CLASSICAL AND QUANTUM OPTIMIZATION FOR SCIENTIFIC COMPUTATION

by

Shree Hari Sureshbabu

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Sabre Kais (Co-Chair)

Department of Chemistry

Dr. Zubin Jacob (Co-Chair)

Elmore Family School of Electrical and Computer Engineering

Dr. Vaneet Aggarwal

School of Industrial Engineering

Dr. Hadiseh Alaeian

Elmore Family School of Electrical and Computer Engineering

Approved by:

Dr. Milind Kulkarni

To my brother, Aravind

ACKNOWLEDGMENTS

This dissertation would not have been possible without the support and guidance of many. Firstly, I would like to thank my advisors, Prof. Sabre Kais and Prof. Zubin Jacob, who have shaped me into the researcher I am today. Their constant encouragement has pushed me to tackle problems beyond my comfort zone and not be afraid to challenge myself consistently. Their drive and passion for science have been a huge inspiration.

I am grateful to Prof. Hadiseh Alaein for being an amazing mentor and friend. All the discussions were undoubtedly fun, interesting, and have inspired me in so many ways. I would also like to thank Prof. Supriyo Datta for some of the wonderful scientific conversations I have ever had. His approach to looking at the big picture and simplifying complicated problems to provide useful solutions has definitely carved my thought process.

I am indebted to Dr. Manas Sajjan and Dr. Fanglin Bao for teaching me fundamental concepts and clearing any doubts of mine at any point in time. Their patience and motivation have certainly been a few of the key ingredients in the completion of this thesis.

I was fortunate to have received an opportunity to intern at the Global Technology Applied Research (GTAR) Center at JPMorgan Chase. I owe a huge thanks to Dr. Marco Pistoia for his constant backing and valuable guidance. I would also like to thank the entire GTAR team and in particular Dylan Herman, Dr. Ruslan Shaydulin, Dr. Shouvanik Chakrabarti, Dr. Yue Sun, Dr. Shaohan Hu, and Dr. Omar Amer for their support and delightful discussions.

I thank my brother, my parents, and Shreeya for their unconditional support and encouragement. It is with utmost certainty I can state that without them, I wouldn't be who I am today and I am forever grateful for everything they have done for me.

I would also like to express my sincere gratitude to every member of Prof. Kais' lab and Prof. Jacob's lab. I will always cherish all the conversations especially those over coffee, lunch, and dinner. I have always learned something new as an outcome of these memorable chats.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

B1 The lower bound for probability of successfully collapsing the ancilla register in state $|\vec{1_a}\rangle$ as deduced in Eq. B25 (R.H.S of Eq. B25) is plotted in red and the actual probability of such an event (Eq. B22 or L.H.S of Eq. B25) is plotted in green as a function of training iteration. We see that the green curve is always slightly above the red one in accordance with Eq. B25 deduced above. The simulation is from the **'RBM-qasm'** variant for conduction band (CB) of MoS_2 at $(k_x, k_y) = K$ symmetry point which is where the direct band-gap is lowest. The simulation is performed by warm starting with initial parameter set from a converged run at a nearby k-point to ensure faster and accurate convergence. With the said warm start the desired accuracy of $\leq 10^{-4}$ eV in energy error was reached within 200 iterations. The high value of the probability of the successful event (and the associated lower bound) as seen from the y-scale is problem-specific as it entirely depends on $(\vec{a}, \vec{b}, \vec{W}, k)$ (see Eq. B22 and Eq. B25) . For these systems, even with a moderate k parameter i.e. $k \leq 1.5$ for all iterations, the remaining set $(\vec{a}, \vec{b}, \vec{W})$ are such that a high value of $P_{success}$ as seen on the y-scale is attained. The parameter set $(\vec{a}, \vec{b}, \vec{W})$ depends on the updates from the cost-function and hence on the Hamiltonian of the system being treated. For this choice of k parameter, the specific values of the y-scale in the plot is thus characteristic of the systems being studied in this report and may be different for other systems. However the lower bound deduced in Eq. B25 is mathematically generic and should be valid for any arbitrary system and a given k parameter, even though the specific value it acquires during training may vary 187

B2 The energy error (eV) for 4 arbitrarily chosen (k_x, k_y) points within the Brillouin zone in the $\Gamma - K - M - \Gamma$ path of the conduction band (CB) of MoS_2 after training the network using the **'RBM-IBMQ'** variant with and without Measurement Error Mitigation (MEM). Each of the 4 points is marked on the x-axis as (1),(2),(3),(4). We also plot the energy error as a function of the training epoch/iteration with and without MEM in (b) for point index (1) in (c) for point index (2) (d) for point index (3) (e) for point index (4). We see from (a) that the results with MEM are of higher accuracy for all 4 points than without MEM. However by far the greatest impact which MEM has on the results is on improving self-convergence. This is best seen from (b)-(d). The error bars in (a) on the points without MEM are to highlight the statistical uncertainty due to time averaging from this poor self-convergence. Each such bar designates the sample standard deviation of the last 30 points (marked in (b)-(d) with a vertical dashed line) in the training process whereas the orange circles in (a) are the corresponding sample mean. Each calculation with and without MEM is done using a single run on IBM-Sydney and followed till 150 iterations. All simulations are performed by warm starting with initial parameter set from a converged run at a nearby k-point in **'RBM-qasm'** variant. This is done so that same initial parameter set is used for simulations with and without MEM which eliminates biases due to random initial parameterization and affords a strictly fair comparison. The 4 points chosen are not the symmetry points as we have seen that symmetry points usually converges better regardless 194

ABSTRACT

Optimization and Machine learning (ML) have emerged as two positively disruptive methodologies and have thus resulted in unprecedented applications in several domains of technology. In recent years, ML has forayed into physical sciences and provided promising outcomes thanks to its ability in representing and generalizing complex functions to reveal underlying relations among variables describing a system. By casting ML as an optimization task, we first focus on its application in solving quantum many-body problems. Leveraging the power of quantum computation, we develop hybrid quantum machine learning protocols and implement benchmark tests to calculate the band structures of two-dimensional materials. We also show how this method can be used to estimate the critical point for a quantum phase transition. One hurdle in such techniques is related to parameter optimization, wherein to obtain the desired result, the parameters have to be optimized, which can be computationally intensive. For a particular class of problem and a choice of algorithm, we deduce a simple parameter setting rule. This rule is projected as a heuristic and is validated numerically for several problem instances. Finally, by venturing into thermal photonics, a framework that takes advantage of the spectral and spatial information of hyperspectral thermal images to establish a completely passive machine perception, titled HADAR is presented. A conventional deep neural network is developed that utilizes the governing equation of HADAR and its performance in semantic segmentation is demonstrated. Altogether, this report establishes the need for creative algorithms that exploit modern hardware to solve complex problems that were previously deemed unsolvable.

1. INTRODUCTION

As electronic computation developed through the advent of a transistor, the domain of algorithmic research also expanded to identifying and solving complex problems. Computation provided a way to go beyond both the mathematical limitation set forth for analytic exploration and also the engineering limitation for experimental investigations. Taking advantage of classical computers to solve problems in physical sciences involved using Monte Carlo methods for problems in neutron diffusion [1], statistical mechanics [2], and eventually understanding molecular dynamics [3], [4].

To motivate the need for numerical simulation, consider the problem of electronic structure. For a pure-state in a quantum-mechanical system, every statistical description, like the probabilistic outcomes of a measurement of any property, the post-measurement state, and the expectation value of the property associated with several observations are all encoded in the wave-vector of the system $|\psi\rangle$. This fundamental entity, which contains and directs information in the quantum state, is the wavefunction ψ. To solve a quantum many-body problem, an exponential amount of information is required to get a complete description of ψ. In order to obtain the solution to this fundamentally many-body bottleneck that scales exponentially as the system size, resorting to computation was the best way forward. From a finite set of one-electron orbitals, approximation of the many-electron wavefunction as a linear combination of antisymmetrized tensor products (Slater determinants) were proposed, to solve the electronic structure problem, wherein these methods formulate an *ansatz* for an eigenstate. Then, the expectation values of observables and correlation functions with regard to that wavefunction are calculated. For molecular systems of size N, the techniques arranged in the increasing order of scaling are described in Table.1.1.

(CCSD(T)) is deemed the gold standard. For solid-state systems, mean-field treatment based on density, DFT, became quite popular as this method was less expensive than the wavefunction methods. However, to solve systems that are strongly correlated and require a larger basis set description, a compact representation of the wavefunction is required.

Solving an inverse problem possesses wide applications in several branches of science and engineering as it provides valuable information about the internal parameters of the system,

Table 1.1. Scaling of certain quantum chemistry techniques as a function of system size N.

Technique	Scaling
Hartree-Fock (HF) [5], [6]	$N^3 \sim N^4$
Coupled-Cluster with Singles and Doubles (CCSD) [7], [8]	N^6
Coupled-Cluster with Singles, Doubles, and perturbative estimate to the connected Triples (CCSD(T)) [9]	N^7
Density Functional Theory (DFT) [10]	N^3

which may or may not be measurable. A method that can learn the measurable entities, having defined a governing equation to inform about the internal structure, can significantly outperform traditional methods in terms of reconstruction accuracy and speed, especially in scenarios where the availability of data is limited.

In both the problems described above, a technique that acts as a universal approximater and contains attributes of dimensionality reduction and feature extraction is required. *Machine Learning* (ML) is credited for possessing these exact qualities and has thus yielded unprecedented applications in several domains of technology. Along with the data revolution, the usage of the *Graphical Processing Units*(GPUs) has steered remarkable progress in the study of ML and is becoming increasingly popular in several domains of physical sciences like astronomy [11], [12], material science [13], [14], cheminformatics [15].

Also, providing solutions to these quantum many-body problems necessitated the usage of parallel processing systems and methods to use these systems. For decades, detailed atomistic simulations were performed on supercomputing clusters. However, all these calculations were done on conventional computers. To deal with such problems that are fundamentally quantum in nature, Richard Feynman remarked that the computation must also be quantum mechanical in nature. If that could be realized, then a problem that grows exponentially with the problem size would scale linearly using quantum resources. This sparked an inspiration

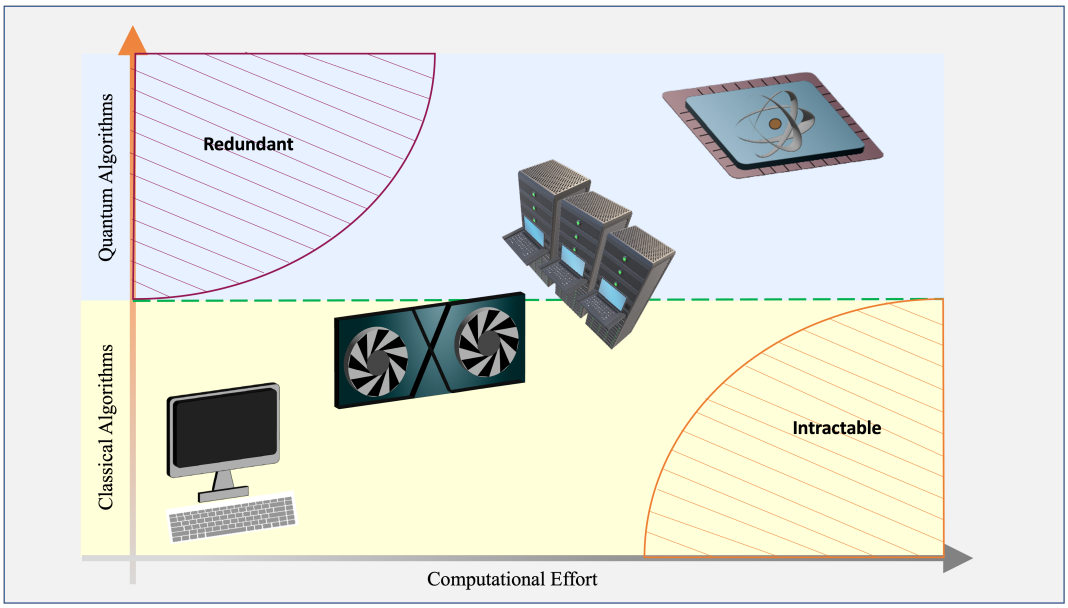

Figure 1.1. As computational effort required to solve a problem grows, the necessity to utilize varied computational resource also extends. Previously, solving highly complex scientific problem were significantly approximated and solved using classical algorithms on classical systems. With the need to extend the system size, the usage of supercomputers also grew due to their parallel processing capabilities. Finally, with the development of quantum hardware and the availability of powerful classical devices, computing solutions to previously intractable problems appears to be a possibility.

to Mathematicians and computer scientists to develop such methods, which eventually led to algorithms such as Shor's factoring algorithm [16] and Grover's search algorithm'[17].

With the growing computational effort required to solve complex problems, the need for efficient computing resources and ingenuous algorithmic advances becomes apparent. The first part of the report deals with encoding a *Neural Network* (NN) as a wavefunction ansatz and devising algorithms that can be executed on a quantum computer to solve the described quantum many-body problems. In the second part, taking inspiration from the success of ML in solving inverse problems, NN based method to decompose hyperspectral thermal images is developed. In essence, we propose algorithms that utilize both classical and quantum information to help our quest in solving complex physio-chemical problems.

1.1 Paradigms of Machine Learning

In general, neural networks are function approximators that are trained to optimize a specific cost function of interest over the dataset. The cost function is chosen so as to maximize an outcome from the output of the neural network. The process of optimizing the cost function is referred to as the 'learning' of the network. Problems solved by ML usually appear in the following categories. Real-life complex problems usually involve a combination of techniques that are used to address each of them independently [18].

- **Supervised Learning:** Given a set of N labeled training examples that can be represented as $D = \{(x_N, y_N)\}$ with x_N representing the input data and y_N representing the corresponding label it gets assigned, while n labels the entries in the data set. The goal of supervised learning is to predict the value of the label for a given input that does not belong to the training set. If y is a continuous variable this problem is termed *regression*, and if it is a discrete set, this problem is termed *classification*.

- **Unsupervised Learning:** Given data that is not labeled, the task is to learn the properties that are intrinsic to the distribution that generated it. Tasks here appear in forms such as dimensionality reduction, feature extraction and representation learning, problems related to representing data in more compact and convenient spaces and generative modeling which involves generating artificial samples that represent the data set.

- **Reinforcement Learning:** In this model, a system working in an environment learns to make the right decisions so as to maximize the rewards given the state of the system and and its accessible environment. This model does not fit itself under supervised learning due to the lack of a function to be optimized neither does it fit under the umbrella of unsupervised learning due to the presence of a feedback loop mechanism of assigning rewards that tells the user about the quality of a given action.

- **Generative modeling:** Here the idea is given the training to try and learn the joint probability distribution of the data that underlies it. It can be visualized as learning

the features of a vector along with a label that corresponds to it. Classifiers based on this method include Naive Bayes classifier, Bayesian network, Markov random fields. These techniques are well suited in generating artificial samples that correspond to a given class and hence the name.

- **Discriminative modeling:** Here the idea is given the training data to try and learn the conditional probability distribution of the label given the features. Classifiers based on this method include Logistic regression, Support Vector Machine, Traditional neural networks, Nearest neighbor. These techniques are well suited in doing classification tasks, thus justifying the name.

1.1.1 Physics Informed Machine Learning

When attempting to solve a physical problem, that involves governing physical laws, it becomes crucial to integrate thee knowledge of those laws rather than relying on the ML model to make arbitrary decisions. This is especially important when there is scarcity in the amount of available data. Intuitively, utilising the information provided by the physical laws, restrict the solution space for the ML model output. This increases the robustness and accuracy of the network being trained on the dataset describing the physical problem, as it can generalize well on other problems described by the same governing equations. A typical workflow is described in Fig. 1.2

1.2 Quantum Circuit Model

A quantum system with two degrees of freedom is called a qubit. In the gate model of the quantum computing paradigm, transformations between states are achieved using unitary matrices which are represented as 'Quantum Gates'. Since all quantum gates are unitary, the inverse of such gates necessarily exists and hence transformations using quantum gates alone are always reversible. The way to incorporate irreversibility in the paradigm is by making projective measurements that disturb the state vector irrevocably making it lose its present memory (interactions with the environment induce irreversibility too in the form of qubit decoherence).

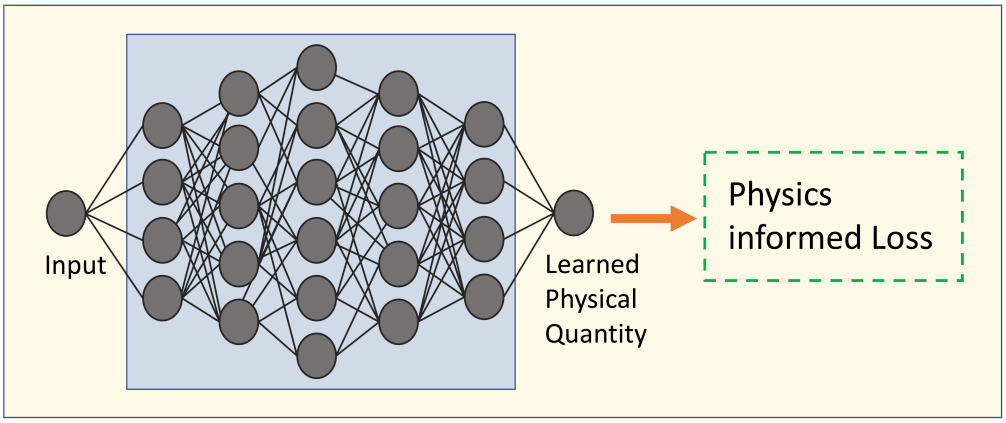

Figure 1.2. The modeling framework for Physics Informed Machine Learning. The loss that is used for minimization comes from the physical governing expression and thus it can be stated that the network learns physical entities.

A single qubit pure state in the two-dimensional basis of $(|0\rangle, |1\rangle)$ can be written in the form:

$$|\psi\rangle \quad = \cos(\tfrac{\theta}{2})|0\rangle + \mathrm{e}^{\mathrm{i}\phi}\sin(\tfrac{\theta}{2})|1\rangle \tag{1.1}$$

with the vector $\Big(n_x, n_y, n_z\Big) = \Big(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta\Big)$ residing on a *Bloch sphere*, shown in Fig 1.3 made up of parameterized angles $\{\theta \in [0, \pi], \phi \in [0, 2\pi]\}$ defined in the Bloch sphere. For multiple qubits (say N), the corresponding state space is $\mathbb{H}_{\mathbb{A}} \otimes \mathbb{H}_{\mathbb{B}} \otimes \mathbb{H}_{\mathbb{C}} ....\mathbb{H}_{\mathbb{N}}$ with the computational basis defined using the Kronecker product as $|\mathrm{i}_A\rangle \otimes |\mathrm{i}_B\rangle .....|\mathrm{i}_N\rangle$ where the labels $(A, B, C...N)$ are physically used to separate the state-space of each qubit.

A multi-qubit pure state can be written as:

$$|\psi\rangle_{A,B,C...N} \quad = \quad \sum_{\mathrm{i}_A=0}^{1}\sum_{\mathrm{i}_B=0}^{1} ... \sum_{\mathrm{i}_N=0}^{1} C_{\mathrm{i}_A\mathrm{i}_B\mathrm{i}_C...\mathrm{i}_N}|\mathrm{i}_A\mathrm{i}_B...\mathrm{i}_N\rangle \tag{1.2}$$

with the coefficients $C_{\mathrm{i}_A\mathrm{i}_B\mathrm{i}_C...\mathrm{i}_N} \in \mathbb{C}^2 \;\; \forall\, \mathrm{i_j} \;\; , \mathrm{j} \in \{A, B, ...N\}$.

A quantum circuit is essentially an assembly of quantum gates which transforms an initial state of a multi-qubit system to a final desired state. The set of quantum gates operationally represents a user-defined unitary transformation. Such operations are frequently followed by

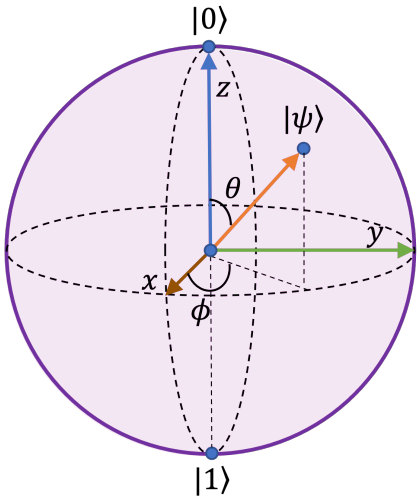

Figure 1.3. Bloch sphere representation of a qubit along with the parametric angle θ and ϕ corresponding to the point represented by $|\psi\rangle$. If the point is assumed to lie on the surface of the sphere and can be represented by $\Big(n_x, n_y, n_z\Big) = \Big(\cos\theta \sin\phi, \sin\theta \cos\phi, \cos\theta\Big)$, then it is deemed a pure state and can be represented by Eq. 1.1

measurement either in the computational basis or in the basis of the operator whose statistics in the prepared state is desired [19] . Some frequently used gates are listed as follows:

- $R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\mathrm{i}\sin\frac{\theta}{2} \\ -\mathrm{i}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$
- $R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$
- $R_z(\theta) = \begin{pmatrix} \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}} & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\frac{\theta}{2}} \end{pmatrix}$

- $\mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- $\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$
- $\text{Controlled-U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & u_{00} & 0 & u_{10} \\ 0 & 0 & 0 & 1 \\ 0 & u_{10} & 1 & u_{11} \end{pmatrix}$

1.3 Variational Quantum Algorithms (VQAs)

A variational quantum algorithm, in general, is a quantum-classical hybrid network, comprising Parameterized Quantum Circuits (PQCs). These variational algorithms can be thought of as similar to ML models, where certain parameters of the model are tuned (during training) to obtain the desired result. The difference however, comes from the fact that PQCs are made up of parameterized unitaries, wherein the unitary operations have tunable parameters denoted by θ as shown in Fig 1.4. An archetypal workflow of VQAs is as listed:

- A state preparation is performed, where the qubits are initialized to the desired state.
- An appropriate parameterized unitary is constructed with the required set of single and multi-qubit operations. Usually, the quantum circuit is transpiled and decomposed into a bunch of elementary gate operations as allowed by the appropriate quantum hardware to execute the required operation.

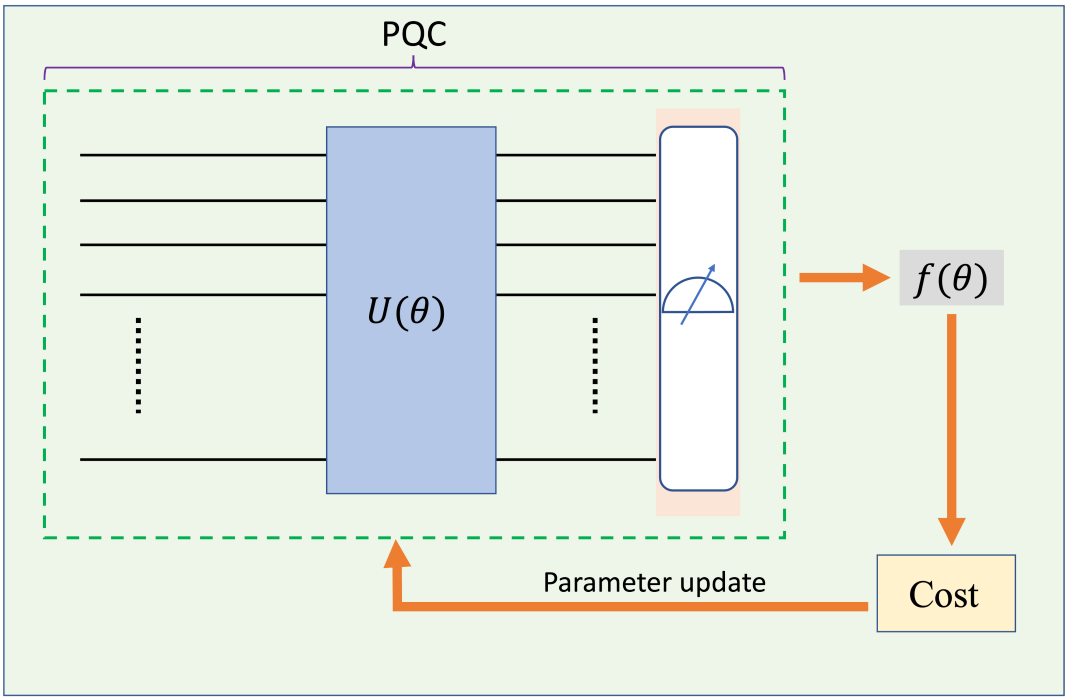

Figure 1.4. A typical variational quantum algorithm, consisting of a PQC. $U(\theta)$ denotes a parameterized multi-qubit unitary, comprising of parameters θ.

- An observable of choice is measured after the action of the unitary.
- The expectation value, denoted by $f(\theta)$ is evaluated and is utilised in a well-defined cost function.
- A classical optimization is performed to identify parameters that reduce the cost and are plugged into the unitary for the second iteration of this process.

In this way, 'learning' of the parameters is executed for a given task.

VQAs are run with reduced quantum resource allocation by using shallow quantum circuits for carrying out computations on a quantum device. This is because, in the near-term hardware, only a short gate sequence, i.e. the circuit depth, can be executed before decoherence is reached. This has garnered popularity for using VQAs to show the possible implications of using quantum computations for certain complex tasks.

1.4 Thesis Overview

A brief highlight that captures the content in each of the chapters in this report is listed as follows:

- In chapter 2, we introduce a Restricted Boltzmann Machine (RBM) based variational quantum algorithm to evaluate the valence bands of two-dimensional materials viz. monolayer graphene and hexagonal Boron Nitride (h-BN). We present the quadratic scaling in resources and how by considering a particular coefficient to the Boltzmann distribution, the probbaility of successful sampling can be improved. We also benchmark the algorithm on IBM-Quantum systems and establish the fidelity of the resulting eigenstates.

- In chapter 3, we propose how the RBM based algorithm can be used to filter any energy eigenstate of the system based on either the symmetry properties or a predefined input choice. We specifically exhibit the results for monolayer transition metal dichalcogenides. Also, a generic lower bound for the successful sampling of the quantum circuit in the algorithm is derived in terms of the parameters of the network. A proof of feasibility of the cost function used to train the network is also presented. We also report the results from a molecular example, LiH wherein the multireference correlation is important due to geometric distortion.

