038/s41467-021-21728-w](https://doi.org/10.1038/s41467-021-21728-w) (2021).
- [24] E. Stoudenmire and D. J. Schwab, in *Advances in Neural Information Processing Systems*, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016).
- [25] J. C. Bridgeman and C. T. Chubb, *Hand-waving and interpretive dance: an introductory course on tensor networks*, *Journal of Physics A: Mathematical and Theoretical* **50**, 223001 (2017).
- [26] E. M. Stoudenmire, *The Tensor Network* (2017), available at <https://tensornetwork.org/diagrams/>.
- [27] J. Y. Araz and M. Spannowsky, *Classical versus Quantum: comparing Tensor Network-based Quantum Circuits on LHC data* [10.48550/ARXIV.2202.10471](https://arxiv.org/abs/2202.10471) (2022).
- [28] Y.-Y. Shi, L.-M. Duan, and G. Vidal, *Classical simulation of quantum many-body systems with a tree tensor network*, Physical Review A **74**, [10.1103/physreva.74.022320](https://doi.org/10.1103/physreva.74.022320) (2006).
- [29] G. Vidal, *Class of Quantum Many-Body States That Can Be Efficiently Simulated*, *Phys. Rev. Lett.* **101**, 110501 (2008).

- [30] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, *Towards quantum machine learning with tensor networks*, *Quantum Science and Technology* **4**, 024001 (2019).
- [31] R. Haghshenas, J. Gray, A. C. Potter, and G. K.-L. Chan, *Variational Power of Quantum Circuit Tensor Networks*, *Physical Review X* **12**, [10.1103/physrevx.12.011047](https://doi.org/10.1103/physrevx.12.011047) (2022).
- [32] M. L. Wall, M. R. Abernathy, and G. Quiroz, *Generative machine learning with tensor networks: Benchmarks on near-term quantum computers*, *Physical Review Research* **3**, [10.1103/physrevresearch.3.023010](https://doi.org/10.1103/physrevresearch.3.023010) (2021).
- [33] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, *Hierarchical quantum classifiers*, *npj Quantum Information* **4**, [10.1038/s41534-018-0116-9](https://doi.org/10.1038/s41534-018-0116-9) (2018).
- [34] M. Lazzarin, D. E. Galli, and E. Prati, *Multi-class quantum classifiers with tensor network circuits for quantum phase recognition*, *Physics Letters A* **434**, 128056 (2022).
- [35] S. Yen-Chi Chen, C.-M. Huang, C.-W. Hsing, and Y.-J. Kao, *Hybrid quantum-classical classifier based on tensor network and variational quantum circuit*, arXiv e-prints (2020), [arXiv:2011.14651](https://arxiv.org/abs/2011.14651) .
- [36] A. V. Uvarov, A. S. Kardashin, and J. D. Biamonte, *Machine learning phase transitions with a quantum processor*, *Phys. Rev. A* **102**, 012415 (2020).
- [37] J. Parkinson and D. J. J. Farnell, *An Introduction to Quantum Spin Systems* (Springer Berlin Heidelberg, 2010).
- [38] V. Ohanyan, *Introduction to quantum spin chains, lecture notes* (2015), available at <http://training.hepi.tsu.ge/rtn/activities/sources/Ohanyan.pdf>.
- [39] A. Tan, *Quantum Ising Models* (2018), available at https://paramekanti.weebly.com/uploads/1/1/2/8/11287579/atan_paper.pdf.
- [40] L. Deng and X. Li, *Machine Learning Paradigms for Speech Recognition: An Overview*, *IEEE Transactions on Audio, Speech, and Language Processing* **21**, 1060 (2013).

- [41] A. Popli, S. Tandon, J. J. Engelsma, N. Onoe, A. Okubo, and A. Namboodiri, *A Unified Model for Fingerprint Authentication and Presentation Attack Detection* [10.48550/ARXIV.2104.03255](https://arxiv.org/abs/10.48550/ARXIV.2104.03255) (2021).
- [42] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, *High-Performance Neural Networks for Visual Object Classification*, arXiv e-prints (2011), [arXiv:1102.0183](https://arxiv.org/abs/1102.0183) [cs.AI] .
- [43] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensari, *Breast cancer classification using machine learning, 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)* [10.1109/ebbt.2018.8391453](https://arxiv.org/abs/10.1109/ebbt.2018.8391453) (2018).
- [44] J. Carrasquilla and R. G. Melko, *Machine learning phases of matter*, *Nature Physics* **13**, 431 (2017).
- [45] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Learning phase transitions by confusion*, *Nature Physics* **13**, 435 (2017).
- [46] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, *Sign problem in the numerical simulation of many-electron systems*, *Phys. Rev. B* **41**, 9301 (1990).
- [47] E. Farhi and H. Neven, *Classification with Quantum Neural Networks on Near Term Processors* [10.48550/ARXIV.1802.06002](https://arxiv.org/abs/10.48550/ARXIV.1802.06002) (2018).
- [48] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, *Variational Quantum Linear Solver*, arXiv e-prints (2019), [arXiv:1909.05820](https://arxiv.org/abs/1909.05820) [quant-ph] .
- [49] U. Azad and H. Singh, *Quantum chemistry calculations using energy derivatives on quantum computers*, *Chemical Physics* **558**, 111506 (2022).
- [50] U. Azad and A. Sinha, *qLEET: Visualizing Loss Landscapes, Expressibility, Entangling Power and Training Trajectories for Parameterized Quantum Circuits*, arXiv e-prints (2022), [arXiv:2205.02095](https://arxiv.org/abs/2205.02095) [quant-ph] .
- [51] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms* [10.48550/ARXIV.1708.07747](https://arxiv.org/abs/10.48550/ARXIV.1708.07747) (2017).

- [52] A. Bernardi, C. D. Lazzari, and F. Gesmundo, *Dimension of tensor network varieties*, Communications in Contemporary Mathematics [10.1142/s0219199722500596](https://doi.org/10.1142/s0219199722500596) (2022).
- [53] B. Fuglede and F. Topsoe, in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.* (IEEE, 2004).
- [54] R. Jozsa, *Fidelity for Mixed Quantum States*, Journal of Modern Optics **41**, 2315 (1994).
- [55] D. A. Meyer and N. R. Wallach, *Global entanglement in multiparticle systems*, Journal of Mathematical Physics **43**, 4273 (2002).
- [56] Y. LeCun and C. Cortes, *MNIST handwritten digit database* (2012), available at <http://yann.lecun.com/exdb/mnist/>.
- [57] Z. SE, *Fashion MNIST Benchmark* (2017), available at <http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/>.
- [58] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv e-prints , arXiv:1412.6980 (2014), arXiv:1412.6980 [cs.LG] .
- [59] Z. Zhang and M. R. Sabuncu, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18 (Curran Associates Inc., Red Hook, NY, USA, 2018) pp. 8792–8802.
- [60] F. Franchini, *An Introduction to Integrable Techniques for One-Dimensional Quantum Systems* (Springer International Publishing, 2017).
- [61] T. Hashizume, I. P. McCulloch, and J. C. Halimeh, *Dynamical phase transitions in the two-dimensional transverse-field Ising model*, Physical Review Research **4**, [10.1103/physrevresearch.4.013250](https://doi.org/10.1103/physrevresearch.4.013250) (2022).
- [62] O. S. Saryyer, *Two-dimensional quantum-spin-1/2 XXZ magnet in zero magnetic field: Global thermodynamics from renormalisation group theory*, Philosophical Magazine **99**, 1787 (2019).
- [63] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, P. Czarnik, M. El Mandouh, M. H. Gordon, Y. Hindy, A. Robertson, P. Thakre, N. Shammah, and W. J. Zeng, *Mitiq: A software package for error mitigation on noisy quantum computers*, arXiv e-prints , arXiv:2009.04417 (2020), arXiv:2009.04417 [quant-ph] .