- In chapter 4, we show the application of the RBM based algorithm to estimate the critical point for the quantum phase transition in the Quantum Rabi Model by proposing an alternative Finite Size Scaling method in which the truncation of the system is done in the Hilbert space instead of the physical space.

- In chapter 5, we address the issue of finding the optimal parameters for a weighted graph for a specific quantum optimization algorithm. We develop parameter setting heuristics for the Quantum Approximate Optimization Algorithm (QAOA) applied to a general class of weighted problems. Having derived the optimal parameters for QAOA with depth $p = 1$ applied to the weighted MaxCut problem under different assumptions

on the weights, we numerically validate this approach on general weighted graphs and show that on average the QAOA energy with the proposed fixed parameters is only 1.1 percentage points away from that with optimized parameters.

- In chapter 6, we venture into the realm of thermography. Taking advantage of intrinsic attributes of thermal radiation, we devise a method for passive machine perception. We also show how a deep learning based neural network can solve an inverse problem to decompose the hyperspectral thermal images into their constituent components and their implications toward practical computer vision are presented.

References

[1] S. Ulam, R. D. Richtmyer, and J. von Neumann, "Statistical methods in neutron diffusion," *LAMS-551, Los Alamos National Laboratory*, pp. 1–22, 1947.

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[3] B. J. Alder and T. E. Wainwright, "Studies in molecular dynamics. i. general method," *The Journal of Chemical Physics*, vol. 31, no. 2, pp. 459–466, 1959.

[4] A. Rahman, "Correlations in the motion of atoms in liquid argon," *Physical review*, vol. 136, no. 2A, A405, 1964.

[5] J. C. Slater, "Note on hartree's method," *Physical Review*, vol. 35, no. 2, p. 210, 1930.

[6] V. Fock, "Näherungsmethode zur lösung des quantenmechanischen mehrkörperproblems," *Zeitschrift für Physik*, vol. 61, no. 1, pp. 126–148, 1930.

[7] F. Coester, "Bound states of a many-particle system," *Nuclear Physics*, vol. 7, pp. 421–424, 1958.

[8] J. Čížek, "On the correlation problem in atomic and molecular systems. calculation of wavefunction components in ursell-type expansion using quantum-field theoretical methods," *The Journal of Chemical Physics*, vol. 45, no. 11, pp. 4256–4266, 1966.

[9] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, "A fifth-order perturbation comparison of electron correlation theories," *Chemical Physics Letters*, vol. 157, no. 6, pp. 479–483, 1989.

[10] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Physical review*, vol. 136, no. 3B, B864, 1964.

[11] N. M. Ball and R. J. Brunner, "Data mining and machine learning in astronomy," *International Journal of Modern Physics D*, vol. 19, no. 07, pp. 1049–1106, 2010.

[12] Ž. Ivezić, A. J. Connolly, J. T. VanderPlas, and A. Gray, *Statistics, data mining, and machine learning in astronomy*. Princeton University Press, 2014.

[13] O. A. von Lilienfeld and K. Burke, "Retrospective on a decade of machine learning for chemical discovery," *Nature communications*, vol. 11, no. 1, pp. 1–4, 2020.

[14] Y. Liu, C. Niu, Z. Wang, *et al.*, "Machine learning in materials genome initiative: A review," *Journal of Materials Science & Technology*, vol. 57, pp. 113–122, 2020.

[15] A. Kumar, S. Loharch, S. Kumar, R. P. Ringe, and R. Parkesh, "Exploiting cheminformatic and machine learning to navigate the available chemical space of potential small molecule inhibitors of sars-cov-2," *Computational and structural biotechnology journal*, vol. 19, pp. 424–438, 2021.

[16] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*, Ieee, 1994, pp. 124–134.

[17] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[19] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, 2002.

2. QUANTUM MACHINE LEARNING FOR VALENCE BAND CALCULATIONS

The contents of this chapter are adapted from the article 'Sureshbabu, Shree Hari, et al. "Implementation of quantum machine learning for electronic structure calculations of periodic systems on quantum computing devices." Journal of Chemical Information and Modeling 61.6 (2021): 2667-2674'.

2.1 Introduction

Machine learning (ML) driven by big data and computing power has made a profound impact on various fields, including science and engineering [1]. Remarkably successful applications of machine learning range from image and speech recognition [2], [3] to autonomous driving [4]. The recent success of machine learning is mainly due to the rapid increase in classical computing power. This impact of ML has made it a useful tool to solve various problems in physical sciences [5]. Quantum computing is a new way of computation by harnessing the quantum properties such as the superposition and entanglement of quantum states. Some quantum algorithms run on quantum computers could solve the problems which are intractable by classical computers [6]. Recent progress in the development of Noisy Intermediate-Scale Quantum (NISQ) devices [7], makes it possible to run and test multiple quantum algorithms for various practical applications.

Quantum machine learning [8], the interplay of classical machine learning techniques with quantum computation, provides new algorithms that may offer tantalizing prospects to improve machine learning. At the same time, these techniques aid in solving the quantum many-body problems [9]–[14]. Using neural networks with a supervised learning scheme, Xu *et al.* [15] have shown that measurement outcomes can be mapped to the quantum states for full quantum state tomography. Cong *et al.* [16] have developed a quantum machine learning model motivated by convolutional neural networks, which makes use of $\mathcal{O}(\log(N))$ variational parameters for input sizes of N qubits that allows for efficient training and implementation on near term quantum devices.

It is important to solve many-body problems accurately for the advancement of material science and chemistry, as various material properties and chemical reactions are related to quantum many-body effects. Carleo and Troyer [17] introduced a novel idea of representing the many-body wavefunction in terms of artificial neural networks, specifically restricted Boltzmann machines (RBMs), to find the ground state of quantum many-body systems and to describe the time evolution of the quantum Ising and Heisenberg models. This representation was modified by Torlai *et al.* [18] for their purpose of quantum state tomography in order to account for the wavefunction's phase.

Quantum chemistry and electronic structure calculations using quantum computing are considered one of the first real applications of quantum computers [19]–[23]. Xia and Kais [24] proposed a quantum machine learning method based on RBM to obtain the electronic structure of molecules. The traditional two-layer RBM was extended to three layers to take into account the signs of the coefficients for the basis functions of the wave function. This method was applied to molecular and spin-lattice systems. Recently, Kanno *et al.* [25] have extended the method proposed by Xia and Kais by providing an additional unit to the third layer of an RBM in order to represent complex values of the wavefunctions of periodic systems.

Since the discovery of graphene, it has sparked a huge interest due to its remarkable properties. Recently, there has been a lot of interest in studying graphene for quantum computing applications [26], [27]. Hexagonal Boron Nitride (h-BN) gained attention when it was shown that graphene electronics is improved when h-BN is used as a substrate for graphene [28]. Of late the interest to study h-BN for quantum information has grown since it was discovered that the negatively charged Boron vacancy spin defects in h-BN display spin-dependent photon emission at room temperature [29]–[31]. Hence, in addition to studying graphene, it is important to study h-BN as it is a potential candidate for creating spin qubits that can be optically initialized and readout.

In this chapter, we implement the quantum machine learning method with a three-layered RBM along with a quantum circuit to sample the Gibbs distribution [24], [25] to calculate the electronic structure of periodic systems. Specifically, the implementation on NISQ devices is shown by modifying this quantum machine learning algorithm to run on an actual

quantum computer. As the benchmark test, we demonstrate the performance of this algorithm first through the simulation of tight-binding and Hubbard Hamiltonians of hexagonal Boron Nitride and monolayer-graphene respectively, on the IBM quantum computing processors, which is done using the IBM quantum experience [32]. The valance band of the 2-D honeycomb lattices is calculated using quantum machine learning methods on IBM-Q and the Qiskit simulator. As we shall see such valence band calculations on IBM-Q after employing a warm start and measurement error mitigation are shown to be in good agreement with the exact calculations.

2.2 Methodology

In this section, we review the basic outline of the machine learning algorithm used and also discuss the implementation details

2.2.1 Quantum Machine Learning Algorithm

A quantum many-body state $|\Psi\rangle$ can be expanded in terms of the basis $|\mathbf{x}\rangle$, $|\Psi\rangle = \sum \Psi(\mathbf{x}) |\mathbf{x}\rangle$ where $\Psi(\mathbf{x})$ is the wavefunction. Carleo and Troyer's [17] method involved representing the trial wave function $\Psi(\mathbf{x};\theta)$ in terms of a neural network with parameters θ and to obtain the ground state by minimizing the expectation value of the Hamiltonian of a quantum many-body system, $E(\theta) = \langle\Psi(\theta)| H |\Psi(\theta)\rangle$. This was shown to use lesser number of parameters compared to tensor-networks, indicating the efficiency of using such a representation. More specifically, the ansatz of a trial wave function is given by the marginal probability $P(\mathbf{x};\theta)$ of a visible layer of the RBM, $\Psi(\mathbf{x};\theta) = \sqrt{\mathbf{P}(\mathbf{x};\theta)}$. While the learning of conventional RBMs is done by maximizing the likelihood function with respect to training data sets, the ground state of a neural network RBM state is obtained by minimizing the energy $E(\theta)$ using the stochastic optimization algorithm.

Xia and Kais [24] introduced the third layer with a single unit to take into account the signs of the wavefunction and apply the quantum Restricted Boltzmann machine on actual quantum computers rather than the Monte-Carlo method on classical digital computers. This quantum machine learning algorithm was further extended to take into account the

complex value of the wavefunction [25]. However, implementation on an actual quantum computing processor was not shown, which would require multiple ancillary qubits as shown in this work.

The RBM we consider here consists of three layers: a visible layer, a hidden layer, and a complex layer, as shown in Fig. 2.1. In contrast with the conventional RBMs with visible and hidden layers, the complex layer is added to take into account the real and imaginary values of the wavefunction of a quantum state.

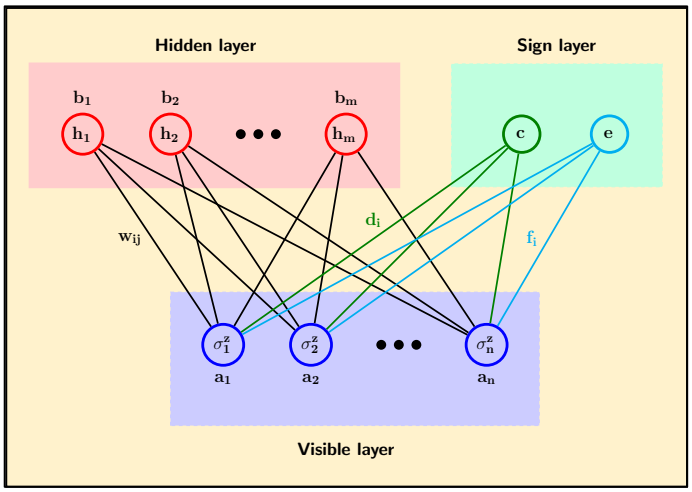

Figure 2.1. Restricted Boltzmann Machine used to calculate the electronic structure of periodic materials. Here, the sign layer consists of two units, one to account for the real part and the other for the complex part of the wavefunction.

The wavefunction of a periodic system can be expressed as:

$$|\Psi\rangle = \sum_{\mathbf{x}} \sqrt{P(\mathbf{x})} s(\mathbf{x}) \, |\mathbf{x}\rangle \ , \tag{2.1}$$

where

$$P(\mathbf{x}) = \frac{\sum_{\{h\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}} \sigma_{\mathrm{i}}^{z} + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}} \sigma_{\mathrm{i}}^{z} h_{\mathrm{j}}}}{\sum_{\mathbf{x}'} \sum_{\{h\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}} \sigma_{\mathrm{i}}^{z'} + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}} \sigma_{\mathrm{i}}^{z'} h_{\mathrm{j}}}} \tag{2.2}$$

$$s(\mathbf{x}) = \tanh \left[(c + \sum_{\mathrm{i}} d_{\mathrm{i}} \sigma_{\mathrm{i}}) + \mathrm{i}(\mathrm{e} + \sum_{\mathrm{i}} f_{\mathrm{i}} \sigma_{\mathrm{i}}) \right] \tag{2.3}$$

Here σ_i^z is the z-component of the Pauli operators at site i, $|\mathbf{x}\rangle = |\sigma_1^z \sigma_2^z \sigma_3^z ... \sigma_n^z\rangle$ is the basis vector and the values that σ_i^z and h_j take are {+1, -1}. a_i, b_i, c, and e denote the trainable bias parameters of the visible units, the hidden units, the unit representing the real part of the complex layer, and the unit representing the complex part of the complex layer, respectively. w_{ij}, d_i, and f_i denote the trainable weights corresponding to the connections between σ_i^z and h_j, σ_i^z and the unit representing the real part of the complex layer, σ_i^z and the unit representing the complex part of the complex layer, respectively. All the parameters are randomly initialized and the values of these random numbers range from -0.02 to 0.02.

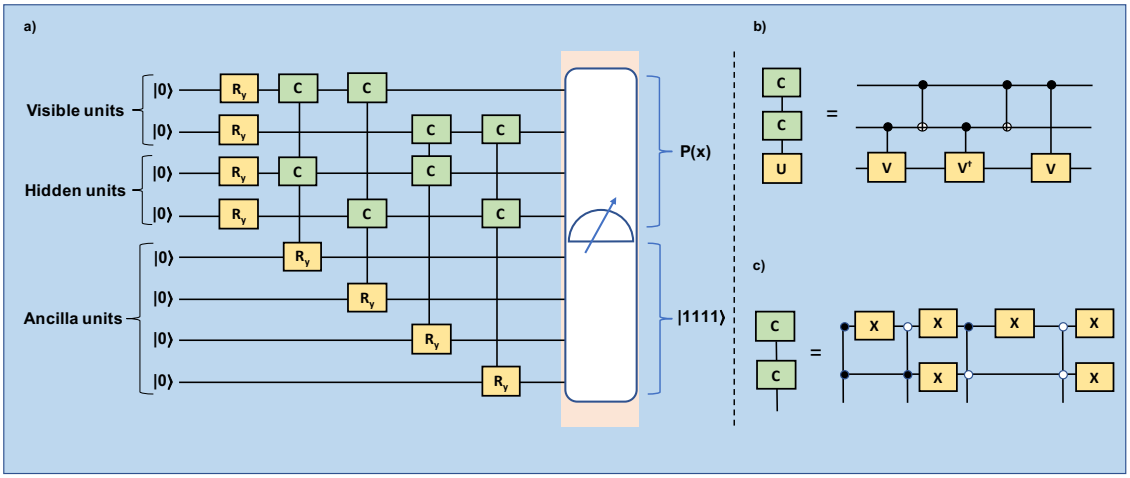

Figure 2.2. a) Quantum circuit to sample Gibbs distribution. This circuit consists of 2 visible units, 2 hidden units, and 4 ancilla qubits. R_y represents the single qubit rotation, $C-C-R_y$ represents the controlled-controlled rotation, with visible and hidden units being the control qubits and ancilla qubit being the target qubit. After measurement, if the ancilla qubits are in $|1111\rangle$, only then the qubits correspnding to the visible and hidden units give the distribution $P(\mathbf{x})$. b) Decomposition of the $C-C-R_y$ gate for $|11\rangle$. Here $U = V^2$ and this leads to choosing $V = R_y(\theta/2)$. c) $C-C-R_y$ conditioned by $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ can be achieved by implementing the circuit in this form.

In order to obtain the probability distribution, the quantum circuit (shown in Fig. 2.2) is employed. The quantum circuit consists of a single qubit rotation (R_y) and a controlled-controlled rotation operations ($C-C-R_y$). The angle by which the R_y operation rotates is determined by the visible and hidden bias parameters a_i and b_j. The angle by which the $C-C-R_y$ operation rotates is determined by the weights connecting the visible and hidden layers w_{ij}. For each combination of visible and hidden units, $y = \{\sigma^z, h\}$, in order to

increase the probability of successful sampling, the distribution $Q(y)$ is sampled rather than $P(y)$ [24]. The two distribution functions $P(y)$ and $Q(y)$ are given by

$$P(y) = \frac{e^{\sum_i a_i \sigma_i^z + \sum_j b_j h_j + \sum_{ij} w_{ij} \sigma_i^z h_j}}{\sum_{y'} e^{\sum_i a_i \sigma_i^{z'} + \sum_j b_j h'_j + \sum_{ij} w_{ij} \sigma_i^{z'} h'_j}} \tag{2.4}$$

$$Q(y) = \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j + \sum_{ij} w_{ij} \sigma_i^z h_j)}}{\sum_{y'} e^{\frac{1}{k}(\sum_i a_i \sigma_i^{z'} + \sum_j b_j h'_j + \sum_{ij} w_{ij} \sigma_i^{z'} h'_j)}} \tag{2.5}$$

Here, k is taken as $max(1, \frac{|w_{ij}|}{2})$ [24]. This is done in order to make the lower bound of the probability of successful sampling a constant. If k is taken to be 1, then the number of measurements required to get successful sampling becomes exponential. (See Appendix A).

The target qubits for the controlled-controlled Rotations are the ancilla qubits. Once all the rotations are completed, the ancilla qubits are measured. If the ancilla qubits are in $|1\rangle$, then the sampling is deemed successful. Then, the qubits corresponding to the visible and hidden units are measured to obtain the distribution $Q(y)$. Once the distribution $Q(y)$ is obtained, the probabilities are calculated to the power of k and then normalized to get $P(y)$. With $P(y)$ computed through our QML algorithm and $s(y)$ computed classically, the wavefunction $|\psi\rangle$ is computed and through this the energy $E(\theta)$ is obtained. This value of $E(\theta)$ is optimized through gradient descent until the eigenvalue of the Hamiltonian is obtained.

For this algorithm, the number of qubits required scales as $O(nm)$ and the complexity of the gates turns out to be $O(nm)$ for one sampling [24], where, n is the number of visible units and m being the number of hidden units.

2.2.2 Implementation methods

The developed quantum machine learning algorithm for calculating the band structures of h-BN and monolayer-graphene is executed using the following tools:

(i) We start with the implementation of the algorithm classically. Classical simulation is performed to ensure the algorithm performs accurately. Here classical simulation implies that the gates were simulated on a classical computer.

(ii) Having ensured that the algorithm works when implemented classically, we move on to implementing it using Qiskit [32]. Qiskit stands for IBM's Quantum Information Software Kit (Qiskit) and is designed to mimic calculations performed on a real noisy-intermediate scale quantum computing device using a classical computer. Specifically, we implemented the algorithm on the *qasm* backend, which is a high-performance quantum circuit simulator amenable to treat the errors (noise) associated with the implementation of the quantum circuit with appropriate customizable noise models. Essentially, the *qasm* simulator is designed to replicate an actual noisy quantum device. Even if a custom noise is not chosen, depending on the circuit being executed the simulator automatically assumes a noise consistent with the hardware of the real device. The $C-C-R_y$ gate can be implemented by using *qiskit's* multi-controlled y-rotation (mcry) operation, by specifying the control, target and ancillary qubits. The circuit is executed multiple times on the simulator each time culminating in the chosen set of measurements. The return values are the probabilities for observing the system in measurement basis states with statistical errors due to finite sampling.

(iii) We conclude our discussion by implementing and demonstrating the validity of our results using two actual IBM-Q quantum computers available. Qiskit's results in (i) are compared with those obtained from these real quantum devices.

In the following section, we display the simulation results. The terms 'RBM Value' and 'Exact Value' stand for the values of valence band energies obtained from training our RBM and from exact diagonalization of the Hamiltonian, respectively.

Initializing the parameters of the RBM randomly can lead to the energies corresponding to certain k points being stuck at local minima. To enhance the generalizing capability of a machine learning model, transfer learning technique has been successfully used. Recently, it has also been extended to the realm of quantum computing [33]. However, in our case in order to improve the convergence, a method of warm starting is sufficient, wherein the parameters of a previously converged point are used to initialize the parameters of the current point of calculation. Noting that the band structure exists in a 4D space corresponding to energy as a function of k_x, k_y, and k_z, in this case too, if the optimization is performed such

that the energy is minimized for every (k_x, k_y, k_z) point, then the parameters of such a point in 4D space can be considered to improve the convergence of the other points.

When implementing the algorithm on NISQ devices, we have to account for the noise that interferes with the accuracy of the results. In this work, we try to mitigate the errors that occur during measurement using *Measurement Error Mitigation*.

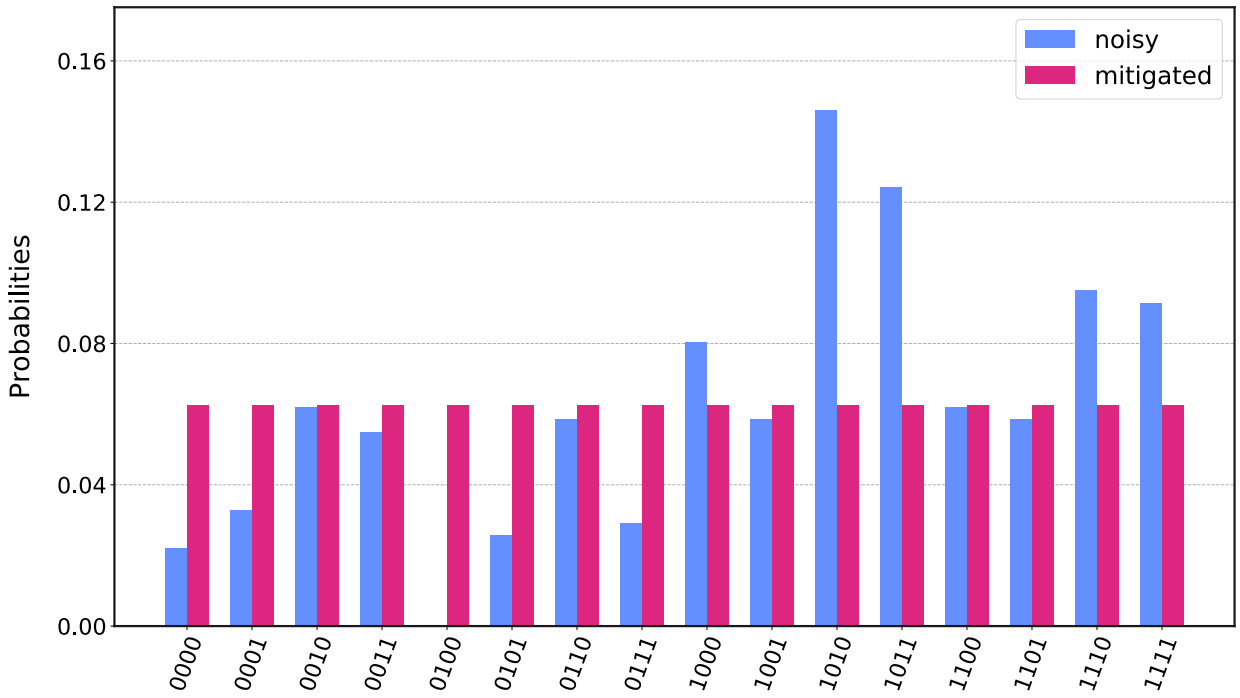

Figure 2.3. The probabilities of states with ancilla qubits being in $|1111\rangle$ for both the cases of with and without measurement error mitigation for the first iteration.

The counts corresponding to each state will not be definite as a result of noise. There will be a finite number of counts corresponding to the other basis states even when the measurement outcome is supposed to result in one. So the counts for each state can be written as a column vector and a matrix, called the calibration matrix, can be defined corresponding to the concatenation of all column vectors describing the counts for all the basis states. The least-squares method can now be used to get the error mitigated probabilities for each of the states by using the calibration matrix, the ideal state vector, and the noisy result that was obtained [32]. An example of the probability distribution $Q(y)$ obtained with and without measurement error mitigation is shown in Fig. 2.3.

2.3 Results and Discussion

As a benchmark test of our quantum machine learning algorithm on existing IBM quantum computers, we calculate the electronic structures of two well-studied 2-dimensional periodic systems with hexagonal lattices namely Boron-Nitride and monolayer graphene. In this section, we discuss the results for each of the two systems.

2.3.1 Band Structure of h-BN

Hexagonal Boron nitride (h-BN) has a unit cell containing one B atom and another N atom. For h-BN, the levels involving the other valence orbitals, the $2s, 2p_x$, and $2p_y$, are either quite far above or far below the Fermi level. The conduction and valence bands, which are around the Fermi level, are formed from the $2p_z$ orbital and hence, a tight-binding Hamiltonian using the frontier $2p_z$ orbital and with third-nearest neighbor interaction on each of the two atoms of the unit cell is employed to obtain the electronic structures of the materials. Such a treatment affords the requisite dimensionality reduction as the number of qubits available on the IBM quantum computers is limited. Considering spin-degeneracy, the tight-binding Hamiltonian of the h-BN is thus given by a 4×4 Hermitian matrix (see Appendix A). The number of visible units needed for the simulation is 2, and the number of hidden units is taken to be equal to the number of visible units. For quantum optimization, 2 qubits are used to represent the visible nodes and 2 qubits to represent the hidden nodes. In addition, 4 ancillary qubits are required (see Fig. 2.2). In total, the number of qubits required is equal to 8. The sampling of Gibb's distribution is performed by applying the following sequences of quantum gates: 4 single-qubit rotation gates (R_y), 16 controlled-controlled Rotation gates ($C - C - R_y$), and 24 bit-flip (X) gates, as illustrated in Fig. 2.2.

For h-BN band structure calculation, we start with the results of training RBM by implementing the gate-set (see Fig. 2.2) classically and then on the Qiskit's quantum simulator, called the *qasm* backend. Fig. 2.4 (a) shows the band structures of h-BN as a function of wave-vector amplitude sampled from the 1st Brillouin zone. We overlay the valence band energies obtained from our RBM network on a classical computer with the exact diagonalization of the 4×4 tight-binding Hamiltonian (black curve). The two results are in excellent

agreement. It must be noted that without a warm start, results may show deviations from the exact value at certain k-points as the optimization protocol may get locally trapped. However, the use of the warm starting technique eliminates such convergence issues. Fig. 2.4 (b) shows the band structure calculation of h-BN wherein for the RBM, the quantum gates are implemented on the Qiskit *qasm* backend. For the sake of our simulations, no noise model was considered and the results obtained are just with statistical errors. Even in this case, if a warm start is provided, the quantum machine learning algorithm on the Qiskit *qasm* simulator renders the exact valence band. In Fig. 2.4 (c) we show the implementation results for the valence band calculations using RBM wherein the gate-set is implemented on real IBM quantum devices, namely the *ibmq_toronto* and *ibmq_sydney*, both of which are 27 qubit devices. We see the results are in excellent agreement with the exact diagonalization when a warm start is provided along with *Measurement Error Mitigation*.

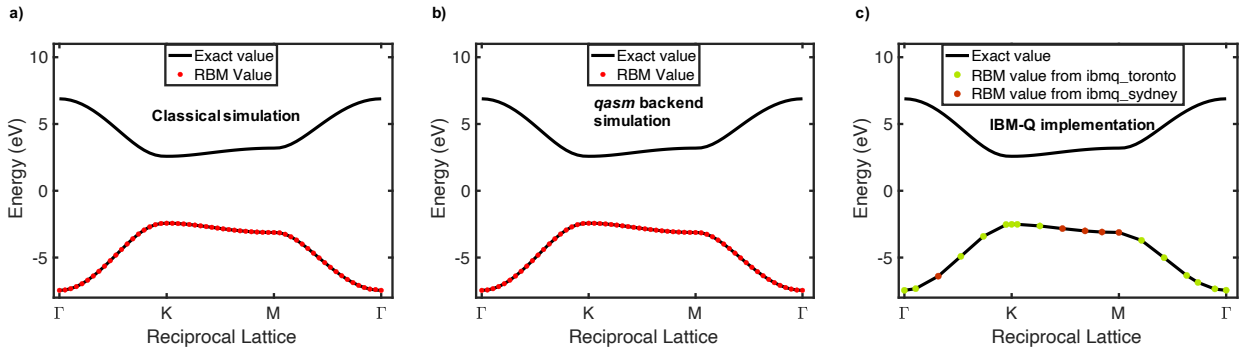

Figure 2.4. Band structures of h-BN calculated using (a) classical simulation with a warm start (red). The solid black curves show the valence and conduction bands from exact diagonalization. (b) the *qasm* backend simulation with the aid of a warm start (red). (c) The implementation the RBM sampling circuit on i*bmq_toronto* (green) and i*bmq_sydney* (red).

2.3.2 Band Structure of monolayer Graphene

Much like h-BN, monolayer graphene also consists of two atoms in its unit cells. However, unlike the previous case, both the atomic centers are made up of carbon. Also, similar to h-BN, in the case of graphene, the levels involving the other valence orbitals, the $2s, 2p_x$, and $2p_y$, are either quite far above or far below the Fermi level. The orbital responsible

for electrical conduction is just the $2p_z$ orbital and hence, a tight-binding Hamiltonian for the valence and conduction band with third-nearest neighbor interaction is constructed by taking into account the frontier $2p_z$ orbital on each of the carbons. The resultant matrix as before is a 4×4 matrix including spin-degeneracy(see Appendix A). We introduce spin-spin interaction in graphene using the Fermi-Hubbard model with an onsite repulsion parameter U between opposite spins. In order to simulate graphene, the number of visible units and the number of hidden units is equal to 2. Therefore, 2 qubits to represent the visible nodes and 2 qubits to represent the hidden nodes, and in addition to that, 4 ancilla qubits are required. In total, the number of qubits required is equal to 8. The number of quantum gates required to sample Gibb's distribution is 4 single qubit Rotation gates (R_y), 16 Controlled-Controlled Rotation gates ($C - C - R_y$), and 24 Bit-flip (X) gates.

The band structures of monolayer graphene are calculated using the IBM Qiskit simulator and by running the QML algorithm on the IBM-Q quantum computers. Fig. 2.5 (a) shows the results for the band structures of graphene at zero U using the classical simulation. As before the results are overlayed on top of the eigenvalues obtained from exact diagonalization of the 4×4 Hamiltonian. In Fig. 2.5 (b) we show the band structure of the graphene for $U = 0$ calculated using the Qiskit qasm simulator. Finally, in Fig. 2.5 (c) we show the results of the quantum machine learning algorithm for calculation of the band structures of the graphene on IBM-Q quantum computers, the *ibmq_toronto*, and *ibmq_sydney*. Even for the case of graphene, the results are in good agreement with the exact diagonalization when a warm start is provided along with *Measurement Error Mitigation*.

To show the band splitting for a non-zero on-site repulsion U, the Fermi level is shifted by a chemical potential $\mu = 15$ eV, which controls the filling of electrons. Fig. 2.5 (d-e)plots the band structures of graphene for $U = 9.3$ eV obtained using the classical simulation, Qiskit *qasm* backend, and the actual implementation on an IBM quantum computer. The RBM results are again in good agreement with that from exact diagonalization in all of the cases.

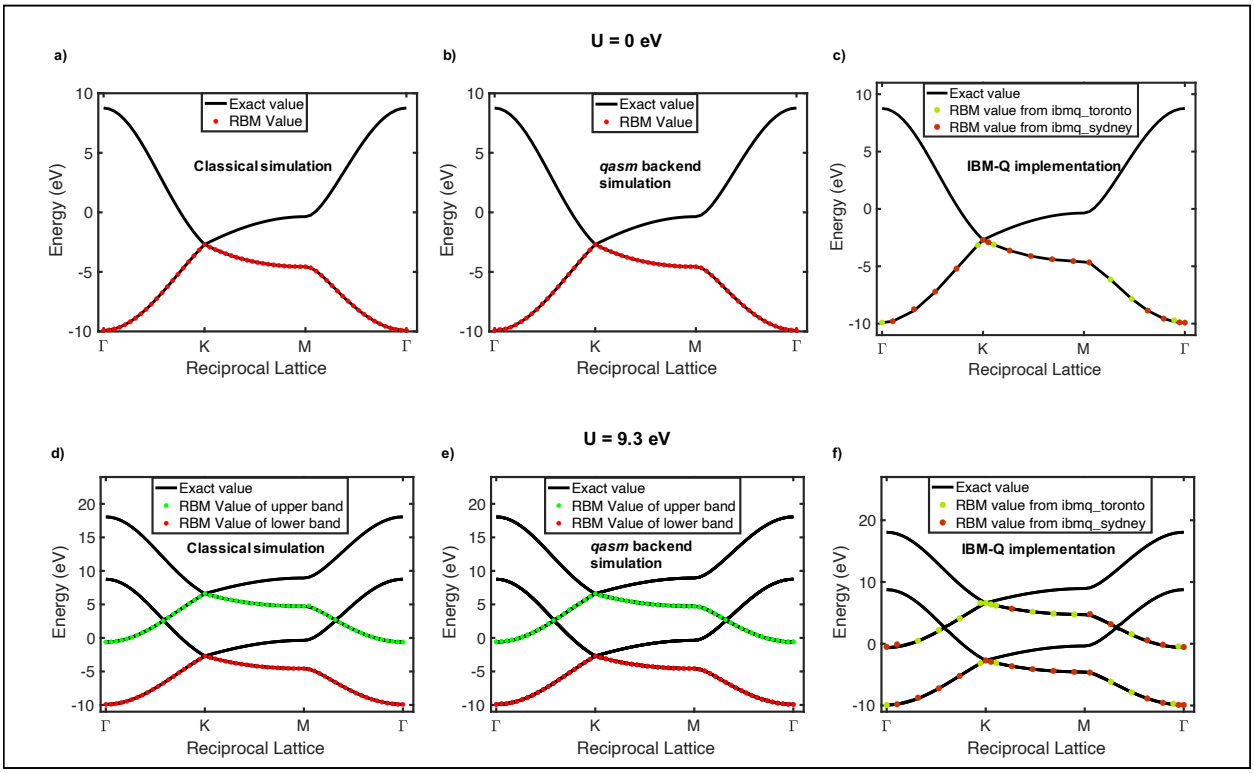

Figure 2.5. Band structure of the graphene for $U = 0\,eV$ calculated using (a) classical simulation with a warm start (red). The solid black curves show the valence and conduction bands from exact diagonalization, (b) the *qasm* backend simulation with the aid of a warm start (red). (c) Implementation on actual IBM computing devices. (d) Same as in (a) with Hubbard on-site interaction $U = 9.3\,eV$. The four bands correspond to the two non-degenerate spin-states for each of the valence and conduction bands in plot (a). (e) Same as in (b) with Hubbard on-site interaction $U = 9.3\,eV$. (f) Same as in (c) with Hubbard on-site interaction $U = 9.3\,eV$.

2.3.3 Fidelity

To verify if the eigenstates provided by the QML algorithm match those obtained from exact diagonalization, the fidelity for each k point is calculated. It can be seen from Fig. 2.6 that the error (1-Fidelity) is very small for classical simulation and simulation on the *qasm* backend for both the materials. The fidelity is calculated as follows:

$$\text{Fidelity} = |\langle\Psi|\Phi\rangle|^2$$

where, $|\Psi\rangle$ is the eigenvector obtained from QML and $|\Phi\rangle$ is the eigenvector obtained from exact diagonalization.

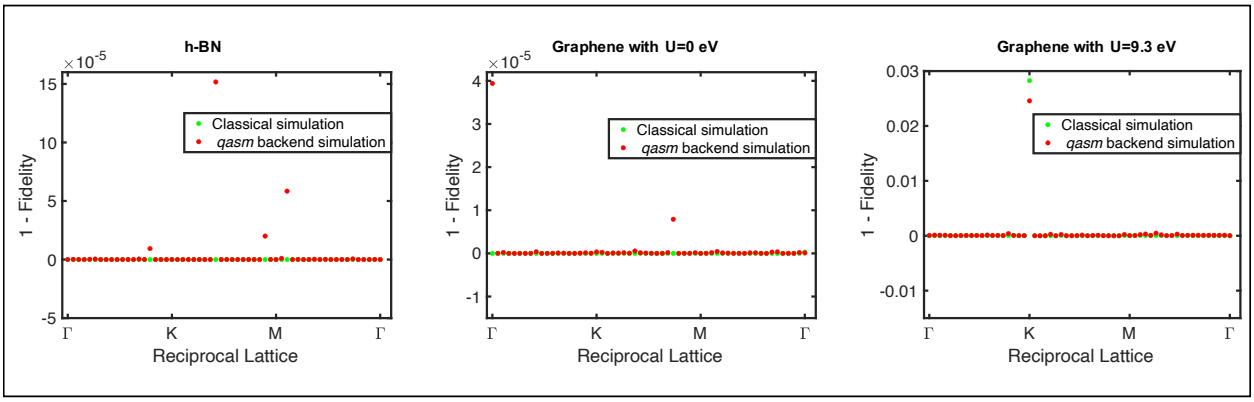

Figure 2.6. Error in fidelity $(1 - F)$, are plotted as a function of the reciprocal lattice vector (k) for classical simulation and *qasm* backend.

2.4 Conclusion

The primary goal of this study was to examine the performance of an RBM on a NISQ device in order to calculate the electronic structure of materials. In this work, the materials that were taken under consideration were hexagonal Boron Nitride (h-BN) and monolayer Graphene, both of which are two-dimensional solids. A tight-binding and a Hubbard Hamiltonian were constructed for h-BN and graphene respectively. By using an RBM and a quantum circuit to sample Gibbs distribution, the valence band energies for each of the two materials were obtained. In the case of graphene, the simulations were performed first for the case when the Hubbard interaction U is equal to 0 and then for the case of non-zero U. The band splitting for the case of non-zero U was also shown. The simulations for both, graphene and h-BN were done using IBM's *qiskit* framework as well as on real IBM quantum computing platforms.

Implementing RBM classically can either use Maximum-likelihood based gradient descent (which has a time complexity that is exponential in the size of the smallest layer)[34] or Contrastive Divergence using Gibbs sampling, a Markov Chain Monte Carlo (MCMC) method (which is a more efficient approach) to estimate the gradients [35]. The time complexity for training an RBM in the classical case scales as $O(N)$, where N is the size of the training data, while the implementation of RBM on a quantum computer has been shown to have quadratic speed-ups [36]. Also, computing the ground state of a given Hamiltonian using

exact diagonalization has a complexity of $\approx q^3$, where q is the dimension of the column space of a given matrix [37]. However, setting $k = max(1, \frac{|w_{ij}|}{2})$ provides a constant lower bound in the probability of successful sampling and thus the complexity for one iteration scales as $O(mnN)$, where N is the number of successful sampling required to get the distribution P(x).

The current quantum machine learning method could calculate only on the ground state energy of the periodic systems, i.e., the valence band, an extension is needed to treat systems with multiple valence bands [38] or to procure higher order energy bands. This can be done by sampling the orthogonal subspace of the previously computed valence band. Also, to calculate the transition matrix elements, the valence and conduction Bloch wavevectors should be obtained. The expectation value of an operator with respect to the ground state may be calculated using the Hellmann-Feynman method [39]. Here, the effect of noise on quantum machine learning is not fully explored, while the Qiskit qasm simulator and IBM-Q noisy quantum computers show the effect of noise on quantum optimization. With the field of quantum computing developing rapidly, the curiosity of combining machine learning and quantum computing has led to very interesting researches. With the development of quantum computers and their capability to scale very fast, quantum machine learning can prove to be useful in not only electronic structure methods, but also as a significant tool in developing new materials and understanding complex phenomena.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] H. Sak, A. Senior, K. Rao, *et al.*, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, pp. 4280–4284.

[4] M. Bojarski, D. Del Testa, D. Dworakowski, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[5] G. Carleo, I. Cirac, K. Cranmer, *et al.*, "Machine learning and the physical sciences," *Rev. Mod. Phys.*, vol. 91, no. 4, p. 45 002, Dec. 2019. DOI: 10.1103/RevModPhys.91.045002. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.91.045002.

[6] F. Arute, K. Arya, R. Babbush, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[7] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, ISSN: 1476-4687. DOI: 10.1038/nature23474. [Online]. Available: http://dx.doi.org/10.1038/nature23474.

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv preprint arXiv:1307.0411*, 2013.

[10] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130 503, 2014.

[11] H. Neven, G. Rose, and W. G. Macready, "Image recognition with an adiabatic quantum computer i. mapping to quadratic unconstrained binary optimization," *arXiv preprint arXiv:0804.4457*, 2008.

[12] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a binary classifier with the quantum adiabatic algorithm," *arXiv preprint arXiv:0811.0416*, 2008.

[13] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a large scale classifier with the quantum adiabatic algorithm," *arXiv preprint arXiv:0912.0779*, 2009.

[14] S. Das Sarma, D.-L. Deng, and L.-M. Duan, "Machine learning meets quantum physics," *Physics Today*, vol. 72, no. 3, pp. 48–54, Mar. 2019, ISSN: 1945-0699. DOI: 10.1063/pt.3.4164. [Online]. Available: http://dx.doi.org/10.1063/PT.3.4164.

[15] Q. Xu and S. Xu, "Neural network state estimation for full quantum state tomography," *arXiv preprint arXiv:1811.06654*, 2018.

[16] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, Aug. 2019, ISSN: 1745-2481. DOI: 10.1038/s41567-019-0648-8. [Online]. Available: http://dx.doi.org/10.1038/s41567-019-0648-8.

[17] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[18] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography," *Nature Physics*, vol. 14, no. 5, pp. 447–450, 2018.

[19] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated quantum computation of molecular energies," *Science*, vol. 309, no. 5741, pp. 1704–1707, 2005.

[20] S. Kais, "Introduction to quantum information and computation for chemistry," in *Quantum Information and Computation for Chemistry*. John Wiley & Sons, Ltd, 2014, ch. 1, pp. 1–38, ISBN: 9781118742631. DOI: 10.1002/9781118742631.ch01. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118742631.ch01. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118742631.ch01.

[21] A. Peruzzo, J. McClean, P. Shadbolt, *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, p. 4213, 2014.

[22] A. Kandala, A. Mezzacapo, K. Temme, *et al.*, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, 2017.

[23] A. Daskin and S. Kais, "Direct application of the phase estimation algorithm to find the eigenvalues of the hamiltonians," *Chemical Physics*, vol. 514, pp. 87–94, 2018.

[24] R. Xia and S. Kais, "Quantum machine learning for electronic structure calculations," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[25] S. Kanno and T. Tada, "Many-body calculations for periodic materials via restricted boltzmann machine-based vqe," *Quantum Science and Technology*, vol. 6, no. 2, p. 025 015, 2021.

[26] I. Joel, J. Wang, D. Rodan-Legrain, *et al.*, "Coherent control of a hybrid superconducting circuit made with graphene-based van der waals heterostructures," *Nature nanotechnology*, vol. 14, no. 2, pp. 120–125, 2019.

[27] I. A. Calafell, J. Cox, M. Radonjić, *et al.*, "Quantum computing with graphene plasmons," *npj Quantum Information*, vol. 5, no. 1, pp. 1–7, 2019.

[28] C. R. Dean, A. F. Young, I. Meric, *et al.*, "Boron nitride substrates for high-quality graphene electronics," *Nature nanotechnology*, vol. 5, no. 10, pp. 722–726, 2010.

[29] A. Gottscholl, M. Kianinia, V. Soltamov, *et al.*, "Initialization and read-out of intrinsic spin defects in a van der waals crystal at room temperature," *Nature materials*, vol. 19, no. 5, pp. 540–545, 2020.

[30] A. Gottscholl, M. Diez, V. Soltamov, *et al.*, "Room temperature coherent control of spin defects in hexagonal boron nitride," *arXiv preprint arXiv:2010.12513*, 2020.

[31] A. L. Exarhos, D. A. Hopper, R. N. Patel, M. W. Doherty, and L. C. Bassett, "Magnetic-field-dependent quantum emission in hexagonal boron nitride at room temperature," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[32] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, *et al.*, "Qiskit: An open-source framework for quantum computing," *Accessed on: Mar*, vol. 16, 2019.

[33] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, "Transfer learning in hybrid classical-quantum neural networks," *Quantum*, vol. 4, p. 340, 2020.

[34] A. Fischer and C. Igel, "Training restricted boltzmann machines: An introduction," *Pattern Recognition*, vol. 47, no. 1, pp. 25–39, 2014.

[35] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning.," in *Aistats*, Citeseer, vol. 10, 2005, pp. 33–40.

[36] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum deep learning," *Quantum Information & Computation*, vol. 16, no. 7-8, pp. 541–587, 2016.

[37] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[38] F. T. Cerasoli, K. Sherbert, J. Sławińska, and M. B. Nardelli, "Quantum computation of silicon electronic band structure," *Physical Chemistry Chemical Physics*, vol. 22, no. 38, pp. 21 816–21 822, 2020.

[39] S. Oh, "Quantum computational method of finding the ground-state energy and expectation values," *Phys. Rev. A*, vol. 77, p. 012 326, 1 Jan. 2008. DOI: 10.1103/PhysRevA.77.012326. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.77.012326.

3. QUANTUM MACHINE LEARNING FOR EIGENSTATE FILTRATION

The contents of this chapter are adapted from the article 'Sajjan, Manas, Shree Hari Sureshbabu, and Sabre Kais. "Quantum machine-learning for eigenstate filtration in two-dimensional materials." Journal of the American Chemical Society 143.44 (2021): 18426-18445'.

3.1 Introduction

Machine learning concerned with identifying and utilizing patterns within a data set has gained tremendous importance within the last decade. Even though the germinal idea can be traced back to the 1950s [1], it is safe to say that the domain has become a pioneering field of research within the last few years due to escalation in computational prowess and data availability, and have metamorphosed several disciplines including autonomous driving [2], image-recognition[3], speech recognition[4], natural language processing [5], computer games [6], and even refugee integration[7]. Consequently the integration of the technique in solving problems of physico-chemical interest [8] have also been explored with remarkable success whether in predicting ground-state density functionals[9], [10], self-energy in Dynamical Mean-Field Theory (DMFT) for the Anderson model[11], atomistic potentials and forcefields for molecular dynamics[12], [13] or unsupervised learning of phases of the 2D-Ising Hamiltonian[14]. Similar advancements have also been made in the field of Deep Learning[15] and Artificial Neural Networks (ANN) which has been successfully used to learn phase transition parameters[16], [17] or in quantum phase recognition [18]. Among the various architectures in this category, Restricted Boltzmann machine (RBM) based generative models being a universally powerful approximator for any probability density[19], [20] have particularly gained attention. RBMs have been successfully used to reconstruct quantum states in tomography from measurement statistics[21]. Carleo and Troyer showed how a neural network encoding a shallow RBM ansatz requires fewer parameters than certain kinds of matrix product states and can predict the ground state energy and unitary dynamical evolution of simple spin models with high accuracy [22].

However, all the algorithms discussed above have trained machine learning or deep-learning models on a classical computer to effectively recreate either a quantum state or its essential features. The past decade has also witnessed unprecedented development in quantum computing as a new paradigm which is fundamentally different than its classical counterpart in processing and storing data and performing logical operations[23] harnessing the power of quantum superposition and non-classical correlations like entanglement. A natural question that has spawned is whether such quantum machines can interpret and produce statistical patterns in data which are either difficult for classical machine learning algorithms or the performance of machine learning algorithms on quantum computer can outperform the classical variants in efficiency [24]. This has naturally motivated the development of a host of quantum -machine learning algorithms like Quantum Principal Component Analysis (PCA) [25], Quantum Support Vector Machines (QSVM)[26], Quantum Reinforcement Learning [27], quantum supervised and unsupervised learning [28], kernel design for Gaussian processes[29], Gaussian process regression[30], quantum classifier[31] or a plethora of linear algebra routines like HHL[32], QSVD[33], qBLAS[34] which forms the backbone of the quantum versions of many other machine learning algorithms. Each of these methods has reported theoretical speedup over the best-known classical algorithm under certain specific circumstances [35]. Similar investigations have also been undertaken for artificial neural networks to discover any unforeseen quantum advantage. For instance, Amin and co-workers have demonstrated a Quantum Boltzmann Machine [36] by adding an off-diagonal transverse field to the training model thereby making it more expressive to treat larger classes of problems [37]. Weibe *et al* have shown how sampling from a Gibbs distribution as is required for training an RBM can be distinctly accelerated using a quantum processor[38].

Motivated by such recent developments, Xia and Kais [39] proposed an actual quantum circuit using polynomial resources to correctly learn the amplitude of the RBM ansatz encoded within a neural network representing the state of a quantum system. The work also extended the neural network to three layers to learn the sign of the various components of the encoded wavefunction. The algorithm was benchmarked by showing the evaluation of ground states on simple molecular systems like H_2, LiH, etc thereby formally extending the efforts mentioned above to actual electronic structure calculations which are considered

to be powerful applications of near-term quantum devices. Indeed, interesting algorithmic advances have been made recently that can capture both the ground and excited state of such electronic structure problems with good accuracy [40]–[46]. Kanno *et al.* [47] modified the above method to encompass the complex phase of each component of the wavefunction by adding an additional neuron to the third layer. However both the work simulated the performance of the algorithm for ground states only on noiseless classical devices. In fact, due to conditional dependence on the sequence of measurements of the ancilla register, straight-forward implementation of the algorithm on a present-day actual NISQ device is difficult.

3.2 Theory

Our objective is to develop an efficient algorithm to train a neural network to perform the following minimization in a d-dimensional space

$$\begin{aligned} &\min_{\forall \psi \in S} \ \langle \psi | \hat{H} | \psi \rangle \\ &S = \{ |x\rangle \ | \ \ \hat{O}|x\rangle = \omega |x\rangle \ \ \forall \, |x\rangle \in \mathbb{C}^d \} \end{aligned} \tag{3.1}$$

where $\hat{H} \in \mathbb{C}^{d \times d}$ is the hermitian Hamiltonian defining the problem. Similarly $\hat{O} \in \mathbb{C}^{d \times d}$ is the user-defined hermitian operator. ω is the eigenvalue (real-valued) of the operator $\hat{O}$ and $|x\rangle$ is the corresponding eigenvector. The set S is the collection of all such eigenvectors with a specific eigenvalue ω. The operators $\hat{O}$ which we shall discuss will generally have more than one element in set S due to degeneracy in the eigenspace labelled by ω. By construction, the form of the algorithm shall always normalize the state $|\psi\rangle$ and hence normalization as a further constraint is unnecessary. We will return to this point later. The primary goal of the network is to then encode a normalized state-vector $|\psi\rangle$ which is a formal solution to Eq.3.1. The corresponding state so obtained is from the eigenspace of $\hat{O}$ with eigenvalue ω. If several such choices exist, the network learns the one with minimum energy. To solve the quadratic minimization problem with quadratic constraint in Eq. 3.1 we will define a penalty procedure as

$$F(|\psi\rangle, \hat{H}, \hat{O}, \lambda) = \langle\psi|\hat{H}|\psi\rangle + \lambda\langle\psi|(\hat{O} - \omega)^2|\psi\rangle \tag{3.2}$$

where $\lambda \geq 0$ is the penalty parameter. We provide a formal and original proof of equivalence of Eq. 3.2 with respect to Eq. 3.1 based on the following Theorem.

Theorem 3.2.1. *Let* $\{\lambda_i\}_{i=1}^{\infty}$ *be a sequence in the penalty parameter such that* $\lambda_1 \leq \lambda_2 \leq \lambda_3 ..... \lambda_\infty \to \infty$. *Also let* $P = \{|\psi_i\rangle\}_{i=1}^{\infty}$ *such that* $\forall\ |\psi_i\rangle \in P$ *the following is true.*

$$|\psi_i\rangle = \arg\min_{\psi} \mathrm{F}(\lambda_i, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle) \tag{3.3}$$

In other words, P *is the set of minimizers for Eq. 3.2 for each penalty parameter* $\lambda \in \{\lambda_i\}_{i=1}^{\infty}$. *If* $|\psi^*\rangle \in P$ *is a limit-point of the convergent sequence* $\{\psi_i\}_{i=1}^{\infty}$ *in* P *i.e* $|\psi^*\rangle = \lim_{i\to\infty} |\psi_i\rangle$ *then* $|\psi^*\rangle \in S$

An original proof of Theorem 3.2.1 is in Appendix B based on the fact that both the 1st and 2nd term in Eq. 3.2 are quadratic forms. An intuitive explanation can be provided that would suffice to appreciate the discussion in this report. One can note that in the cost function defined in Eq. 3.2 the term $\langle\psi|\hat{H}|\psi\rangle$ imposes the minimization of energy as required in Eq. 3.1. The second term i.e. $\langle\psi|(\hat{O} - \omega)^2|\psi\rangle$ is the variance of the operator $\hat{O}$ with the mean being the eigenvalue ω and is non-negative by construction. For large values of the penalty parameter λ, the minimization of the overall cost function is afforded if the variance term is pinned to zero i.e. the state $|\psi^*\rangle$ so chosen is an eigenstate of the operator $\hat{O}$ with eigenvalue ω. The space of such states is defined by the set S in Eq. 3.1. If several such choices exist, the role of the 1st term kicks in to guarantee optimality in energy.

While penalized optimization schemes with cost function of the kind in Eq. 3.2 has been employed in classical algorithms like in Density Matrix Renormalization Group (DMRG)[48], in Quantum Monte-Carlo methods in the past [49] and even recently [50] are also beginning to gain attention in recent literature on quantum algorithms beyond quantum-machine learning i.e. in algorithms using Unitary-Coupled Cluster Ansatz (UCC) of variational quantum eigensolver (VQE) [51], yet a formal proof is lacking. Besides a more popular choice that has

been studied in some detail is constraining the average value of the operator $\langle\psi|\hat{O}|\psi\rangle$ [52], [53] with the required eigenvalue instead of penalizing the variance as in Eq. 3.2. However, this recent study [51] shows Eq. 3.2 is a better penalty procedure in terms of feasibility and final error than restraining the average without providing a formal proof of equivalence between Eq. 3.2 and Eq. 3.1. Ref[51] also implemented the same to target symmetry operators on molecular systems using UCC-VQE using Qulacs[54] which is an ideal simulator of a real quantum computer. However, in this report, we shall use Eq. 3.2 to develop and train a shallow neural network using a quantum machine learning algorithm with quadratic resource requirements in terms of the size of qubit register, number of gates and parameter counts. The ansatz which the neural network would encode for the quantum state $|\psi\rangle$ would correspond to a probability density represented by RBM. We benchmark our algorithm on important 2D periodic materials like transition metal di-chalcogenides (TMDCs) and show implementations not only on quantum simulators but on actual NISQ devices (IBM-Q). TMDCs have never been studied before using any quantum algorithm. In the next few sections, we shall show how to filter any specific state of these 2D materials using either symmetry operators of the Hamiltonian or user-defined constructions of operator $\hat{O}$ in a unified manner using the same algorithm. Such an attempt to the best of our knowledge is the first of its kind in QML as all previous reports have focused exclusively on targeting the ground state of the system alone [47], [55].

3.2.1 Filter for specific excited states

To target the first excited state of the system, one can use a user-defined operator ($\hat{O} = |g\rangle\langle g|, \omega = 0$) where $|g\rangle$ is the ground state of the system obtained by training the network in a previous computation with $\lambda = 0$ in Eq. 3.2. In essence, we require the neural-network to return a state-vector in the null space of operator $|g\rangle\langle g|$. Since the null-space is $d-1$ dimensional, the minimum energy criterion as enforced by the 1st term in Eq. 3.2 guarantees the first excited state. This method using the penalty program in Eq. 3.2 is formally equivalent to deflation technique if one recognizes the idempotency of $\hat{O} = |g\rangle\langle g|$. Deflation has been the cornerstone of many classical algorithms in the past for obtaining

excited states[56], [57] and even a quantum algorithm as well with UCC-VQE [58]. But the formal reduction of our penalty procedure to deflation in Eq. 3.2 based on Theorem 3.2.1 offers a slightly different perspective. Moreover as we shall see shortly, the penalty program in Eq. 3.2 is more general and can be used to sieve any state based on arbitrary operator $\hat{O}$. For higher excited states (say the t th) one can add similar terms to Eq. 3.2 with the set $\{\hat{O}_{\mathrm{i}}\}_{\mathrm{i=i}}^{t-1}$ which forms a set of commuting operators with progressively refined null-space. For the choice of the penalty parameter λ in Eq. 3.2, one can choose any number greater than the spectral range of the Hamiltonian $\hat{H}$ as that would always work. The spectral range can be computed from the knowledge of the ground state and $||\hat{H}||_2$.

3.2.2 Filter for arbitrary states using symmetry operators

Eq. 3.2 can be used to solve a more general problem with any symmetry operator of the system $\hat{O}$ (by definition such operators satisfy $[\hat{O}, \hat{H}] = 0$ and hence share the same eigenspace). The corresponding user-desired eigenvalue ω labels the symmetry sector (set S in Eq. 3.1). Unlike in the previous case in section 3.2.1, usual symmetry operators need not satisfy idempotency and hence relaxation to deflation is impossible. To demonstrate our point, here we shall use $\hat{O} = L^2$ where L^2 is the squared-orbital angular momentum operator, a symmetry for 2D materials. ω would be set to the desired eigenvalue of L^2 . We shall see that the network will always learn the lowest energy eigenstate correctly despite multiple-fold degeneracy. To sieve other states from the entire degenerate subspace one can use a combination filter of $\hat{O}_1 = L^2$ and $\hat{O}_2 = |v\rangle\langle v|$ where $|v\rangle$ is the lowest energy state in the symmetry subspace obtained from the RBM. The penalty parameter λ can be chosen using the prescription in [51].

3.3 Algorithm

3.3.1 The Model

In the early 1980s, Hopfield networks [59] defined a probability distribution over a set of random variables which is encoded within the nodes of a unidirected graph using the physical notion of energy of interaction between the nodes. Boltzmann machines (BM) are

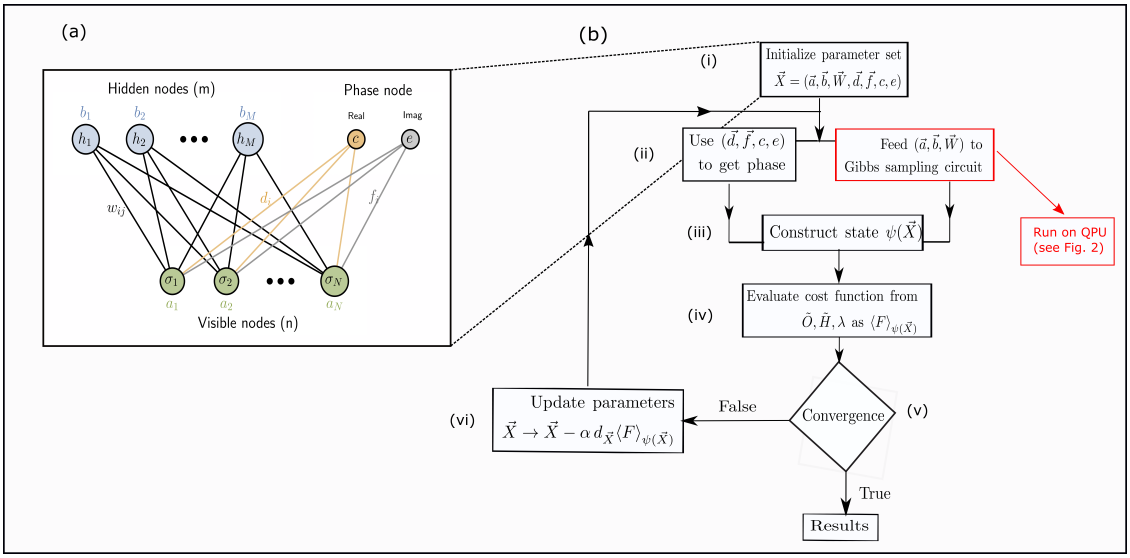

Figure 3.1. (a) The RBM architecture used in this work. The visible node contains n neurons (green), the hidden node has m neurons (blue) and the phase node contains 2 neurons, one to model the real part(orange) of the phase of the wavefunction and the other to model the imaginary part (grey). The weights and biases of the respective units are displayed. The RBM ansatz for the required state is defined from the Boltzmann distribution over the state-space of the visible-hidden units (b) The QML algorithm used to perform the variance penalized optimization. The part of step (ii) marked within the red box is performed on a quantum processor (QPU). All other steps are performed on a classical computer. Each step is marked with a Roman numeral. We follow each of these Roman numerals for discussing the algorithm in section 3.3.2

extensions of such a network that categorizes the node space into visible/physical layer and hidden/latent layers maintaining all to all connectivity[60]. Restricted Boltzmann Machine (RBM) [8], [20], [61]–[64] is a practically useful sub-category of BM which permits interaction only between the visible layer and hidden layer. The energy function used in the RBM model is thus that of a partially connected classical Ising network and the ansatz for the probability distribution is the corresponding thermal distribution. The ansatz is optimized to mimic the underlying probability distribution of the given data using free parameters called *weights* and *biases*[19], [62]–[67]. The goal of this paper is to use the RBM distribution to encode the amplitude field of an arbitrary quantum state $|\psi\rangle$ which is a solution to Eq. 3.1. Such neural-network quantum states (NQS) have been successfully employed in a variety of problems recently [8], [20], [22], [68] by training the *weights* and *biases* using a classical

computer. Herein we shall train the network by constructing the RBM distribution using a quantum circuit and discuss the quantum advantages.

The RBM network we use in this report consists specifically of three layers each having multiple neurons. The schematic of the network architecture is presented in Fig. 3.1(a). The first layer is the visible node consisting of n neurons, the second layer is the hidden node consisting of m neurons and the last layer is a phase node consisting of two neurons. While the n neurons are responsible for encoding the actual state, the purpose of the hidden neurons m is to add more controllable parameters to make the joint probability distribution (to be defined in Eq. 3.4 soon) more expressive and induce higher order correlation among n neurons [20]. Variables encoded by the visible node neurons (henceforth denoted by $\{\sigma_{\mathrm{i}}\}_{\mathrm{i}=1}^{n}$) and those by the hidden node neurons (henceforth denoted by $\{h_{\mathrm{j}}\}_{\mathrm{j}=1}^{m}$) are both binary random variables as σ_{i} and $h_{\mathrm{j}} \in \{1, -1\}$. As depicted in Fig. 3.1 (a), the bias vector of the visible neurons is denoted as $\vec{a} \in \mathbb{R}^{\mathrm{n}}$, bias vector of hidden neurons is denoted as $\vec{b} \in \mathbb{R}^{\mathrm{m}}$, the interconnecting weights of the visible and hidden neurons are denoted as $\vec{W} \in \mathbb{R}^{\mathrm{n}\times\mathrm{m}}$. The joint RBM distribution[8], [20], [62]–[64] $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ defined over the variables $(\vec{\sigma}, \vec{h})$ is

$$P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h}) = \frac{\mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}h_{\mathrm{j}}}}{\sum_{\{\sigma h\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}h_{\mathrm{j}}}} \tag{3.4}$$

For an electronic Hamiltonian with r spin-orbitals and N electrons, a Jordan Wigner mapping (JW) [69] would make $n = r$ or (or $n \approx O(log_2(r))$ for Bravi-Kitaev mapping [69]). However, it is well understood now that qubit requirements can be tapered by using additional symmetries like Z_2 [70]. Chemically inspired process of reducing qubit cost like using an active space [71] (wherein number of physical qubits required is still r but logical qubits required are much less as some qubits have frozen occupation/eigenvalue with Z operator) or using point-group symmetry or angular-momentum symmetry of the required state[70], [72] are also being recently employed. Thus a direct relationship of n with r would depend on the specification of the mapping and tapering used. Whatever may be the method, if the final $\hat{H}$ matrix is $\mathbb{C}^{d\times d}$ (as used in Section 3.2) then it is safe to say that in our model $n = \log_2(\mathrm{d})$. The number of hidden units m in our model is user-defined (for almost all data in this chapter we have used $n = m$) but the hidden node density $\alpha = \frac{m}{n}$ can be tuned to

enhance the final accuracy desired. We shall return to this point later. Neurons in the phase node are always 2 in number.

The purpose of the neurons in the phase node is to account for complex values and capture the phase of the wavefunction [47] unlike in conventional two-layer RBM networks[21] which faithfully recovers only the amplitude. As shown in Fig.3.1(a), for the phase node, the biases are denoted by $\{c, \mathrm{e}\} \in \mathbb{R}^2$ where c is the bias for the neuron capturing the real part of the phase and e is the bias for the neuron encoding the corresponding imaginary part. The phase node shares interconnections with the visible node only and is defined by $\vec{d} \in \mathbb{R}^{\mathrm{n}}$ for the real part of the phase and $\vec{f} \in \mathbb{R}^{\mathrm{n}}$ for the associated imaginary part. The corresponding phase function for the quantum state $|\psi\rangle$ defined using these nodes is

$$s(\vec{d}, \vec{f}, c, \mathrm{e}, \vec{\sigma}) = \tanh\left[(c + \sum_{\mathrm{i}} d_{\mathrm{i}}\sigma_{\mathrm{i}}) + \mathrm{i}(\mathrm{e} + \sum_{\mathrm{i}} f_{\mathrm{i}}\sigma_{\mathrm{i}})\right] \tag{3.5}$$

Together the set $\vec{X} = (\vec{a}, \vec{b}, \vec{W}, \vec{d}, \vec{f}, c, \mathrm{e})$ thus defines the complete set of trainable parameters of the model which the network shall learn iteratively to mimic the coefficients of the quantum state $|\psi\rangle$ in the chosen basis. We shall discuss the algorithm to do this in the next section.

3.3.2 Outline of the Method

The entire algorithm is schematically depicted in Fig. 3.1(b). It goes as follows.

(i) The first step is to initialize the parameters in the parameter vector $\vec{X} = (\vec{a}, \vec{b}, \vec{W}, \vec{d}, \vec{f}, c, \mathrm{e})$ on a classical computer. All parameters are randomly initialized in the parameter range [-0.02, 0.02] to avoid the vanishing gradient of the activation function for the phase node [39]. Sometimes if random initialization returns a poorly converged result, we use the initial parameter set of a converged point in a similar problem as the starting guess, a process known as warm optimization.

(ii) In the second step the set $(\vec{a}, \vec{b}, \vec{W})$ is fed into a quantum circuit for Gibbs sampling shown in Fig. 3.2. This step is performed on a quantum computer. The circuit

requires $n+m$ qubits to encode the visible node and the hidden node respectively and additionally $m \times n$ ancillary qubits. The entire register is initialized to $|0\rangle$. The purpose of the circuit is to sample a bit string $(\vec{\sigma}, \vec{h}) \in \{1, -1\}^{m+n}$ from the RBM distribution $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ defined in before in Eq. 3.4 [55]. In reality the circuit actually draws a sample $(\vec{\sigma}, \vec{h})$ from

$$Q(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h}) = \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j + \sum_{ij} w_{ij} \sigma_i h_j)}}{\sum_{\{\sigma h\}} e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j + \sum_{ij} w_{ij} \sigma_i h_j)}} \tag{3.6}$$

and then reconstruct $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h}) \propto Q(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})^k$. The real-valued parameter k will be discussed shortly.

The state of the visible node qubits and hidden node qubits are denoted henceforth as $|\sigma_i\rangle$ and $|h_j\rangle$ respectively. Note when σ_i (or h_j) $= -1$, $|\sigma_i\rangle$ (or $|h_j\rangle$) $= |0\rangle$ and $|1\rangle$ otherwise. In the circuit shown in Fig. 3.2 the single-qubit R_y gates acting only on the visible and hidden units have rotation angles parameterized by $(\vec{a}, \vec{b})$ and are responsible for creating the non-interacting part of the distribution in $Q(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ while the interaction terms $\{\sum_{i,j} w_{ij} \sigma_i h_j\}$ are turned on through using $C-C-R_y$ gates acting on ancilla register as the target. The rotation angles of these doubly-controlled R_y gates are parameterized by $\vec{W}$ and are different for different configurations of the control qubits (always 1 hidden and 1 visible). Various such choices can be realized by using X gates as shown in Fig. 3.2. After all such operations, we measure all the $(m+n+m \times n)$ qubits and post-select the results wherein the ancilla qubits have collapsed to state $|1111...1_{mn}\rangle$ only. We show that the probability of such a successful event has a generic lower bound determinable in terms of the parameters of the network $(\vec{a}, \vec{b}, \vec{W})$ (for details of the derivation of the generic bound refer to Appendix B). This master lower bound generalizes the previously noted one[39] as a special case. The role of the real-valued parameter k kicks in here. It serves as a regulator and is chosen in simulation to make the aforesaid lower bound a constant value (see Appendix B). After the post-selection, the corresponding states of the visible and hidden units are equivalent to all possible bit strings sampled from the distribution $Q(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ from which the

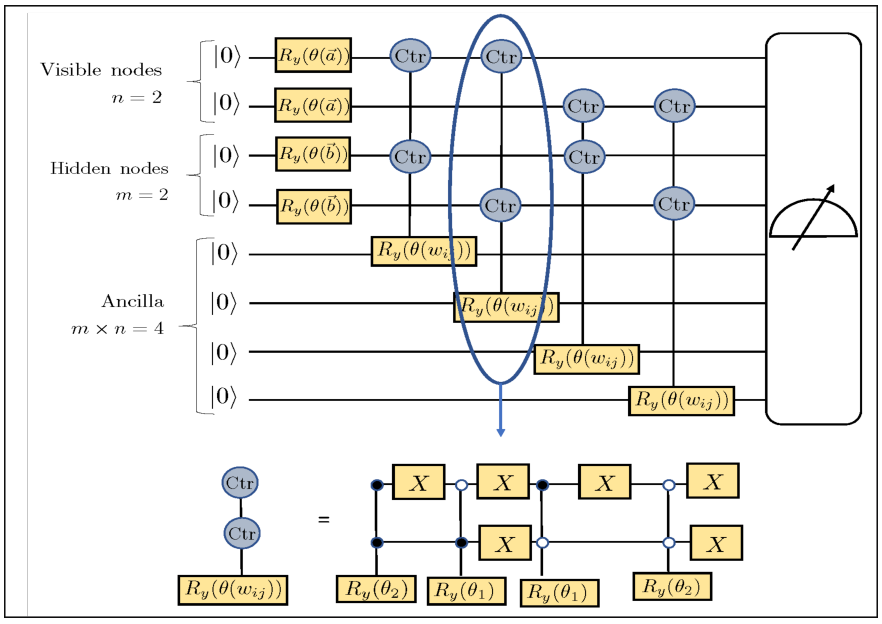

Figure 3.2. The Gibbs sampling quantum circuit used to create the Boltzmann distribution in Fig. 3.1(b) (highlighted within the red box in Fig. 3.1(b) step(ii)) for the case of $n = m = 2$. The circuit contains single-qubit R_y gates parameterized by biases $(\vec{a}, \vec{b})$ of hidden and visible neurons and $C - C - R_y$ gates parameterized by weights $\vec{W}$ between the hidden and visible neurons. Each $C - C - Ry$ gate is conditioned to rotate by different angles θ_1 and θ_2 for different choices of configurations of the control qubits. This can be implemented by use of X gates as illustrated at the bottom. The open circles show a node in state $|0\rangle$ and the closed circles show a node in state $|1\rangle$. At the end of the circuit all qubits are measured and configurations wherein the ancilla qubits are all in state $|1\rangle$ are post-selected (see text for details). For $(n + m)$ visible and hidden neurons, there will be $(n + m)$ visible and hidden qubits and also $(n + m)$ single R_y gates as there are that many biases. However since the $C - C - R_y$ gates are always controlled by 1 visible and 1 hidden qubit, there will be $m \times n$ such possibilities each of which targets one ancilla thereby making the size of the ancilla register $m \times n$. Thus there will be $O(n \times m)$ gates and number of qubits in the circuit. We discuss this further in section 3.3.3.

desired distribution $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ is constructed. The primary quantum advantage in our algorithm comes at this step where the full RBM distribution is constructed. Indeed we shall elaborate in Section 3.3.3, that there exist no polynomial-time classical algorithms for the construction of full RBM distribution. In our case, we can access the full distribution using quadratic resources by leveraging a quantum computer. The physical reason for this advantage is rooted in quantum parallelism which before a projective measurement allows the general state of the $(m + n + m \times n)$ qubits to be a

superposition of all possible bit-strings with the coefficients sampled from the full RBM distribution. Many such measurements are necessary to construct the RBM distribution encoding the target state as post-measurement we can retrieve only one such bit-string. As explained above, the k parameter in our model is useful here as it can be adaptively chosen by the user to control the measurement statistics (see Appendix B). Besides, for all systems primarily treated in this manuscript, we shall show that the chances of the ancilla register collapsing in the favorable state are naturally high even for modest values of the k parameter. (see Appendix B) With $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ constructed, one can now compute the marginal distribution over the state space of the visible units only as $\tilde{p}(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma})$ where $\tilde{p}(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}) = \sum_h P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$. Now $\sqrt{\tilde{p}(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma})}$ defines the amplitude of wavefunction over basis states of the visible units i.e. $|\sigma_1\sigma_2....\sigma_n\rangle$. The phase of each component of the wavefunction is now constructed classically using $(\vec{d}, \vec{f}, c, \mathrm{e})$ and *tanh* activation of neurons in the phase node as defined before in Eq. 3.5

(iii) With the two information from step (ii), the target wavefunction can now be constructed classically as

$$\psi(\vec{X}) = \sum_{\sigma} \sqrt{\tilde{p}(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma})} s(\vec{d}, \vec{f}, c, \mathrm{e}, \vec{\sigma}) |\sigma_1\sigma_2...\sigma_n\rangle \tag{3.7}$$

(iv) With the wavefunction, the cost function in Eq. 3.2 can now be constructed classically with the $(\hat{H}, \hat{O}, \lambda)$ from the user where $\hat{H}$ and $\hat{O}$ are the Hamiltonian and filter operator for the system being investigated respectively and λ is the penalty parameter.

(v) The next step is to check for convergence criterion or maximum number of iterations (to be discussed later). If either of the criterion is satisfied, results are printed

(vi) If either of the criterion from the previous step is not satisfied then the parameter set $\vec{X}$ is updated using steepest - descent algorithm with a learning rate (set to 0.005 in all our calculations). The updated parameter vector $\vec{X}$ is fed into step (ii) for the next iteration of the algorithm. We have also used the ADAM optimizer[73] but there is no significant change in convergence for the systems treated in this report. It must be

emphasized that unlike in classical supervised deep learning models, the learning of our network does not require prior training against a pre-assigned labeled data-set. The network learns the target eigenstate directly through minimization of the cost function (see Eq. 3.2) using the optimizer of choice (gradient descent in this case).

3.3.3 Resource Requirements

The power of an RBM ansatz even though underutilized in material science is beginning to gain attention in many areas of fermionic and bosonic physics [20], [68], [74]. Using n visible neurons and m hidden neurons, a recent study [75] has shown explicitly how a shallow RBM ansatz ($\alpha = \frac{m}{n} = 1$) like ours already captures several orders of perturbation theory and is a good approximant to the exact state. Classically, constructing such a full RBM distribution will require tracking amplitudes from a 2^{m+n} dimensional state space and hence has exponential resource requirements in preparation. Ref[76] formalizes and consolidates this statement by proving that a polynomial-time algorithm for classically simulating or constructing a full RBM distribution is not only absent now but is unlikely to exist even in the future as long as the polynomial hierarchy remains uncollapsed. However, such analysis does not preclude the existence of efficient quantum algorithms such as the one considered in this work. The quantum circuit in our algorithm (see Fig. 3.2) uses $m+n+m\times n$ qubits only for constructing the state indicating an $O(m\times n)$ scaling in qubit resource which if expressed in terms of hidden node density $\alpha = \frac{m}{n}$ is $O(\alpha n^2)$ The gate-set comprising single-qubit R_y gates scales as $m+n$ too, one for each of the bias terms $(\vec{a}, \vec{b})$ of the visible and hidden node qubits. Each $C-C-R_y$ gate in the circuit mediate a single interaction term within the $\vec{W}$ matrix between a spin of the visible layer σ_{i} and a spin of the hidden layer h_{j}. Since there are $m\times n$ such terms, the number of $C-C-R_y$ gates are $m\times n$ too, with the targets being each qubit in the ancilla register. Toggling between the various configurations of the control qubits (1 visible + 1 hidden) would require 6 R_x gates additionally in each $C-C-R_y$ (see Fig. 3.2) and hence the total number of such R_x gates is $6mn$. This indicates the total gate requirements of our sampling circuit is also $O(m\times n)$ which is equivalent to $O(\alpha n^2)$. The number of variational parameters in our algorithm for amplitude encoding using RBM is

$m+n$ for the biases of the two nodes and $m \times n$ for the $\vec{W}$ matrix. For the phase encoding, the variables are two n-dimensional vectors $(\vec{d}, \vec{f})$ and two scalars (c, e). Thus the total number of variational parameters is $m \times n + m + 3n + 2 = \alpha n^2 + \alpha n + 3n + 2$ which is also quadratic. The upshot is then, our algorithm for an RBM ansatz uses $O(\alpha n^2)$ qubits (circuit width), gate-set (circuit depth), and variational parameters to encode any arbitrary quantum state of n qubits in a $d = 2^n$-dimensional Hilbert space. Removing redundancy in global phase and normalization, a general such state would require $2(2^n - 1)$ parameters. One must know, in the RBM construction circuit no specific structure or sparsity has been assumed in the $\vec{W}$ matrix which if present may lower the requirements further. Quantum advantages have also been observed in supervised learning using the RBM distribution [38]. The study indicated that for the data-set of size N, a quantum circuit with amplitude amplification reduces the complexity of the algorithm from the conventional $O(N)$ to $O(\sqrt{N})$, a quadratic boost. It must also be emphasized that all the results in this manuscript are primarily treated for the case of $\alpha = 1$ as that suffices for the description of the system we study. We show how the results change for changing hidden node density α in Appendix B. Even though $\alpha = 1$ is good for systems in this report, for the case where the state is highly entangled, the user may be required to enhance the hidden node density as that increases the number of variational parameters and make the ansatz more expressive [20]. That may also be the case for molecular systems under geometric distortion wherein multi-reference correlation is important (we explore this point briefly in Appendix B). In this work all our results are compared against exact diagonalization as it affords the best accuracy in a given basis. The exact diagonalization results are obtained using 'Numpy' package[77] in python 3.0 with LAPACK routine.

3.3.4 Implementation Methods

We implement the algorithm in three flavors of computation. The first flavor henceforth designated as **'RBM-cl'** involves implementing the entire gate set of the Gibbs sampling circuit on a classical computer. This computation returns to us the exact state after the termination of the circuit. The second flavor is henceforth designated as **'RBM-qasm'**.

This has been implemented by simulating the Gibbs sampling circuit using Qiskit which stands for IBM's Quantum Information Software Kit (Qiskit) [78]. We specifically used the *qasm_simulator* at Aer provider (hence the name **RBM-qasm**) which is a quantum computer simulator and hence can mimic calculations performed on a noisy-intermediate scale quantum computing device even using a classical computer with options to incorporate customizable noise models. Unlike in **'RBM-cl'** where the exact state is returned, in **'RBM-qasm'**, the Gibbs sampling circuit in Fig. 3.2 is interrogated multiple times to build measurement statistics. From the observed bit-strings, the measurement probabilities $P(\vec{a}, \vec{b}, \vec{W}, \vec{\sigma}, \vec{h})$ are computed and hence the results are subjected to statistical fluctuations due to finite sampling errors. No noise model was used during the simulation in **'RBM-qasm'**. Finally to see the effect of noise we also investigated the performance of our algorithm on real IBM-Q quantum computers using the Qiskit interface. We used IBM-Q Sydney[79] and IBM-Q Toronto[80] interchangeably both of which are 27 qubit machines and hence suitable for our case studies. Calculations of this flavor are henceforth referred to as **'RBM-IBMQ'**. To reduce the effect of noise on the sampling probabilities we employ Measurement Error Mitigation (MEM) [81] directly implementable on Qiskit. We show in this report that MEM alone guarantees smooth and clear self-convergence in training (see Appendix B). The final accuracy of the results is affected by both MEM and warm-starting. We have seen without warm-starting convergence can not only be slow but sometimes the network can even be trapped in a local minima. It is in general difficult to assess apriori when the need for warm-starting can arise without a knowledge of the optimization surface as the objective function being optimized for the amplitude and the phase are non-convex in the arguments (see Eq.3.4 and Eq.3.5). It has been noted that the algorithm converges better without the need for warm-starting near optima (symmetry points for the system being treated in this report as discussed later). For the **'RBM-qasm'** and **'RBM-cl'** simulations, the maximum number of iterations within which well-converged results to be discussed below were obtained is $\leq$ 30,000 either with a warm-start or randomly initialized parameter set depending on the case. The **'RBM-IBMQ'** simulations were performed by breaking into two sessions/runs with the maximum iteration $\leq$ 700 for each session to reduce the job queue . Normally most calculations converged well before 700 iterations were reached within the first run as

warm-starting and MEM has been used as described above. For the few that did not, the final parameter set of the first run is punched for initializing the second session to ensure one continuous run. It must be emphasized that the entire code-base for training the network is home-built in Python 3.0 using standard packages like Numpy [77]. As mentioned before, we have extensively used Qiskit though as an interface to communicate with the IBMQ hardware and with *qasm_simulator*.

3.4 Results and Discussion

As a test of our method, we target state filtration of energy eigenstates of two well-established transition-metal dichalcogenides (TMDCs) - monolayer Molybdenum di-Sulfide (MoS_2) and monolayer Tungsten di-Sulfide (WS_2). Monolayer TMDCs have so far eluded attention in quantum simulations even though it is imperative to study their electronic structures to understand novel properties[82], [83] like high carrier mobility, high photoluminescence due to the direct band-gap, lack of inversion symmetry leading to large spin-orbit coupling and intra-valley transport etc. Indeed such features have made them attractive candidates for applications in Field-Effect Transistors[84], supercapacitors[85], spintronics[86], opto-electronics[87], [88], valleytronics[89]. We first show how the entire conduction band (CB) in such materials can be simulated using an appropriate choice of operator $\hat{O}$ as the ground state projector as discussed before and then later show how to 'sieve' eigenstates based on angular momentum symmetry. In all cases, we implement our algorithm on three flavors of RBM calculations - RBM-cl, RBM-qasm, RBM-IBMQ as discussed.

3.4.1 Filter for target excited states - Simulation of low energy bands in MoS_2 and WS_2 and effect of Spin-Orbit Coupling

The geometrical structure of monolayer TMDCs like MoS_2 or WS_2 indicates the presence of a trigonal prismatic real space unit cell [83] with D_{3h} point group symmetry as shown in Fig. 3.3. The transition metal is at the centre and the sulfur atoms are at the six corners of the triangular prism (see Fig. 3.3(b)). Consequently, the best orbital decomposition to evaluate the band structure of this periodic material should involve not only the s, p, d-

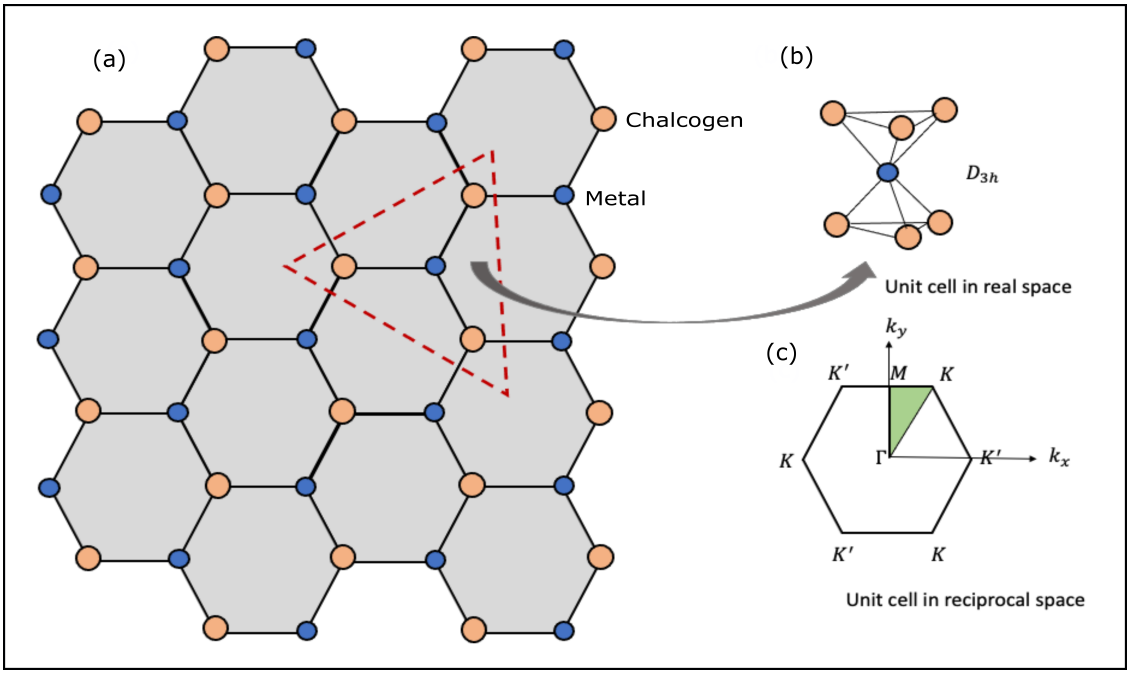

Figure 3.3. (a) The top view of the TMDC monolayer as studied in this report. The orange atoms are a chalcogen whereas the blue atoms are the metal centre. (b) The real-space trigonal prismatic unit cell highlighting D_{3h} symmetry. This shows that in the TMDC monolayer unlike in graphene, the constituent atoms have a non-coplanar arrangement. (c) The unit cell in reciprocal space showing the important symmetry points (Γ, K, M, K'). We shall investigate the energy and other properties within the sector marked in green following the usual $\Gamma - K - M - \Gamma$ path as in [90]. The co-ordinates of the symmetry points as (k_x, k_y) are : $\Gamma = (0,0), K = (\frac{4\pi}{3a_0}, 0), M = (\frac{\pi}{a_0}, \frac{\pi}{\sqrt{3}a_0})$ where a_0 is the metal-chalcogen bond length. For systems studied in this report the metal centre is Mo, W and the chalcogen is S

orbitals of the central metal atom but also of the surrounding sulfur atoms. Indeed several reports exist which treats the electronic structure of such materials using a tight-binding description obtainable from a 5 band, 7-band or an 11-band model using varying degree of inclusion of the orbital set of the metal and the chalcogen[91]–[94]. However, recently a 3-band parameterization has been demonstrated to yield remarkable accuracy in energy over the entire Brillouin zone[90]. A tight binding Hamiltonian in this description is obtained by fitting the energy curves against DFT calculations (with GGA and LDA functionals) employing the d_{z^2}, d_{xy} and $d_{x^2-y^2}$ orbitals of the metal centre[90] only. This choice is based on the fact that for trigonal prismatic coordination, the d-orbital set of the metal splits into three groups-A_1' containing d_{z^2} orbital only, E' containing $d_{xy}, d_{x^2-y^2}$ and E'' containing d_{xz}, d_{yz} orbitals. However, reflection symmetry of D_{3h} restricts inter-coupling between the

orbitals of E'' set with the remaining two groups. Indeed E'' contributes exclusively to higher energy bands and has no role to play in the low-energy physics of the valence and conduction band which is considered in this work. The absence of chalcogen p-orbitals is definitely an approximation albeit a good one as seen from Ref [90]. We shall return to this point shortly.

We use a tight-binding model comprising of third-nearest neighbor (TNN) metal-metal hopping[90] of the aforesaid three band Hamiltonian for all our calculations henceforth. The parameters of the model are obtained from the more accurate GGA calculation set [90]. Appendix B enlists details of the Hamiltonian and parameters for completeness and brevity. Our working Hamiltonian, for both the systems are thus a 3×3 Hermitian matrix. For qubitization we convert it into a 4×4 Hermitian matrix by padding an additional 1×1 block with a diagonal entry chosen to be $\geq$ spectral range of $H_{3\times3}$ as that would keep the low-lying eigenvalue structure of the resultant matrix undisturbed for the training to successfully proceed. Thus for both the systems, our neural network comprises of a visible node with 2 neurons to encode the state, two hidden neurons and additional 2 neurons for the phase node too. For the Gibbs sampling circuit in Fig. 3.2, we thus need 2 qubits to represent the entire visible layer and 2 qubits for the hidden layer. In addition we need 4 ancilla qubits to serve as targets for $(C-C-R_y)$ rotation thereby requiring 8 qubits in total. For the circuit in Fig. 3.2, we use 4 single qubit Rotation gates (R_y), 4 Controlled-Controlled Rotation gates $(C-C-R_y)$, and also 24 Bit-flip (X) gates. The optimization in each case starts from a randomly initialized parameter set. In case if the accuracy is poor, we re-start the algorithm by feeding the initial parameter from the results of a nearby converged k-point as a warm start. We see the results are in excellent agreement with the exact diagonalization when a such a warm start is employed along with MEM as described before. For IBMQ implementation we have used 'IBM-Sydney' and 'IBM-Toronto' both of which are 27 qubit machines. To reduce the operational time on the actual quantum device for job queue and isolate the effect of gate-infidelity, IBMQ simulations for each k-point were often warm-started with an initial parameter set obtained from the initial parameters of the *qasm* simulation of a nearby but non-identical k-point.

The results from the algorithm using the cost function in Eq. 3.2 is displayed in Fig. 3.4 for MoS_2 and Fig. 3.5 for WS_2. In Fig. 3.4(a) we have overlayed the energies obtained

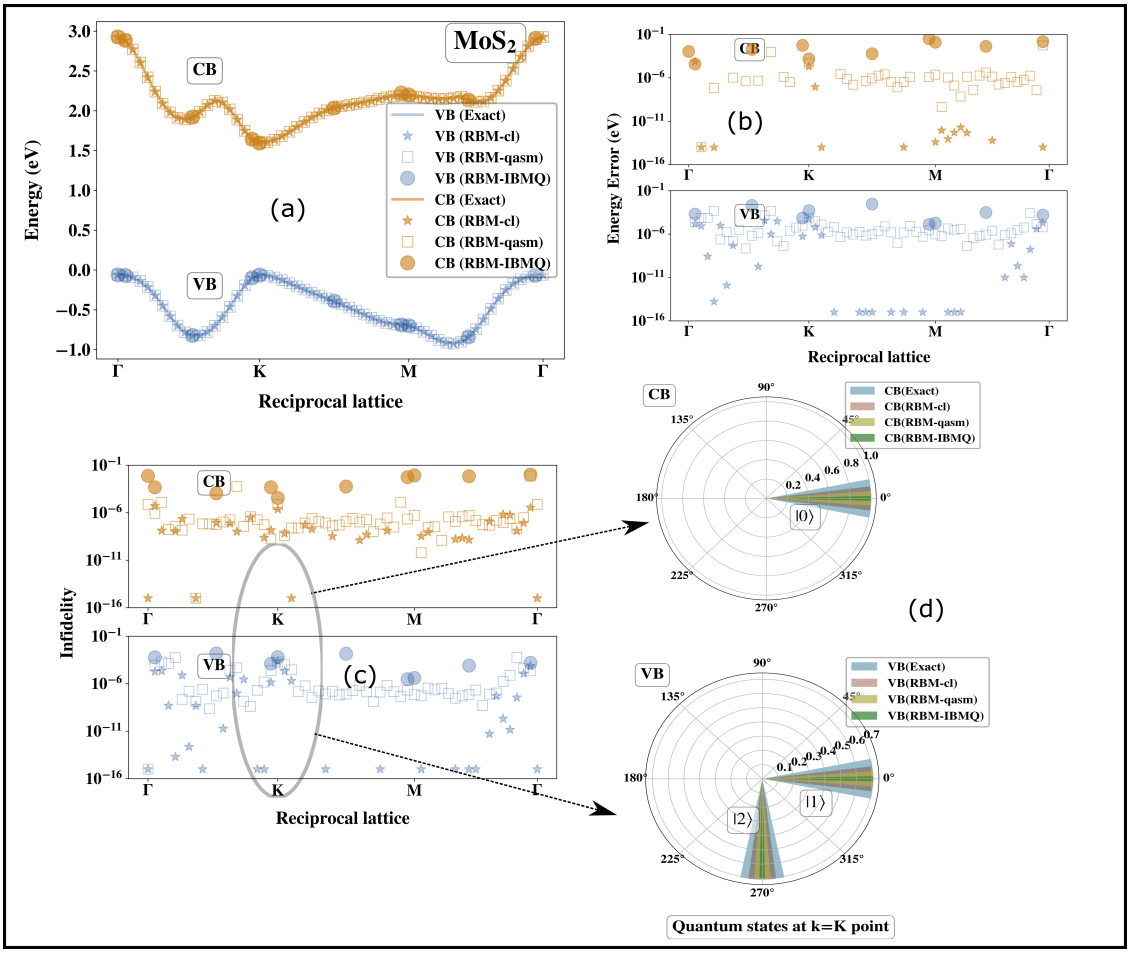

Figure 3.4. (a) Valence (VB) and conduction band (CB) of MoS_2 calculated using all flavors of RBM and overlayed against exact diagonalization. The valence band is simulated using $\lambda = 0$ in Eq. 3.2 and the conduction band using ($O = |v_0\rangle\langle v_0|$, $\omega = 0$, $\lambda = 5$) in Eq. 3.2 where $|v_0\rangle$ is the valence band state at each k-point. For IBMQ implementations we used 'IBM-Sydney' and 'IBM-Toronto'. All parameters are randomly initialized (see Fig. 3.2) or warm-started with the initial guess of a converged nearby k-point. (b) The corresponding energy errors from (a) in eV. (c) The corresponding state infidelities (1-Fid) where $Fid = |\langle\Psi_{\text{RBM}}|\Psi_{\text{Exact}}\rangle|^2$ (d) The orbital decomposition of the states at K-point where $|0\rangle = d_{z^2}$, $|1\rangle = d_{xy}$, $|2\rangle = d_{x^2-y^2}$. The states from RBM calculations matches well with those from exact diagonalization in phase and amplitude. The width for each bar is set differently for visual clarity.

from our algorithm as a function of the wave-vector index sampled from the Brillouin zone following the usual $\Gamma - K - M - \Gamma$ path (see Fig. 3.3(c)). The result for the valence band (VB) is denoted in blue and is obtained by setting $\lambda = 0$ in Eq. 3.2 which corresponds to the usual variational optimization to obtain the ground state at each k-point. The results for the conduction band (CB) are shown in orange in Fig. 3.4(a). They are thereafter computed

as a separate set of calculations using $O = |v_0\rangle\langle v_0|$ and $\omega = 0$ in the cost function in Eq. 3.2 where the corresponding ground state in the valence band (VB) is denoted as $|v_0\rangle$. The penalty parameter is $\lambda = 5$. The cost-function now samples a state orthogonal to ground state (null space of the projector $|v_0\rangle\langle v_0|$) for each of the k-points. The minimum energy criterion imposed by the first term in the cost function in Eq. 3.2 guarantees obtaining the next higher excited state which happens to be the state space in the conduction band.

We see for all flavors of our algorithm (RBM-cl, RBM-qasm, RBM-IBMQ) the simulated energy values for both the valence and the conduction band are in good agreement with the ones obtained from exact diagonalization. The corresponding errors in energy are displayed in Fig. 3.4(b) and are usually $\leq 10^{-4}$ eV for RBM-cl and RBM-qasm which are noiseless pristine implementations but is around $10^{-2} - 10^{-4}$ eV for the valence band (VB) and the conduction band for RBM-IBMQ indicating the worsening of performance due to faulty gate implementations in the Gibbs sampling circuit. Fig. 3.4(c) plots the state infidelities i.e. 1-Fid where $Fid = |\langle\Psi_{\text{RBM}}|\Psi_{\text{Exact}}\rangle|^2$. We see that the infidelities are also quite small for each band with the performance worsened only in the IBMQ variant of the RBM implementation.

Like Fig. 3.4(a), Fig. 3.5(a) displays the band structure of WS_2 wherein the energies for both the valence and conduction band are overlayed against the energy values obtained from exact diagonalization. All three flavors of RBM implementation yield reasonably accurate results as in the case for Fig. 3.4(a). Fig. 3.5(b) and Fig. 3.5(c) display the energy error and the state infidelities of the state obtained from the RBM calculations against exact diagonalization. The error ranges in each case is similar to what has been discussed for MoS_2.

Fig. 3.4(d) and Fig. 3.5(d) displays the orbital decomposition of the states in the conduction and valence band at the most important symmetry point i.e. the K- point. In our calculations qubit $|0\rangle = d_{z^2}$, $|1\rangle = d_{xy}$, $|2\rangle = d_{x^2-y^2}$ where $\{0, 1, 2\}$ are the integer equivalents of the two-qubit bit strings encoding the neurons of the visible node. We see from our calculations however that the exact state generated from the model lines up correctly against the RBM states in both amplitude and phase. While the state of the conduction band at K-point is exclusively populated by the d_{z^2}, that in the valence band is a superposition of d_{xy} and $d_{x^2-y^2}$ with a phase shift of $3\pi/2$. This is consistent with the orbital decomposition given

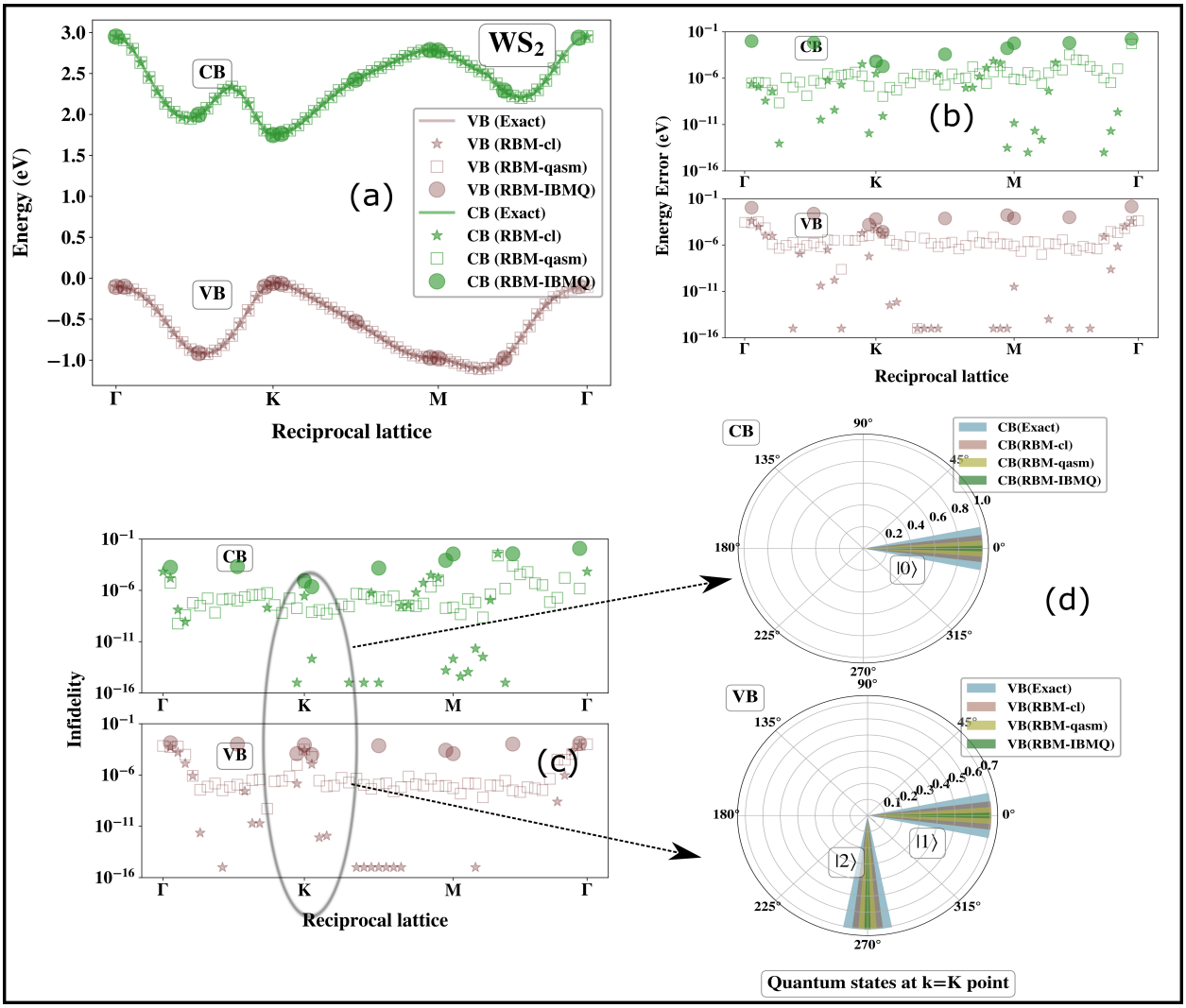

Figure 3.5. (a) Valence (VB) and conduction band (CB) of WS_2 calculated using all flavors of RBM and overlayed against exact diagonalization. The valence band is simulated using $\lambda = 0$ in Eq. 3.2 and the conduction band using ($O = |v_0\rangle\langle v_0|$, $\omega = 0$, $\lambda = 5$) in Eq. 3.2 where $|v_0\rangle$ is the valence band state at each k-point. For IBMQ implementations we used 'IBM-Sydney' and 'IBM-Toronto'. All parameters are randomly initialized (see Fig. 3.2) or warm-started with the initial guess of a converged nearby k-point. (b) The corresponding energy errors from (a) in eV. (c) The corresponding state infidelities (1-Fid) where $Fid = |\langle\Psi_{\text{RBM}}|\Psi_{\text{Exact}}\rangle|^2$ (d) The orbital decomposition of the states at K-point where $|0\rangle = d_{z^2}$, $|1\rangle = d_{xy}$, $|2\rangle = d_{x^2-y^2}$. The states from RBM calculations matches well with those from exact diagonalization in phase and amplitude. The width for each bar is set differently for visual clarity.

in Fig. 2 of Ref[90] and is partly the reason given by the authors to use this three orbitals for generating the tight-binding Hamiltonian as the model yields correct state description near the band-gap. However as is clear from Fig. 2 of Ref[90], the orbital composition of

the states at the Γ and M-point has contribution from the p-orbitals of S and s-orbitals of both the metal and the S atoms. This makes the three-band model an approximation for the exact character of the states even though it can replicate the energy very well throughout the Brillouin zone.

We further concentrate in this report on describing the low-energy physics near the K or K' valley for which, as mentioned before, the state-description of the three-band model suffices. We construct the Hamiltonian [95]–[98] near the K-valley in the basis of the states of the conduction band i.e. $|d_{z^2}\rangle$ (see Fig. 3.4(d) and Fig. 3.5(d)) and that of the valence band i.e. $\frac{1}{\sqrt{2}}(|d_{x^2-y^2}\rangle + \mathrm{i}|d_{xy}\rangle)$ (see Fig. 3.4(d) and Fig. 3.5(d)). The states at the K' valley are related to those at the K valley due to time-reversal symmetry[97], [99] and hence is ignored from further discussion. The Hamiltonian is:

$$\begin{aligned} H = &(\frac{E_c}{2} + \frac{\lambda_c s}{2})(\hat{I} - \hat{\sigma_z}) + (\frac{E_v}{2} + \frac{\lambda_v s}{2})(\hat{I} + \hat{\sigma_z}) \\ &+ \gamma k_x \hat{\sigma_x} + \gamma k_y \hat{\sigma_y} \\ &+ \frac{\alpha}{2}(k_x^2 + k_y^2)(\hat{I} + \hat{\sigma_z}) + \frac{\beta}{2}(k_x^2 + k_y^2)(\hat{I} - \hat{\sigma_z}) \\ &+ \kappa_{TW}(k_x + \mathrm{i}k_y)^2(\hat{\sigma_x} + \mathrm{i}\hat{\sigma_y}) + \kappa_{TW}(k_x - \mathrm{i}k_y)^2(\hat{\sigma_x} - \mathrm{i}\hat{\sigma_y}) \end{aligned} \tag{3.8}$$

A effective description such as Eq. 3.8 is often referred in literature as the two-band $k \cdot p$ model constructed using Lowdin-Partitioning[90], [96]. The first two terms in Eq. 3.8 is the massive term required to create the band-gap(Δ) in the material at the K-point. These terms are absent in graphene. In most reports this term is written as $\frac{\Delta}{2}\sigma_z$ with a symmetrically located origin but we choose to use the Ec and Ev values obtained from our calculations in Fig. 3.4 and Fig. 3.5. The additional summands in each of the 1st two terms (λ_v, λ_c) refer to band-splitting at thre always2 in numbere K-point due to spin-orbit coupling (SOC). In the three-band basis, SOC is entirely due to the L_z operator (more on this in the next section) contribution of which in the chosen basis of can be effectively modeled as the first two terms [95], [98], [100]. Unlike the Bloch state in the conduction band, the valence band is exclusively dominated by metal orbitals $|d_{x^2-y^2}\rangle$ and $|d_{xy}\rangle$ with non-zero angular momentum leading to strong splitting [90]. The spin-orbit splitting in the conduction band is weak[90],

[95], [99] and below the resolvable limit of NISQ devices and hence has been ignored herein i.e. $\lambda_c = 0$. The parameter $s \in \{1, -1\}$ is the spin index and labels the SOC split valence bands. The 3rd-6th term is the linear and quadratic extrapolation away from the K point and yields a spherically isotropic band surface. The 7th-8th terms (parameterized by κ_{TW}) break the isotropy and lead to the well-known effect of trigonal warping (TW). The warped band surfaces in these materials are a consequence of the presence of a perpendicular C_3 axis due to the D_{3h} symmetry of the associated real-space unit cells (see Fig. 3.3(b)). Further terms in [96] which removes anisotropy between valence and conduction band are ignored due to their small unresolvable contributions.

Since the Hamiltonian in Eq. 3.8 is 2×2, we require a single visible neuron to encode the eigenstates, a single hidden neuron consistent with $\alpha = 1$ and 1 additional ancillary qubit. The number of single-qubit R_y gates is 2 and the number of $C - C - R_y$ gates is 1 and 6 R_x gates. Calculations are performed using $\lambda = 0$ in Eq. 3.2 for the two SOC split valence bands with $s = \pm 1$ and ($\hat{O} = |\nu_0\rangle\langle\nu_0|$, $\lambda = 5, \omega = 0$) for the conduction band. For NISQ devices we use 'IBM-Sydney' and 'IBM-Toronto' interchangeably as before. All calculations are performed for (k_x, k_y) pairs centered at the K-point and with a cutoff $|k|$ of $0.1K$ point to probe the low-energy regime. Since the (k_x, k_y) pairs are near a symmetry point (K-point) warm starting was rarely observed to be required in RBM-cl and RBM-qasm but has been occasionally used in RBM-IBMQ for hastening convergence and reducing job queue. Each point on RBM-IBMQ are performed within a single run with Measurement Error Mitigation (MEM) as before for smooth self-convergence and consistency with other results. Parameters for warping are obtained from [96].

In Fig. 3.6(a), (b) and (c) we plot the exact 2D band surfaces obtained from Eq. 3.8 for the two SOC split valence bands (s=± 1) and the conduction band. The crosses in each plot refer to the (k_x, k_y) pairs wherein all flavors of RBM calculations have been performed. The results of such RBM calculations for each such pair are displayed as energy errors (eV) in 3.6 (d)-(f). The x-axis in each such plot is a flattened point index mapping (k_x, k_y) pairs to integers by starting from pairs closest to the K-valley at the origin and proceeding spirally outwards. In other words, for a given $|k|$ the flattened point index groups all (k_x, k_y) pairs

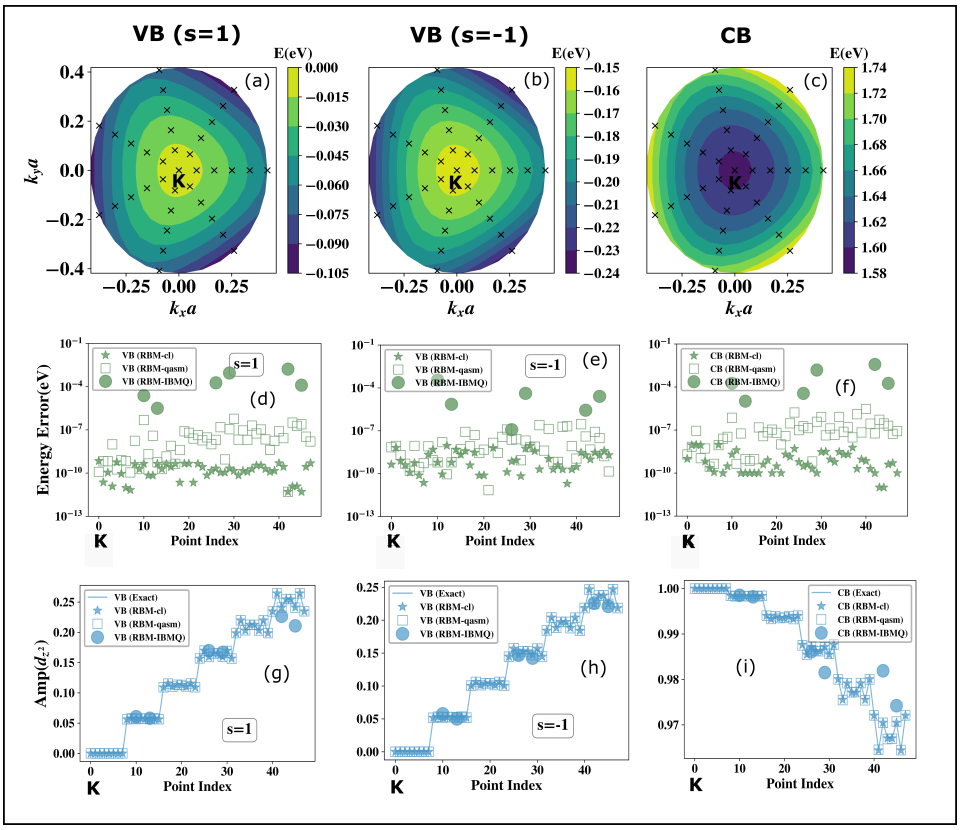

Figure 3.6. (a) The exact energy contours in valence band (VB) for s=1 within the three-band approximation for the Hamiltonian in Eq. 3.8 as a function of (k_x, k_y) near the K-point in MoS_2 (b) Same as in a) but for s=-1 (c) Same as in a) for the conduction band (CB). The crosses in (a), (b) and (c) denotes the (k_x, k_y) pair wherein calculations for all three flavors of RBM have been executed. (d) Energy errors in eV from three flavors of RBM calculations for points denoted as cross in a) for the valence band (s=1) case computed using $\lambda = 0$ in Eq. 3.2 in MoS_2. The x-axis is a flattened point index with (k_x, k_y) pairs marked as crosses in (a) mapped to integers such that the origin is at the K-point. From the K-point, the flattened point index scale moves spirally outwards grouping all (k_x, k_y) pairs satisfying $|k| = \sqrt{k_x^2 + k_y^2}$ as consecutive integers and then proceeding to the next $|k|$ (e) Same as in d) but with points denoted in b) as crosses for other valence band with s=-1 (f) Same as in d) but for points denoted in c) as crosses for the conduction band computed with $(\lambda = 5, \omega = 0, \hat{O} = |\nu_0\rangle\langle\nu_0|)$ in Eq. 3.2. (g) The amplitude for the occupancy of d_{z^2} orbital on the metal for states computed at (k_x, k_y) pairs near the K-point from all three flavors of RBM as well as the exact states in valence band (s=1) for MoS_2. The amplitude of states with the same $|k| = \sqrt{k_x^2 + k_y^2}$ appear bunched together as 'steps' due to flattened point-index scale used. Near the K-point the amplitude is the same for all such pairs within a given step due to isotropy of the energy surface. However away from the K-point deviations appear due to trigonal warping owing to the D_{3h} symmetry of the unit cells in TMDCs. The states from all flavors of RBM can resolve the influence of warping accurately with the performance worsened for the noisy variant. (h) Same as in g) for valence band (s=-1) (i) Same as in g) for conduction band.

satisfying $|k| = \sqrt{k_x^2 + k_y^2}$ as consecutive integers and then proceeds to the next $|k|$. We see that the energy error in each case is low for the RBM-cl and RBM-qasm variant ($\leq 10^{-4}$ eV) for all three bands and $\leq 10^{-2}$ eV for the IBMQ variant. Thus given the energy scale and extent of the splitting in the valence bands ($s = \pm 1$) in Fig. 3.6(a)-(b) and the scale of the energy errors in Fig. 3.6(d)-(e), it suffices to say that the performance of our algorithm is good enough to resolve band splitting due to features like spin-orbit coupling. To study the effect of warping parameters in Eq. 3.8 in the state, we plot in Fig. 3.6(g)-(i) the amplitude of the corresponding states in the basis of $|d_{z^2}\rangle$ for the two SOC split valence bands ($s = \pm 1$) and the conduction band. The x-axis in each case is the flattened pair index as in Fig. 3.6(d)-(f). At the K-point (origin), the conduction band is exclusively populated by $|d_{z^2}\rangle$ as discussed before but the reverse is true for the valence bands. In each of the plots Fig. 3.6 (g)-(i) all (k_x, k_y) pairs which satisfy $|k| = \sqrt{k_x^2 + k_y^2}$ are bunched together as 'steps' due to the flattened point index scale chosen. We see that near the K-point wherein the effect of warping is not prominent, all such points within a given 'step' (same $|k|$) share the same amplitude. However away from the K-point deviation starts to become predominant. The amplitudes computed from the states of all three variants of RBM calculations line up well against the exact curve with the IBMQ variant showing some deviations albeit small considering the y-scale in these plots. Our algorithm thus can successfully resolve finer features like trigonal warping too in these Bloch states. A similar panel for WS_2 is presented in Appendix B. Accurate computation of such Bloch states with these finer features preserved is necessary as momentum matrix elements between these states become important in simulating important properties of materials like optical conductivity[101], [102], electrical and thermal conductivity[103] etc.

3.4.2 Filter for arbitrary states using symmetry operators

In this section, we shall use the same set of TMDCs discussed above to explore how one can sieve arbitrary states based on symmetry constraints. To demonstrate the point we use orbital angular momentum symmetry. The L_z operator in the three-band approximation commutes with the Hamiltonian [90] in absence of spin-orbit coupling as has been considered

in this work. The operators L_x, L_y are essentially null matrices in the three-band basis of $\{d_{z^2}, d_{xy}, d_{x^2-y^2}\}$ as mentioned in [90]. Hence L^2 enjoys exclusive contribution from L_z and is a symmetry operator in the system. For computation, we use the Hamiltonian of the system at the K-point because the three-band approximation as discussed before is extremely accurate therein.

The complete set of eigenvalues and eigenstates of L_z and hence of L^2 operator is given in Appendix B. From the knowledge of the spectrum of L^2 operator we see that it has two distinct eigenvalues which are $\{0, 4\}$ in atomic units. One of the eigenvectors of the doubly-degenerate eigenspace with eigenvalue 4 is the state in the valence band and the other is a higher energy excited state above the conduction band (not shown in Fig. 3.4 or Fig. 3.5). Both these states are exclusively made from the contribution of $\{d_{xy}, d_{x^2-y^2}\}$ as seen from the state decomposition in Appendix B. The sector with eigenvalue 0 has single-fold degeneracy and is made from the excited state in the conduction band. As discussed before in Fig. 3.4(d) and Fig. 3.5(d) (also in Appendix B) this state is exclusively made from the contribution of the d_{z^2} which explains the absence of z-component angular momentum. We would thus expect that if we choose $\hat{O} = L^2$ and $\omega = \{0, 4\}$ in Eq. 3.2 for training the network, we should yield the excited state in the conduction band for $\omega = 0$ and should yield the ground state in the valence band for $\omega = 4$ as that is of lower energy (in compliance with the first term in Eq. 3.2) than the other degenerate eigenstate.

The qubit and gate resource requirements of this simulation are exactly the same as discussed in section 3.4.1 with 2 visible node neurons and 2 hidden node neurons for each of the two systems MoS_2 and WS_2. The Gibbs sampling circuit in Fig. 3.2 would need a total of 8 qubits as before (2 for visible node + 2 for hidden node + 4 ancillary qubits). The gate requirements for the circuit to reproduce the amplitude are thus 4 single qubit Rotation gates (R_y), 4 Controlled-Controlled Rotation gates ($C - C - R_y$) and also 24 Bit-flip (X) gates. We start the optimization with randomly initialized parameters.

In Fig. 3.7 we display the results of our simulation. Like before, the results from all three flavors of RBM (marked as 2 = RBM-cl, 3 = RBM-qasm and 4 = RBM-IBMQ) are compared against the exact expected state (marked as 1 = Exact). In Fig. 3.7(a), the results of energy in eV from the three RBM simulations and the exact one are displayed for

the eigenvalue sector $\omega = 0$ a.u. This happens to be the conduction band (CB) energy in Fig. 3.4(a). We find an extremely good agreement for all flavors of RBM with the exact value. The corresponding energy error is displayed in Fig. 3.7(c) and is in the range of $10^{-5} - 10^{-4}$ eV for RBM-cl and RBM-qasm but is within $10^{-4} - 10^{-3}$ eV for the RBM-IBMQ variant. Fig. 3.7(b) displays the constraint violation error i.e. how much the state encoded in the neural network after training has an $\langle L^2 \rangle$ equal to the target value of ω (in this case $\omega = 0$ a.u.). We see that the violations are quite small for the noiseless implementations. Even for implementation on actual NISQ devices of IBM-Q, it is close to 10^{-3} a.u.

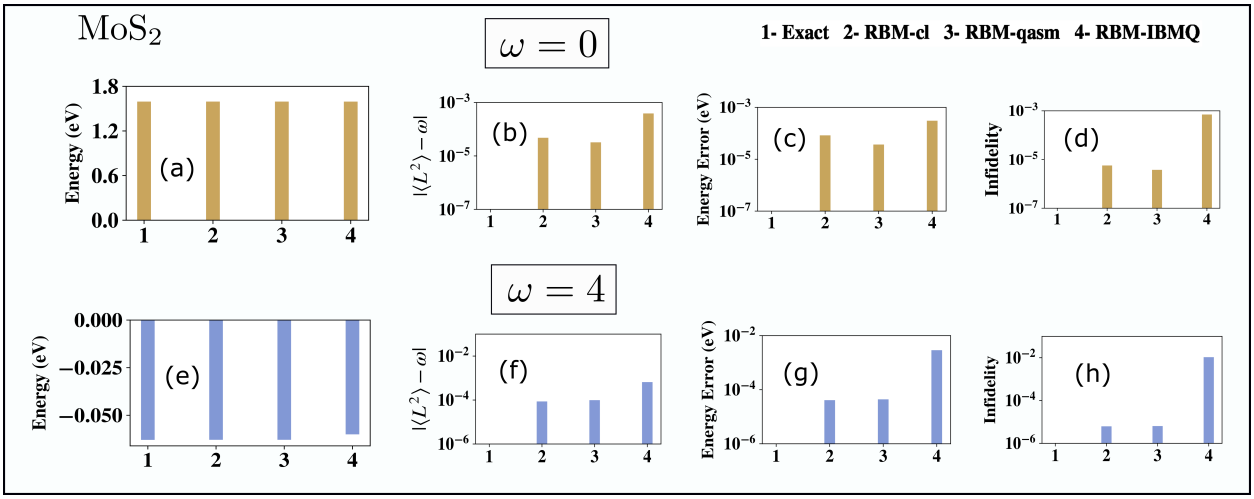

Figure 3.7. (a) The energy comparison between exact (1), RBM-cl (2), RBM-qasm (3), RBM-IBMQ (4) for computation with $\hat{O} = L^2$ and eigenvalue $\omega = 0.0$ a.u. in Eq. 3.2. The exact energy is 1.5950 eV and is the conduction band energy at K-point in MoS_2 shown in Fig. 3.4. (b) The constraint violation error $|\langle L^2 \rangle - \omega|$ of the state obtained from different flavors of RBM and the desired value ω. (c) The energy error in eV from (a) of the states obtained from RBM. (d) The state infidelities (1-Fid where $Fid = |\langle \Psi_{\text{RBM}} | \Psi_{\text{Exact}} \rangle|^2$) obtained from RBM and the exact one (e-h) corresponds to an equivalent set of plots as in (a-d) just described but with the other eigenspace of L^2 with eigenvalue $\omega = 4$ a.u. The exact energy here is the valence band energy at K-point for MoS_2 shown in Fig. 3.4 and is -0.0629 eV.

Fig. 3.7(c) displays the energy error and Fig. 3.7(d) displays the state infidelity error (1-Fid where $Fid = |\langle \Psi_{\text{RBM}} | \Psi_{\text{Exact}} \rangle|^2$). We see that for all flavors of RBM implementation the infidelities are quite small with the performance worsened for implementation on the actual IBM-Q device. Fig. 3.7(e-h) corresponds to similar plots as discussed above but this time in the other eigenvalue sector with $\omega = 4$ a.u. We again see that the energy values (in eV) in Fig. 3.7(e) matches with the exact for all flavors of RBM-implementation. This

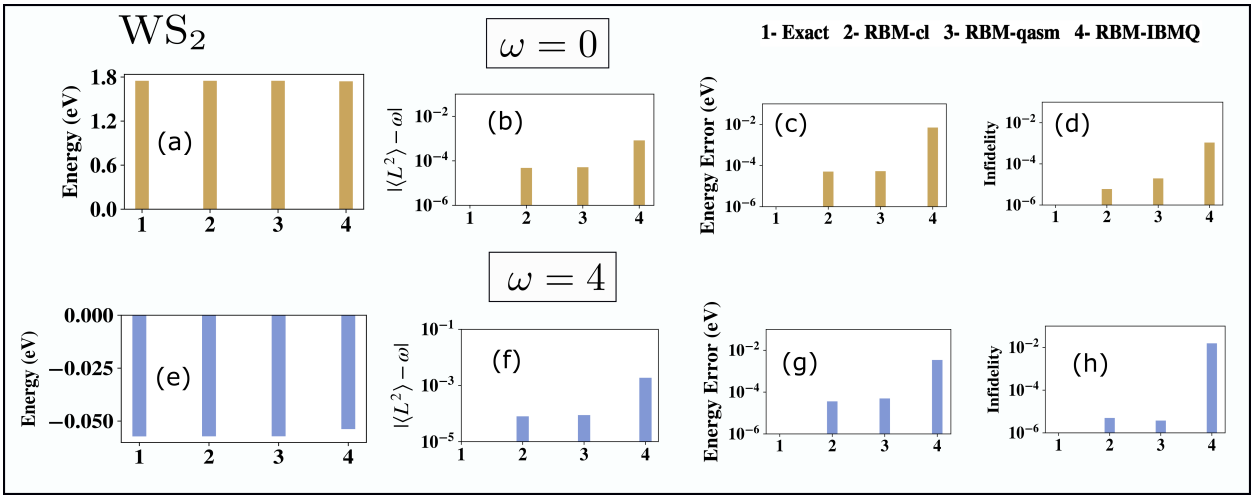

Figure 3.8. (a) The energy comparison between exact (1), RBM-cl (2), RBM-qasm (3), RBM-IBMQ (4) for computation with $\hat{O} = L^2$ and eigenvalue $\omega = 0.0$ a.u. in Eq. 3.2. The exact energy is 1.749 eV and is the conduction band energy at K-point in WS_2 shown in Fig. 3.5. (b) The constraint violation error $|\langle L^2\rangle - \omega|$ of the state obtained from different flavors of RBM and the desired value ω. (c) The energy error in eV from (a) of the states obtained from RBM. (d) The state infidelities (1-Fid where $Fid = |\langle\Psi_{\text{RBM}}|\Psi_{\text{Exact}}\rangle|^2$) obtained from RBM and the exact one (e-h) corresponds to an equivalent set of plots as in (a-d) just described but with the other eigenspace of L^2 with eigenvalue $\omega = 4$ a.u.The exact energy here is the valence band energy at K-point for WS_2 shown in Fig. 3.5 and is -0.0572 eV.

state happens to be the ground state in the valence band (VB) shown in Fig. 3.4(a). The corresponding energy errors shown in Fig. 3.7(g) are like in the previous case ($\omega = 0$) low for RBM-cl and RBM-qasm but in the range of $10^{-3} - 10^{-2}$ eV for RBM-IBMQ. Similar analysis as in the case of $\omega = 0$ a.u. can also be made for the constraint violation error in Fig. 3.7(f) and the state infidelity in Fig. 3.7(h). Both of these have low errors with the respective ranges as displayed.

Fig. 3.8 shows a similar plot for the other system studied WS_2. Just as before we display the results for $\omega = 0$ a.u. in Fig. 3.8(a-d) and for $\omega = 4$ a.u. in Fig. 3.8(e-h). Fig. 3.8(a) shows the energy match between the RBM implementations and the exact value for $\omega = 0$ a.u. and Fig. 3.8(e) shows the same for $\omega = 4$ a.u. The former is equal to the state in the conduction band at K-point (see Fig. 3.5(a)) and the latter is the corresponding state in the valence band (see Fig. 3.5(a)). We see good agreement for all RBM variants and the exact expected value. The corresponding energy errors are low (see Fig. 3.8(c) and Fig. 3.8(g)) with the range for IBMQ implementation being $10^{-3} - 10^{-2}$ eV and even lesser

for the pristine implementations. The respective constraint violation errors are displayed in Fig. 3.8(b) and Fig. 3.8(f) and are small too as seen from the scale. A similar statement can also be made for the state infidelity displayed in Fig. 3.8(d) and Fig. 3.8(h). We have seen that in both the systems MoS_2 and WS_2, the state infidelity and energy errors are higher in the $\omega = 4$ a.u. eigensector than in $\omega = 0$ a.u. eigensector in the IBMQ implementation especially. In fact, the relative energy errors for the said sector are close to 5% for RBM-IBMQ. However, the corresponding errors (both relative and absolute) are low for the noiseless implementation (RBM-cl and RBM-qasm) indicating that the higher % error is attributable to the imperfect implementation of gates in the Gibbs sampling circuit in an IBM-Q machine and hence can be mitigated with future quantum computing devices with better gate fidelities and error-correction schemes.

3.5 Conclusion

In this study, we have demonstrated an algorithm which can filter arbitrary energy eigenstates in 2D materials like TMDCs using a quantum circuit with quadratic resources. We provided an original proof of feasibility for our cost function employed for the constrained optimization. We also proved a generic lower bound for the successful sampling of our quantum circuit from which previously known bounds can be extracted. Our circuit trains a three-layered neural network that encodes the desired state using an RBM ansatz for the probability density. As an illustration, we were able to filter energy eigenstates in the conduction band of important TMDCs like MoS_2 and WS_2, and faithfully reproduce the band-gap. We were also able to filter arbitrary states based on a user-defined orbital angular momentum symmetry constraint. We trained the network on various flavors of computation using not only a classical computer, *qasm* backend quantum simulator in Qiskit but also a real IBMQ machine (IBM Sydney and IBM Toronto) with the objective to see the performance of the algorithm on actual NISQ devices. In all flavors of computation, our algorithm demonstrated very high accuracy when compared to the exact values obtained from direct diagonalization.

Venturing beyond the ground state to obtain arbitrary states based on user-defined restrictions is the first of its kind in all flavors of QML. Furthermore, the systems of our choice

happen to be TMDCs, an important class of 2D-periodic systems which have never been studied using any quantum algorithm. Periodic systems in general have received scanty attention as far as quantum algorithms are concerned. Only two reports exist[47], [55] both of which have simulated just the valence band in graphene and hexagonal Boron Nitride (h-BN).

It must also be emphasized that a host of classical algorithms have been developed in traditional quantum chemistry that are extremely accurate and polynomially efficient. Over the past few decades, Density functional theory (DFT) has emerged into a leading candidate for accurate computation of wide-variety of electronic structure problems in molecules and materials[104], [105]. Variants of it are being developed for cases wherein multi-reference correlation would be important too[106]. Reduced density-matrix based methods are also polynomially scaling [107] and have shown excellent accuracy in strongly correlated systems[108], [109]. Tensor-network based methods like Density-Matrix Renormalization Group (DMRG) [110]–[112] have been developed which even though capable of exploiting rank-sparsity in strongly correlated one-dimensional systems yet loses the polynomial advantage in multi-dimensions. Like our algorithm which attempts to construct the many-body state, a plethora of similar wave-function based *ab-initio* methods exist in traditional quantum chemistry too starting from the uncorrelated Hartree-Fock method to post-Hartree methods which can recover dynamic correlation like perturbative approaches (like MP2[113]–[115]), Truncated Configuration-Interaction or CI (like CISD) [114], [116], Couple-Cluster (CC) methods [117] (like CCSD, CCSD(T), CCSDT or EOM-CCSD for excited states), recently developed SHCI methods[118], [119] to ones which are good for capturing static-correlation like Multi-Configurational Self-Consistent Field (MCSCF[120]). A direct comparison of a quantum algorithm like ours with these classical algorithms can be attempted to be made in terms of accuracy and resource cost. In terms of resource requirements, the comparison is made difficult by the fact that certain parameters like circuit-width, circuit-depth, etc which affect the performance of quantum algorithms like ours, have no classical analogues. If we consider an N_{elec} electron system with $r = r_o + r_{uo}$ spin orbitals/fermionic modes such that $r_o = N_{elec}$ is the occupied orbital set in Hartree-Fock reference and r_{uo} are virtual orbitals excluded from Hartree-Fock reference, then under the assumption that

the orbital space rank loosely equates to qubits or number of visible neurons n (see section 3.3.1 and [69]), we have shown in section 3.3.3 that the spin-orbital cost of our algorithm would be $\approx O(r^2) = O(r_o^2 + r_{uo}^2 + 2r_{uo}r_o)$. The numerical parameter count of our algorithm is also quadratic i.e. $O(\alpha n^2) \approx O(r^2) = O(r_o^2 + r_{uo}^2 + 2r_{uo}r_o)$. This is unlike methods like CCSD (Coupled-Cluster Singles Doubles) which has a computational cost of $\approx O(r_o^2 r_{uo}^4)$ (for CCSDT it is $\approx O(r_o^3 r_{uo}^5)$ and for CCSD(T) it is $\approx O(r_o^3 r_{uo}^4)$) [117], [121]. Also CCSD evaluates $\approx O(r_o^2 r_{uo}^2)$ cluster amplitudes as parameters defining the excitations. For chemically important phenomenon like dissociation events which are no longer single-referenced are known to be difficult to treat with CCSD [122], even though pair cluster doubles can ameliorate the situation to some extent [123]. That being said, it must also be noted that traditional variant of CCSD unlike ours is non-variational. As far as accuracy is concerned, all results in this report are benchmarked against exponentially scaling exact diagonalization as that affords the exact value in a given basis. Not only the physics of Bloch states in the material TMDC but also a molecular example like LiH has been treated using our algorithm (see Appendix B). For both the ground and excited states of LiH we see good accuracy and improvement of error by enhancing the hidden node density which makes the ansatz more expressive. Studies on larger molecular systems for which the results of exact diagonalization may not be available may be undertaken in future. That will provide a platform for comparison in accuracy with a subset of the aforesaid classical algorithms. Desirable chemical features like size-consistency and size-extensivity may be probed too.

One must also note that several quantum algorithms already exist which aim at obtaining ground and excited states of fermionic systems[124]. Non-variational quantum algorithms like Quantum Phase estimation(QPE)[40], [125], [126] have exponential speed up[127] yet require high circuit depth and long coherent operations which are beyond the limits of near-term hardwares [23]. Hybrid Variational Quantum Algorithms (VQA) have also been developed which can ameliorate some of the above problems[46]. The most notable one in the list is Unitary Coupled-Cluster Variational Quantum Eigensolver (UCC-VQE) [43]. In its most traditional variant, the unitary ansatz which UCC-VQE uses for state preparation consists of single and double-excitations[128]–[130](hence often called the name Unitary Coupled Cluster Singles Doubles or UCCSD) from the reference state. However, the circuit

depth in preparing such an ansatz is still large and the circuit is parameterized by many variables which necessitates a high-dimensional classical optimization routine [131] to update the parameters. To be concrete, for r_o and r_{uo} with the same meaning as described in previous paragraph, the UCCSD-VQE uses [124] $O(r_o)$ qubits, $O(r_o^2 r_{uo}^2)$ cluster amplitudes as parameters and $O(f r_o^4)$ gates where $f = O(r_o)$ or $\mathrm{O(log(r_o))}$ depending on the qubit-mapping. Besides the UCC-VQE method can suffer from errors incurred due to operator ordering or Trotterization [132]. Also, the ansatz requires a high-degree of qubit connectivity for non-local operations which may not always be available in all hardware [133]. Hardware-efficient ansatz [44], [134] has been developed to help solve the above issues which use an alternating framework of single-qubit gates and fixed entangling operations which can be chosen with the specific capabilities of the device in hand. However, unlike UCC-VQE, such an ansatz is not physically inspired and often suffers from trainability issues during parameter optimization [46], [135]. Besides the number of parameters grow as a function of entangling blocks and can even surpass the size of the Hilbert space [124], [134]. A third variant that has low circuit dept h and parameter cost is the ADAPT-VQE approach [136]. Unlike in the previous two cases, this variant constructs the circuit from a pre-selected pool of operators and changes the circuit architecture adaptively by adding operators from the pool which affects the energy gradient the most. The chosen pool decides the parameter count and gate-counts in the circuit. In this method, the number of measurement shots can be high for computing the gradients [131] and also it is generally not clear how to pre-select the operator pool and what guarantees that the pool is complete i.e. the ansatz it produces is expressive enough. Many different variants for each method have been constructed for which the reader is referred to many excellent reviews like [46], [124], [133]. For excited states [46], deflation assisted VQE as described before [58] exist but for its implementation used the UCCSD ansatz which inherits some of the above problems of high parameter count and gates. A recent promising method known as Weighted Subspace- Search VQE uses an input array of several orthogonal states to construct a weighted Lagrangian as the cost function [137]. In this case, the input states are mapped to the excited states of the system using a parameterized ansatz circuit. Depending on the nature of the ansatz circuit, the algorithm can have different gate-count or parameter count and hence it is hard to mention a general

estimate.

Our algorithm is also a hybrid variational algorithm like the ones in the aforesaid list but requiring quadratic resources always (see Section 3.3.3). However there are some key differences as well which need to be acknowledged. Unlike the above list of algorithms which prepare a unitary ansatz on a quantum computer to mimic the state, our algorithm proposes to construct a probability distribution that mimics the amplitude field of the target state on a quantum computer. As a result our algorithm is a distribution sampling protocol on a quantum computer using a non-unitary ansatz (RBM) which is manifested in the usage of ancilla and its subsequent measurement collapse. The measurement statistics of such a collapse are discussed in Appendix B. Since the distribution encoding the amplitude field is based on RBM, unlike the ADAPT-VQE method, our protocol is largely *problem agnostic*. This is due to the fact that RBM can act as a universal approximant to any probability density [19] and hence can be used for a variety of problems provided it is made sufficiently expressive with an adequate hidden node density. Also unlike other algorithms wherein the nature of the excitations or operator pool used decides the cost-function gradient, in our case the distribution function being RBM always permits training the network with analytical gradients. Besides we have already demonstrated in section 3.3.3 using Ref [76] that an analogous classical construction of RBM distribution has an exponential overhead whereas using a quantum algorithm like ours one can construct it using quadratic resources thereby illustrating the distinct quantum-classical advantage in our algorithm directly.

Further extension of this algorithm can be made to compute operators using Hellmann-Feynmann method [138], to characterize the influence of noise on the algorithm and to see it being extended to study other interesting phenomena on 2D materials like Rashba splitting in polar TMDCs[139] or even effect of strain[140]. One must also note that in this work we construct the full $d = 2^n$-dimensional eigenstate from the amplitude encoding using the RBM ansatz (Eq. 3.4) and the phase encoding using the Eq. 3.5. This is because the primary quantum advantage of our algorithm lies in the fact we use quadratic resources to learn the full RBM distribution which classically would require exponential resources as necessitated in [76]. Besides access to the full state allows us to compute matrix elements of arbitrary operator between eigenstates important for spectral information i.e. learning in excitonic

features[101] or thermal and electronic conductivity [103] which as said before are important future extensions of this work. Also once trained for a given system, the neural-network in our algorithm can be used to learn eigenstates of a closely related system accurately with faster convergence and lesser number of iterations indicating partial transferability of these models (see Appendix B). Benefits and scope of such 'transferable training' for other chemically motivated systems will be investigated in future. It must be noted that the symmetry partitioning of the metal orbitals in TMDCs guaranteed in [90] have reduced the effective size of the orbital space and qubit requirements in this study. However understanding spectral information in excitonic physics would require more involved models with a larger orbital space. A way forward may be focusing on low-energy excitons with a certain symmetry (like overall spin-angular momentum) characteristics only. For molecular systems such symmetry inspired cost-reductions are already beginning to be noticed [70], [72] as discussed earlier. However such an initiative for materials is largely an uncharted territory. Further reduction in qubit resource requirements of our algorithm may also help, even though the non-unitary nature of the ansatz as discussed before makes it harder. From the hardware point of view, robust large scale error mitigation strategies are beginning to be made available now [141], [142] and devices with over 1000 qubits with low qubit decoherence errors and gate infidelities are also being promised in recent future [143]. Such resources would will certainly be beneficial to extensions of studies like these.

From the algorithmic point of view, besides being quadratic scaling in qubit and gate requirements and parameter count, our algorithm does not have any dependence on oracular objects like qRAM [35] which is responsible for creating a superposition of all possible basis states and is known to commonly sought in most quantum machine learning modules. As futuristic quantum devices are being developed with proper error mitigation schemes, we expect to have more such cross-pollination between machine learning algorithms and quantum computing with the promise to study electronic structure and dynamics in new complex materials.

References

[1] S. Cohen, "Chapter 1 - The evolution of machine learning: past, present, and future," in *Artificial Intelligence and Deep Learning in Pathology*, S. Cohen, Ed., Elsevier, 2021, pp. 1–12, ISBN: 978-0-323-67538-3. DOI: https://doi.org/10.1016/B978-0-323-67538-3.00001-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323675383000014.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2722–2730. DOI: 10.1109/ICCV.2015.312.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] H. Sak, A. Senior, K. Rao, *et al.*, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, pp. 4280–4284.

[5] M. B. Hoy, "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants.," eng, *Medical reference services quarterly*, vol. 37, no. 1, pp. 81–88, 2018, ISSN: 1540-9597 (Electronic). DOI: 10.1080/02763869.2018.1404391.

[6] D. Silver, A. Huang, C. J. Maddison, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016, ISSN: 1476-4687. DOI: 10.1038/nature16961. [Online]. Available: https://doi.org/10.1038/nature16961.

[7] K. Bansak, J. Ferwerda, J. Hainmueller, *et al.*, "Improving refugee integration through data-driven algorithmic assignment," *Science*, vol. 359, no. January, pp. 325–329, 2018.

[8] G. Carleo, I. Cirac, K. Cranmer, *et al.*, "Machine learning and the physical sciences," *Rev. Mod. Phys.*, vol. 91, no. 4, p. 45 002, Dec. 2019. DOI: 10.1103/RevModPhys.91.045002. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.91.045002.

[9] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding Density Functionals with Machine Learning," *Phys. Rev. Lett.*, vol. 108, no. 25, p. 253 002, Jun. 2012. DOI: 10.1103/PhysRevLett.108.253002. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.108.253002.

[10] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the Kohn-Sham equations with machine learning," *Nature Communications*, vol. 8, no. 1, p. 872, 2017, ISSN: 2041-1723. DOI: 10.1038/s41467-017-00839-3. [Online]. Available: https://doi.org/10.1038/s41467-017-00839-3.

[11] L. F. Arsenault, A. Lopez-Bezanilla, O. A. Von Lilienfeld, and A. J. Millis, "Machine learning for many-body physics: The case of the Anderson impurity model," *Physical Review B - Condensed Matter and Materials Physics*, vol. 90, no. 15, pp. 1–16, 2014, ISSN: 1550235X. DOI: 10.1103/PhysRevB.90.155136. arXiv: 1408.1143.

[12] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons," *Phys. Rev. Lett.*, vol. 104, no. 13, p. 136 403, Apr. 2010. DOI: 10.1103/PhysRevLett.104.136403. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.104.136403.

[13] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," *Physical Review Letters*, vol. 114, no. 9, pp. 1–5, 2015, ISSN: 10797114. DOI: 10.1103/PhysRevLett.114.096405.

[14] L. Wang, "Discovering phase transitions with unsupervised learning," *Physical Review B*, vol. 94, no. 19, pp. 2–6, 2016, ISSN: 24699969. DOI: 10.1103/PhysRevB.94.195105. arXiv: 1606.00318.

[15] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, ISSN: 14764687. DOI: 10.1038/nature14539.

[16] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, pp. 431–434, Feb. 2017, ISSN: 1745-2481. DOI: 10.1038/nphys4035. [Online]. Available: http://dx.doi.org/10.1038/nphys4035.

[17] K. Ch'Ng, J. Carrasquilla, R. G. Melko, and E. Khatami, "Machine learning phases of strongly correlated fermions," *Physical Review X*, vol. 7, no. 3, pp. 1–9, 2017, ISSN: 21603308. DOI: 10.1103/PhysRevX.7.031038. arXiv: 1609.02552.

[18] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, Aug. 2019, ISSN: 1745-2481. DOI: 10.1038/s41567-019-0648-8. [Online]. Available: http://dx.doi.org/10.1038/s41567-019-0648-8.

[19] N. Le Roux and Y. Bengio, "Representational Power of Restricted Boltzmann Machines and Deep Belief Networks," *Neural Computation*, vol. 20, no. 6, pp. 1631–1649, 2008. DOI: 10.1162/neco.2008.04-07-510.

[20] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, "Restricted Boltzmann machines in quantum physics," *Nature Physics*, vol. 15, no. 9, pp. 887–892, 2019, ISSN: 17452481. DOI: 10.1038/s41567-019-0545-1. [Online]. Available: http://dx.doi.org/10.1038/s41567-019-0545-1.

[21] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography," *Nature Physics*, vol. 14, no. 5, pp. 447–450, 2018.

[22] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[23] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[24] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017, ISSN: 1476-4687. DOI: 10.1038/nature23474. [Online]. Available: https://doi.org/10.1038/nature23474.

[25] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014, ISSN: 17452481. DOI: 10.1038/NPHYS3029. arXiv: 1307.0401.

[26] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130 503, 2014.

[27] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-Enhanced Machine Learning," *Physical Review Letters*, vol. 117, no. 13, pp. 1–6, 2016, ISSN: 10797114. DOI: 10.1103/PhysRevLett.117.130501. arXiv: 1610.08251.

[28] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv preprint arXiv:1307.0411*, 2013.

[29] M. Otten, I. R. Goumiri, B. W. Priest, G. F. Chapline, and M. D. Schneider, "Quantum machine learning using gaussian processes with performant quantum kernels," *arXiv preprint arXiv:2004.11280*, 2020.

[30] Z. Zhao, J. K. Fitzsimons, and J. F. Fitzsimons, "Quantum-assisted gaussian process regression," *Physical Review A*, vol. 99, no. 5, p. 052 331, 2019.

[31] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a large scale classifier with the quantum adiabatic algorithm," *arXiv preprint arXiv:0912.0779*, 2009.

[32] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum Algorithm for Linear Systems of Equations," *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150 502, Oct. 2009. DOI: 10.1103/PhysRevLett.103.150502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[33] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, "Quantum singular-value decomposition of nonsparse low-rank matrices," *Phys. Rev. A*, vol. 97, no. 1, p. 12 327, Jan. 2018. DOI: 10.1103/PhysRevA.97.012327. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.97.012327.

[34] X. Cui and Y. Shi, *QBLAS : A Quantum Basic Linear Algebra and Simulation Library*.

[35] C. Ciliberto, M. Herbster, A. D. Ialongo, *et al.*, "Quantum machine learning: a classical perspective," *arXiv*, 2017, ISSN: 23318422.

[36] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum Boltzmann Machine," *Physical Review X*, vol. 8, no. 2, 2018, ISSN: 21603308. DOI: 10.1103/PhysRevX.8.021050. arXiv: 1601.02036.

[37] M. Kieferová and N. Wiebe, "Tomography and generative training with quantum Boltzmann machines," *Phys. Rev. A*, vol. 96, no. 6, p. 62 327, Dec. 2017. DOI: 10.1103/PhysRevA.96.062327. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.96.062327.

[38] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum deep learning," *arXiv preprint arXiv:1412.3489*, 2014.

[39] R. Xia and S. Kais, "Quantum machine learning for electronic structure calculations," *Nature Communications*, vol. 9, no. 1, p. 4195, 2018, ISSN: 2041-1723. DOI: 10.1038/s41467-018-06598-z. [Online]. Available: https://doi.org/10.1038/s41467-018-06598-z.

[40] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated Quantum Computation of Molecular Energies," *Science*, vol. 309, no. 5741, pp. 1704–1707, 2005, ISSN: 0036-8075. DOI: 10.1126/science.1113479. [Online]. Available: https://science.sciencemag.org/content/309/5741/1704.

[41] S. E. Smart and D. A. Mazziotti, "Quantum solver of contracted eigenvalue equations for scalable molecular simulations on quantum computing devices," *Physical Review Letters*, vol. 126, no. 7, p. 070 504, 2021.

[42] S. Kais, "Introduction to quantum information and computation for chemistry," in *Quantum Information and Computation for Chemistry*. John Wiley & Sons, Ltd, 2014, ch. 1, pp. 1–38, ISBN: 9781118742631. DOI: 10.1002/9781118742631.ch01. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118742631.ch01. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118742631.ch01.

[43] A. Peruzzo, J. McClean, P. Shadbolt, *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, p. 4213, Jul. 2014. [Online]. Available: https://doi.org/10.1038/ncomms5213%20http://10.0.4.14/ncomms5213%20https://www.nature.com/articles/ncomms5213%7B%5C#%7Dsupplementary-information.

[44] A. Kandala, A. Mezzacapo, K. Temme, *et al.*, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, p. 242, Sep. 2017. [Online]. Available: https://doi.org/10.1038/nature23879%20http://10.0.4.14/nature23879%20https://www.nature.com/articles/nature23879%7B%5C#%7Dsupplementary-information.

[45] A. Daskin and S. Kais, "Direct application of the phase estimation algorithm to find the eigenvalues of the hamiltonians," *Chemical Physics*, vol. 514, pp. 87–94, 2018.

[46] M. Cerezo, A. Arrasmith, R. Babbush, *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[47] S. Kanno and T. Tada, "Many-body calculations for periodic materials via restricted boltzmann machine-based vqe," *Quantum Science and Technology*, vol. 6, no. 2, p. 025 015, 2021.

[48] E. M. Stoudenmire and S. R. White, "Studying two-dimensional systems with the density matrix renormalization group," *Annu. Rev. Condens. Matter Phys.*, vol. 3, no. 1, pp. 111–128, 2012.

[49] C. Umrigar, K. Wilson, and J. Wilkins, "Optimized trial wave functions for quantum monte carlo calculations," *Physical Review Letters*, vol. 60, no. 17, p. 1719, 1988.

[50] S. Pathak, B. Busemeyer, J. N. Rodrigues, and L. K. Wagner, "Excited states in variational monte carlo using a penalty method," *The Journal of Chemical Physics*, vol. 154, no. 3, p. 034 101, 2021.

[51] K. Kuroiwa and Y. O. Nakagawa, "Penalty methods for a variational quantum eigensolver," *Physical Review Research*, vol. 3, no. 1, pp. 21–24, 2021. DOI: 10.1103/physrevresearch.3.013197. arXiv: 2010.13951.

[52] I. G. Ryabinkin, S. N. Genin, and A. F. Izmaylov, "Constrained Variational Quantum Eigensolver: Quantum Computer Search Engine in the Fock Space," *Journal of Chemical Theory and Computation*, vol. 15, no. 1, pp. 249–255, Jan. 2019, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.8b00943. [Online]. Available: https://doi.org/10.1021/acs.jctc.8b00943.

[53] G. Greene-Diniz and D. Muñoz Ramo, "Generalized unitary coupled cluster excitations for multireference molecular states optimized by the variational quantum eigensolver," *International Journal of Quantum Chemistry*, vol. 121, no. 4, pp. 1–20, 2021, ISSN: 1097461X. DOI: 10.1002/qua.26352. arXiv: 1910.05168.

[54] Y. Suzuki, Y. Kawase, Y. Masumura, *et al.*, "Qulacs: A fast and versatile quantum circuit simulator for research purpose," 2020. arXiv: 2011.13524 [quant-ph].

[55] S. H. Sureshbabu, M. Sajjan, S. Oh, and S. Kais, "Implementation of Quantum Machine Learning for Electronic Structure Calculations of Periodic Systems on Quantum Computing Devices," *arXiv*, pp. 1–31, 2021. arXiv: 2103.02037. [Online]. Available: http://arxiv.org/abs/2103.02037.

[56] R. A. Miranda-Quintana and M. M. González, "Deflation techniques in quantum chemistry: Excited states from ground states," *International Journal of Quantum Chemistry*, vol. 113, no. 22, pp. 2478–2488, 2013, ISSN: 00207608. DOI: 10.1002/qua.24486.

[57] S. Hemmatiyan, M. Sajjan, A. W. Schlimgen, and D. A. Mazziotti, "Excited-state spectra of strongly correlated molecules from a reduced-density-matrix approach," *The Journal of Physical Chemistry Letters*, vol. 9, no. 18, pp. 5373–5378, 2018, PMID: 30183311. DOI: 10.1021/acs.jpclett.8b02455.

[58] O. Higgott, D. Wang, and S. Brierley, "Variational Quantum Computation of Excited States," *Quantum*, vol. 3, p. 156, 2019, ISSN: 2521-327X. DOI: 10.22331/q-2019-07-01-156. arXiv: 1805.08138.

[59] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[60] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[61] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[62] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recognition*, vol. 47, no. 1, pp. 25–39, 2014, ISSN: 00313203. DOI: 10.1016/j.patcog.2013.05.025.

[63] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[64] P. Mehta, M. Bukov, C. H. Wang, *et al.*, "A high-bias, low-variance introduction to Machine Learning for physicists," *Physics Reports*, vol. 810, pp. 1–124, 2019, ISSN: 03701573. DOI: 10.1016/j.physrep.2019.03.001. arXiv: 1803.08823.

[65] H. C. Nguyen, R. Zecchina, and J. Berg, "Inverse statistical problems: From the inverse ising problem to data science," *Advances in Physics*, vol. 66, no. 3, pp. 197–261, 2017.

[66] N. Zhang and S. Sun, "Multiview graph restricted boltzmann machines," *IEEE Transactions on Cybernetics*, 2021.

[67] D. Chen, J. Lv, and Z. Yi, "Graph regularized restricted boltzmann machine," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2651–2659, 2017.

[68] Z. Cai and J. Liu, "Approximating quantum many-body wave functions using artificial neural networks," *Physical Review B*, vol. 97, no. 3, p. 035 116, 2018.

[69] A. Tranter, P. J. Love, F. Mintert, and P. V. Coveney, "A comparison of the bravyi–kitaev and jordan–wigner transformations for the quantum simulation of quantum chemistry," *Journal of chemical theory and computation*, vol. 14, no. 11, pp. 5617–5630, 2018.

[70] K. Setia, R. Chen, J. E. Rice, A. Mezzacapo, M. Pistoia, and J. D. Whitfield, "Reducing qubit requirements for quantum simulations using molecular point group symmetries," *Journal of Chemical Theory and Computation*, vol. 16, no. 10, pp. 6091–6097, 2020.

[71] T. Takeshita, N. C. Rubin, Z. Jiang, E. Lee, R. Babbush, and J. R. McClean, "Increasing the representation accuracy of quantum simulations of chemistry without extra quantum resources," *Physical Review X*, vol. 10, no. 1, p. 011 004, 2020.

[72] F. Zhang, N. Gomes, N. F. Berthusen, *et al.*, "Shallow-circuit variational quantum eigensolver based on symmetry-inspired hilbert space partitioning for quantum chemical calculations," *Physical Review Research*, vol. 3, no. 1, p. 013 039, 2021.

[73] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015. arXiv: 1412.6980.

[74] P. Nekrasov, J. Freeze, and V. Batista, "Using restricted boltzmann machines to model molecular geometries," *arXiv preprint arXiv:2012.06984*, 2020.

[75] A. Borin and D. A. Abanin, "Approximating power of machine-learning ansatz for quantum many-body states," *Phys. Rev. B*, vol. 101, no. 19, p. 195 141, May 2020. DOI: 10.1103/PhysRevB.101.195141. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.101.195141.

[76] P. M. Long and R. A. Servedio, "Restricted Boltzmann Machines are hard to approximately evaluate or simulate," *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pp. 703–710, 2010.

[77] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[78] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, *et al.*, "Qiskit: An open-source framework for quantum computing," *Accessed on: Mar*, vol. 16, 2019.

[79] *27-qubit backend: "ibmq team, ibm q 27 sydney backend specification v1.0.46,"*, 2021. [Online]. Available: https://quantum-computing.ibm.com.

[80] *27-qubit backend: "ibmq team, ibm q 27 toronto backend specification v1.4.25,"*, 2021. [Online]. Available: https://quantum-computing.ibm.com.

[81] G. S. Barron and C. J. Wood, "Measurement error mitigation for variational quantum algorithms," *arXiv preprint arXiv:2010.08520*, 2020.

[82] L. Yang, C. Xie, J. Jin, *et al.*, "Properties, Preparation and Applications of Low Dimensional Transition Metal Dichalcogenides," *Nanomaterials*, vol. 8, no. 7, 2018, ISSN: 2079-4991. DOI: 10.3390/nano8070463. [Online]. Available: https://www.mdpi.com/2079-4991/8/7/463.

[83] "2D transition metal dichalcogenides," *Nature Reviews Materials*, vol. 2, 2017, ISSN: 20588437. DOI: 10.1038/natrevmats.2017.33.

[84] W. Bao, X. Cai, D. Kim, K. Sridhara, and M. S. Fuhrer, "High mobility ambipolar MoS2 field-effect transistors: Substrate and dielectric effects," *Applied Physics Letters*, vol. 102, no. 4, p. 42 104, 2013. DOI: 10.1063/1.4789365. [Online]. Available: https://doi.org/10.1063/1.4789365.

[85] N. Choudhary, M. D. Patel, J. Park, B. Sirota, and W. Choi, "Synthesis of large scale MoS2 for electronics and energy applications," *Journal of Materials Research*, vol. 31, no. 7, pp. 824–831, 2016. DOI: 10.1557/jmr.2016.100.

[86] E. C. Ahn, "2D materials for spintronic devices," *npj 2D Materials and Applications*, vol. 4, no. 1, p. 17, 2020, ISSN: 2397-7132. DOI: 10.1038/s41699-020-0152-0. [Online]. Available: https://doi.org/10.1038/s41699-020-0152-0.

[87] K. F. Mak, C. Lee, J. Hone, J. Shan, and T. F. Heinz, "Atomically Thin MoS_2: A New Direct-Gap Semiconductor," *Phys. Rev. Lett.*, vol. 105, no. 13, p. 136 805, Sep. 2010. DOI: 10.1103/PhysRevLett.105.136805. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.105.136805.

[88] C. Gong, Y. Zhang, W. Chen, *et al.*, "Electronic and Optoelectronic Applications Based on 2D Novel Anisotropic Transition Metal Dichalcogenides," *Advanced Science*, vol. 4, no. 12, p. 1 700 231, 2017. DOI: https://doi.org/10.1002/advs.201700231. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201700231.

[89] K. F. Mak, D. Xiao, and J. Shan, "Light valley interactions in 2D semiconductors," *Nature Photonics*, vol. 12, no. 8, pp. 451–460, 2018, ISSN: 1749-4893. DOI: 10.1038/s41566-018-0204-6. [Online]. Available: https://doi.org/10.1038/s41566-018-0204-6.

[90] G. B. Liu, W. Y. Shan, Y. Yao, W. Yao, and D. Xiao, "Three-band tight-binding model for monolayers of group-VIB transition metal dichalcogenides," *Physical Review B - Condensed Matter and Materials Physics*, vol. 88, no. 8, pp. 1–11, 2013, ISSN: 10980121. DOI: 10.1103/PhysRevB.88.085433. arXiv: 1305.6089.

[91] E. Ridolfi, D. Le, T. S. Rahman, E. R. Mucciolo, and C. H. Lewenkopf, "A tight-binding model for MoS_2 monolayers," *Journal of Physics: Condensed Matter*, vol. 27, no. 36, p. 365 501, Aug. 2015. DOI: 10.1088/0953-8984/27/36/365501. [Online]. Available: https://doi.org/10.1088/0953-8984/27/36/365501.

[92] F. Zahid, L. Liu, Y. Zhu, J. Wang, and H. Guo, "A generic tight-binding model for monolayer, bilayer and bulk MoS_2," *AIP Advances*, vol. 3, no. 5, p. 52 111, 2013. DOI: 10.1063/1.4804936. [Online]. Available: https://doi.org/10.1063/1.4804936.

[93] M. Shahriari, A. Ghalambor Dezfuli, and M. Sabaeian, "Band structure and orbital character of monolayer MoS_2 with eleven-band tight-binding model," *Superlattices and Microstructures*, vol. 114, pp. 169–182, 2018, ISSN: 0749-6036. DOI: https://doi.org/10.1016/j.spmi.2017.12.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0749603617329154.

[94] S. Fang, R. Kuate Defo, S. N. Shirodkar, S. Lieu, G. A. Tritsaris, and E. Kaxiras, "Ab initio tight-binding Hamiltonian for transition metal dichalcogenides," *Phys. Rev. B*, vol. 92, no. 20, p. 205 108, Nov. 2015. DOI: 10.1103/PhysRevB.92.205108. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.92.205108.

[95] D. Xiao, G.-B. Liu, W. Feng, X. Xu, and W. Yao, "Coupled spin and valley physics in monolayers of mos 2 and other group-vi dichalcogenides," *Physical review letters*, vol. 108, no. 19, p. 196 802, 2012.

[96] A. Kormányos, V. Zólyomi, N. D. Drummond, P. Rakyta, G. Burkard, and V. I. Fal'Ko, "Monolayer MoS2: Trigonal warping, the Γ valley, and spin-orbit coupling effects," *Physical Review B - Condensed Matter and Materials Physics*, vol. 88, no. 4, pp. 1–8, 2013, ISSN: 10980121. DOI: 10.1103/PhysRevB.88.045416. arXiv: 1304.4084.

[97] O. L. Berman and R. Y. Kezerashvili, "High-temperature superfluidity of the two-component bose gas in a transition metal dichalcogenide bilayer," *Physical Review B*, vol. 93, no. 24, p. 245 410, 2016.

[98] Y. Ominato, J. Fujimoto, and M. Matsuo, "Valley-Dependent Spin Transport in Monolayer Transition-Metal Dichalcogenides," *Physical Review Letters*, vol. 124, no. 16, p. 166 803, 2020, ISSN: 10797114. DOI: 10.1103/PhysRevLett.124.166803. arXiv: 2001.08670. [Online]. Available: https://doi.org/10.1103/PhysRevLett.124.166803.

[99] M. Van Der Donck, M. Zarenia, and F. M. Peeters, "Excitons and trions in monolayer transition metal dichalcogenides: A comparative study between the multiband model and the quadratic single-band model," *Physical Review B*, vol. 96, no. 3, 2017, ISSN: 24699969. DOI: 10.1103/PhysRevB.96.035131. arXiv: 1707.07509.

[100] M. Van der Donck, "Excitonic complexes in transition metal dichalcogenides and related materials," Ph.D. dissertation, University of Antwerp, 2019.

[101] H. Fang, H. A. Bechtel, E. Plis, *et al.*, "Quantum of optical absorption in two-dimensional semiconductors," *Proceedings of the National Academy of Sciences*, vol. 110, no. 29, pp. 11 688–11 691, 2013.

[102] M. L. Trolle, G. Seifert, and T. G. Pedersen, "Theory of excitonic second-harmonic generation in monolayer mos 2," *Physical Review B*, vol. 89, no. 23, p. 235 410, 2014.

[103] P. Sengupta, Y. Tan, G. Klimeck, and J. Shi, "Low-temperature thermal transport and thermopower of monolayer transition metal dichalcogenide semiconductors," *Journal of Physics: Condensed Matter*, vol. 29, no. 40, p. 405 701, 2017.

[104] K. Burke, "Perspective on density functional theory," *The Journal of chemical physics*, vol. 136, no. 15, p. 150 901, 2012.

[105] R. J. Maurer, C. Freysoldt, A. M. Reilly, *et al.*, "Advances in density-functional calculations for materials modeling," *Annual Review of Materials Research*, vol. 49, pp. 1–30, 2019.

[106] L. Gagliardi, D. G. Truhlar, G. Li Manni, R. K. Carlson, C. E. Hoyer, and J. L. Bao, "Multiconfiguration pair-density functional theory: A new way to treat strongly correlated systems," *Accounts of chemical research*, vol. 50, no. 1, pp. 66–73, 2017.

[107] D. A. Mazziotti, "Large-scale semidefinite programming for many-electron quantum mechanics," *Physical review letters*, vol. 106, no. 8, p. 083 001, 2011.

[108] D. A. Mazziotti, "Two-electron reduced density matrix as the basic variable in many-electron quantum chemistry and physics," *Chemical reviews*, vol. 112, no. 1, pp. 244–262, 2012.

[109] J. M. Montgomery and D. A. Mazziotti, "Strong electron correlation in nitrogenase cofactor, femoco," *The Journal of Physical Chemistry A*, vol. 122, no. 22, pp. 4988–4996, 2018.

[110] S. R. White, "Density matrix formulation for quantum renormalization groups," *Physical review letters*, vol. 69, no. 19, p. 2863, 1992.

[111] G. K.-L. Chan and S. Sharma, "The density matrix renormalization group in quantum chemistry," *Annual review of physical chemistry*, vol. 62, no. 1, pp. 465–481, 2011.

[112] U. Schollwöck, "The density-matrix renormalization group," *Reviews of modern physics*, vol. 77, no. 1, p. 259, 2005.

[113] M. Head-Gordon, J. A. Pople, and M. J. Frisch, "Mp2 energy evaluation by direct methods," *Chemical physics letters*, vol. 153, no. 6, pp. 503–506, 1988.

[114] K. Raghavachari and J. B. Anderson, "Electron correlation effects in molecules," *The Journal of Physical Chemistry*, vol. 100, no. 31, pp. 12 960–12 973, 1996.

[115] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular electronic-structure theory*. John Wiley & Sons, 2014, ch. 14, pp. 724–816.

[116] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular electronic-structure theory*. John Wiley & Sons, 2014, ch. 11, pp. 523–597.

[117] R. J. Bartlett and M. Musiał, "Coupled-cluster theory in quantum chemistry," *Reviews of Modern Physics*, vol. 79, no. 1, p. 291, 2007.

[118] J. Li, M. Otten, A. A. Holmes, S. Sharma, and C. J. Umrigar, "Fast semistochastic heat-bath configuration interaction," *The Journal of chemical physics*, vol. 149, no. 21, p. 214 110, 2018.

[119] A. A. Holmes, C. Umrigar, and S. Sharma, "Excited states using semistochastic heat-bath configuration interaction," *The Journal of Chemical Physics*, vol. 147, no. 16, p. 164 111, 2017.

[120] P. G. Szalay, T. Muller, G. Gidofalvi, H. Lischka, and R. Shepard, "Multiconfiguration self-consistent field and multireference configuration interaction methods and applications," *Chemical reviews*, vol. 112, no. 1, pp. 108–181, 2012.

[121] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular electronic-structure theory*. John Wiley & Sons, 2014, ch. 13, pp. 648–723.

[122] P.-D. Fan and P. Piecuch, "The usefulness of exponential wave function expansions employing one-and two-body cluster operators in electronic structure theory: The extended and generalized coupled-cluster methods," *Advances in Quantum Chemistry*, vol. 51, pp. 1–57, 2006.

[123] I. W. Bulik, T. M. Henderson, and G. E. Scuseria, "Can single-reference coupled cluster theory describe static correlation?" *Journal of chemical theory and computation*, vol. 11, no. 7, pp. 3171–3179, 2015.

[124] Y. Cao, J. Romero, J. P. Olson, *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[125] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, *et al.*, "Towards quantum chemistry on a quantum computer," *Nature Chemistry*, vol. 2, p. 106, Jan. 2010. [Online]. Available: https://doi.org/10.1038/nchem.483%20http://10.0.4.14/nchem.483%20https://www.nature.com/articles/nchem.483%7B%5C#%7Dsupplementary-information.

[126] L. Veis, J. Višňák, H. Nishizawa, H. Nakai, and J. Pittner, "Quantum chemistry beyond born–oppenheimer approximation on a quantum computer: A simulated phase estimation study," *International Journal of Quantum Chemistry*, vol. 116, no. 18, pp. 1328–1336, 2016.

[127] D. S. Abrams and S. Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," *Physical Review Letters*, vol. 83, no. 24, p. 5162, 1999.

[128] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023 023, 2016.

[129] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," *Quantum Science and Technology*, vol. 4, no. 1, p. 014 008, 2018.

[130] M.-H. Yung, J. Casanova, A. Mezzacapo, *et al.*, "From transistor to trapped-ion computers for quantum chemistry," *Scientific reports*, vol. 4, no. 1, pp. 1–7, 2014.

[131] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, "Vqe method: A short survey and recent developments," *Materials Theory*, vol. 6, no. 1, pp. 1–21, 2022.

[132] H. R. Grimsley, D. Claudino, S. E. Economou, E. Barnes, and N. J. Mayhall, "Is the trotterized uccsd ansatz chemically well-defined?" *Journal of chemical theory and computation*, vol. 16, no. 1, pp. 1–6, 2019.

[133] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, "Quantum algorithms for quantum chemistry and quantum materials science," *Chemical Reviews*, vol. 120, no. 22, pp. 12 685–12 717, 2020.

[134] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, *et al.*, "Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions," *Physical Review A*, vol. 98, no. 2, p. 022 322, 2018.

[135] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[136] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature communications*, vol. 10, no. 1, pp. 1–9, 2019.

[137] K. M. Nakanishi, K. Mitarai, and K. Fujii, "Subspace-search variational quantum eigensolver for excited states," *Physical Review Research*, vol. 1, no. 3, p. 033 062, 2019.

[138] S. Oh, "Quantum computational method of finding the ground-state energy and expectation values," *Phys. Rev. A*, vol. 77, p. 012 326, 1 Jan. 2008. DOI: 10.1103/PhysRevA.77.012326. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.77.012326.

[139] Q.-F. Yao, J. Cai, W.-Y. Tong, *et al.*, "Manipulation of the large Rashba spin splitting in polar two-dimensional transition-metal dichalcogenides," *Phys. Rev. B*, vol. 95, no. 16, p. 165 401, Apr. 2017. DOI: 10.1103/PhysRevB.95.165401. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.95.165401.

[140] Z. Peng, X. Chen, Y. Fan, D. J. Srolovitz, and D. Lei, "Strain engineering of 2D semiconductors and graphene: from strain fields to band-structure tuning and photonic applications," *Light: Science and Applications*, vol. 9, no. 1, 2020, ISSN: 20477538. DOI: 10.1038/s41377-020-00421-5. [Online]. Available: http://dx.doi.org/10.1038/s41377-020-00421-5.

[141] Z. Chen, K. J. Satzinger, J. Atalaya, *et al.*, "Exponential suppression of bit or phase errors with cyclic error correction," *Nature*, vol. 595, no. 7867, pp. 383–387, 2021.

[142] M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen, "General error mitigation for quantum circuits," *Quantum Information Processing*, vol. 19, no. 11, pp. 1–17, 2020.

[143] J. Gambetta, "Ibm's roadmap for scaling quantum technology," 2020.

4. FINITE-SIZE SCALING ON A DIGITAL QUANTUM SIMULATOR

The contents of this chapter are adapted from the article 'Khalid, B., Sureshbabu, S. H., Banerjee, A., & Kais, S. (2022). Finite-size scaling on a digital quantum simulator using quantum restricted Boltzmann machine. Frontiers in Physics, 464'.

4.1 Introduction

A phase transition occurs whenever the thermodynamic state variables of a system become non-analytic e.g. as a liquid changes into a gas, the density of the system changes discontinuously. If the phase transition occurs at a finite temperature $T \neq 0$, the transition is called a classical phase transition (CPT) as it is dominated by thermal fluctuations. On the other hand, if the transition occurs by tuning some parameter in the system's Hamiltonian as $T \to 0$, it is called a quantum phase transition (QPT) since it is dominated by quantum fluctuations. A CPT appears only when the system is infinite i.e. in the thermodynamic limit[1]. On the other hand, a QPT doesn't necessarily require the thermodynamic limit. Recently there has been a lot of interest in QPTs occurring in finite size quantum optical systems[2]–[7].

Quantum Rabi Model (QRM) describes the interaction of a two-level system with a bosonic field mode (see Eq. (4.1) for the Hamiltonian.) This model has gained a lot of significance in the study of ultrastrong light-matter coupling regimes e.g. in circuit QED where the so-called counterrotating terms can not be ignored[8]. Quantum Rabi Model has been shown to exhibit a QPT[2]. Namely, when the energy separation of the two levels in the system Ω becomes infinitely large compared to the frequency of the bosonic mode ω_0, the ground state of the Hamiltonian undergoes a phase transition from a normal phase to a superradiant phase as the light-matter coupling exceeds the critical value. Moreover, the ground state of the Jaynes-Cummings model (JCM) which can be obtained from the QRM by performing the rotating-wave approximation has also been shown to exhibit the normal-superradiant phase transition[3]. Later on, a more general anisotropic QRM in which the rotating and counter-rotating terms can have different coupling strengths was also considered[4]. The

QRM and JCM are limiting cases of this model. It was shown that the ground state for this more general case also undergoes the normal-superradiant phase transition. The phase transition in QRM has also been demonstrated experimentally using a $^{171}Yb^{+}$ ion in a Paul trap[7]. This experimental demonstration of a phase transition in a single two-level system has incited a lot of interest since this opens up an avenue for studying critical phenomena in controlled, small quantum systems.

In CPTs and some QPTs (which require $N \to \infty$), a finite-size scaling (FSS) analysis can be done to extract the critical point and the critical exponents of the transition[1], [9]. While this procedure is inapplicable to the QPTs discussed above since these phase transitions occur at a finite system size, the phase transitions in these paradigmatic light-matter interaction models occur only in the limit $\Omega/\omega_0 \to \infty$ and FSS analysis can be done in Ω/ω_0[2]–[4] instead. In this chapter, however, we propose a different approach to studying such phase transitions. We apply the FSS in Hilbert space method[10]–[15] to the QPT in Quantum Rabi Model. In this approach, the truncation of the system is done not in the physical space but in the Hilbert space. The set of basis states spanning the infinite dimensional Hilbert space is truncated to a finite set and the scaling ansatz is employed in terms of the size of this set. This approach has previously been developed and applied to a single particle in Yukawa potential[11], [13] and the problem of finding electronic structure critical parameters for atomic and molecular systems[10], [12], [14]–[16].

In recent years, digital and analog quantum simulators have emerged as a promising platform for the simulation of quantum phenomena. Quantum simulators have already been used to study phase transitions using the method of partition function zeros[17] and the Kibble-Zurek quench mechanism[18], [19]. In this chapter, a protocol to implement the finite-size scaling method using the Quantum Restricted Boltzmann Machine (QRBM) algorithm to find the critical point of the Quantum Rabi model on a digital quantum simulator has been presented.

4.2 Theory

4.2.1 Quantum Rabi Model

The QRM describes a two-level system interacting with a bosonic field mode. The Hamiltonian is[2],

$$H_{Rabi} = \frac{\Omega}{2}\sigma_z + \omega_0 a^\dagger a - \lambda\sigma_x(a + a^\dagger) \tag{4.1}$$

where we've chosen $\hbar = 1$. Here, σ_z and σ_x are the Pauli Z and X matrices respectively, Ω is the energy separation between the two levels in the system, ω_0 is the frequency of the bosonic mode and λ is the system-atom coupling strength. The parity operator $\Pi = e^{i\pi(a^\dagger a + |\uparrow\rangle\langle\uparrow|)}$ commutes with H_{Rabi}. So, H_{Rabi} has a Z_2 symmetry.

This model has a critical point at $g = 2\lambda/\sqrt{\omega_0\Omega} = g_c = 1$ in the limit $\Omega/\omega_0 \to \infty$[2]. $\Omega/\omega_0 \to \infty$ is analogous to the thermodynamic limit for this case, and in experiments where Ω/ω_0 has to be finite, we'll observe finite-size effects like in any other phase transition[2]. For $g < 1$, the system is in the *normal phase* and the ground state is $\left|\phi^0_{np}(g)\right\rangle = \mathcal{S}[r_{np}(g)]\,|0\rangle\,|\downarrow\rangle$ where $\mathcal{S}[x] = \exp\left[\frac{x}{2}(a^{\dagger 2} - a^2)\right]$ and $r_{np}(g) = -\frac{1}{4}\ln(1 - g^2)$. The rescaled ground state energy and photon number are $e_G(g) = \frac{\omega_0}{\Omega}\langle H_{Rabi}\rangle = -\omega_0/2$ and $n_G(g) = \frac{\omega_0}{\Omega}\left\langle a^\dagger a\right\rangle = 0$ respectively. For $g > 1$, the system is in a *superradiant phase* and the ground state is two-fold degenerate, $\left|\phi^0_{sp}(g)\right\rangle = \mathcal{D}[\pm\alpha_g]\mathcal{S}[r_{sp}(g)]\,|0\rangle\,|\downarrow^\pm\rangle$ here $r_{sp}(g) = -\frac{1}{4}\ln(1 - g^{-4})$ and $\mathcal{D}[\alpha] = \exp\left[\alpha(a^\dagger - a)\right]$. $|\downarrow^\pm\rangle$ is the negative eigenvalue eigenstate of $\frac{1}{2g^2}\sigma_z \pm \frac{2\lambda\alpha_g}{g^2\Omega}\sigma_x$ where $\alpha_g = \sqrt{\frac{\Omega}{4g^2\omega_0}(g^4 - 1)}$. The rescaled ground state energy and photon number are $e_G(g) = \frac{\omega_0}{\Omega}\langle H_{Rabi}\rangle = -\omega_0(g^2 + g^{-2})/4$ and $n_G(g) = \frac{\omega_0}{\Omega}\left\langle a^\dagger a\right\rangle = (g^2 - g^{-2})/4$ respectively.

As shown in Fig. 4.1(a) and (b), d^2e_G/dg^2 is discontinuous at $g = g_c = 1$, indicating a continuous phase transition and $n_G = \frac{\omega_0}{\Omega}\left\langle a^\dagger a\right\rangle$ is an order parameter for this phase transition. In the *normal phase*, n_G is zero whereas in the *superradiant phase*, Z_2 symmetry is spontaneously broken and n_G becomes non-zero.

We can also write effective low-energy Hamiltonians in both the *normal* and the *superradiant phases*. For $g < 1$, H_{Rabi} can be reduced to the following effective Hamiltonian[2],

$$H_{np} = \omega_0 a^\dagger a - \frac{\omega_0 g^2}{4}(a + a^\dagger)^2 - \frac{\Omega}{2}. \tag{4.2}$$

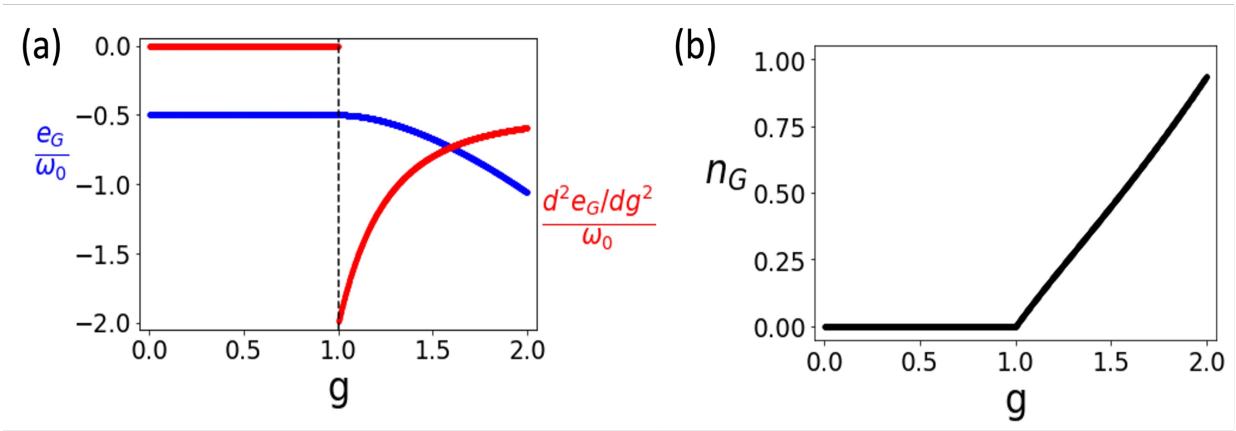

Figure 4.1. Phase Transition in Quantum Rabi Model. (a) The rescaled ground state energy $e_G/\omega_0 = \langle H_{Rabi} \rangle / \Omega$ and $(d^2 e_G/dg^2)/\omega_0$ as functions of g. The discontinuity in $(d^2 e_G/dg^2)/\omega_0$ at $g = g_c = 1$ indicates a continuous phase transition. (b) The order parameter $n_G = \frac{\omega_0}{\Omega} \langle a^\dagger a \rangle$ as a function of g. n_G becomes non-zero when the Z_2 symmetry is spontaneously broken at $g > g_c = 1$.

The system's degrees of freedom have been removed by projecting to $|\downarrow\rangle \langle\downarrow|$, since this is a low energy description. Similarly, for $g > 1$ the effective Hamiltonian can be written as[2],

$$H_{sp} = \omega_0 a^\dagger a - \frac{\omega_0}{4g^4}(a + a^\dagger)^2 - \frac{\Omega}{2}(g^2 + g^{-2}), \tag{4.3}$$

where this time around the Hamiltonian has been projected along $|\downarrow^\pm\rangle \langle\downarrow^\pm|$. In Sec. 4.3, we'll use H_{np} and H_{sp} to find the critical point of the model.

4.2.2 Finite-Size Scaling

The FSS method is widely used to determine the critical points and the critical exponents in phase transitions[1]. To demonstrate the method, consider that we have an infinite $2d$ system that undergoes a classical phase transition at a critical temperature $T = T_c$[9]. Suppose Q is a quantity that becomes singular at $T = T_c$ with some power law behavior

$$Q_\infty(T) \sim |T - T_c|^{-\omega}. \tag{4.4}$$

We can also think of this system as an infinite collection of infinite stripes, where the stripes are infinitely extended along one direction and stacked along the perpendicular direction. Now suppose there are only an N number of stripes. If N is finite, Q should be regular at $T = T_c$ since finite systems cannot have non-analyticities at $T \neq 0$. The singularity at $T = T_c$ should appear only when $N \to \infty$. The finite size scaling hypothesis assumes the existence of a scaling function F_Q such that

$$Q_N(T) \simeq Q_\infty(T) F_Q(N/\xi_\infty(T)), \tag{4.5}$$

where Q_N is the observable Q for a system with N stripes and Q_∞ corresponds to the system in the thermodynamic limit. ξ_∞ is the correlation length for the infinite system. Eq. (4.5) is valid when N is large. The correlation length also diverges as a power law near the critical point,

$$\xi_\infty(T) \sim |T - T_c|^{-\nu}. \tag{4.6}$$

Substituting Eq. (4.4) and (4.6) in Eq. (4.5),

$$Q_N(T) \simeq |T - T_c|^{-\omega} F_Q(N|T - T_c|^\nu). \tag{4.7}$$

Since $Q_N(T)$ should be regular at $T = T_c$, the scaling function should cancel the divergence due to $|T - T_c|^{-\omega}$. Therefore, the scaling function should be of the form $F_Q(x) \sim x^{\omega/\nu}$ as $x \to 0$. We should then have,

$$Q_N(T_c) \sim N^{\omega/\nu}. \tag{4.8}$$

If we define a function $\Delta_Q(T; N, N')$ such that

$$\Delta_Q(T; N, N') = \frac{\log(Q_N(T)/Q_{N'}(T))}{\log(N/N')}, \tag{4.9}$$

then the value of this function at $T = T_c$, $\Delta_Q(T_c; N, N') \simeq \omega/\nu$ is independent of N and N'. Therefore, for three different values N, N' and N'', the curves $\Delta_Q(T; N, N')$ and $\Delta_Q(T; N', N'')$ will intersect at the critical point $T = T_c$. This is how we can locate the critical point using the finite size scaling hypothesis.

We can also find the critical exponents ω and ν. Noting from Eq. (4.4) that

$$\frac{\partial Q_\infty(T)}{\partial T} \sim |T - T_c|^{-(\omega+1)}. \tag{4.10}$$

Therefore, we should have $\Delta_{\partial Q/\partial T}(T_c; N, N') \simeq (\omega+1)/\nu$. Define a new function $\Gamma_\omega(T; N, N')$ such that

$$\Gamma_\omega(T; N, N') = \frac{\Delta_Q(T; N, N')}{\Delta_{\partial Q/\partial T}(T; N, N') - \Delta_Q(T; N, N')}. \tag{4.11}$$

The value of this function at the critical point $\Gamma_\omega(T_c; N, N') \simeq \omega$ is independent of N and N' and gives us the critical exponent ω. Then ν can be determined using

$$\nu \simeq \frac{\omega}{\Delta_Q(T_c; N, N')}. \tag{4.12}$$

As we've already stated in the *Introduction*, this method cannot be used for the kinds of phase transitions we are interested in which occur at a finite system size. However, for such cases we can consider an extension of the approach discussed above[10]–[16]. In this extended approach, instead of truncating the system in the physical space, the system is truncated in the Hilbert space[16]. The FSS ansatz looks exactly the same except that N now represents the size of the set of basis states which spans the truncated Hilbert space[16]. Moreover, the temperature T will be replaced by the parameter g which is being tuned across the critical point. This approach has been shown by Kais and co-workers to work in the case of a particle in Yukawa potential[11], [13] and the calculation of electronic structure critical parameters for atomic and molecular systems[10], [12], [14]–[16].

4.2.3 Quantum Restricted Boltzmann Machine

Solving quantum many-body problem accurately has been a taxing numerical problem since the size of the wavefunction scales exponentially. The idea of taking advantage of the aspects of Machine Learning (ML) related to dimensionality reduction and feature extraction to capture the most relevant information came from the work by Carleo and Troyer [20], which introduced the idea of representing the many-body wavefunction in terms of an

Artificial Neural Network (ANN) to solve for the ground states and time evolution for spin models, with a Restricted Boltzmann Machine (RBM) as the chosen architecture for this ANN. More recently, the critical behavior of the quantum Hall plateau transition based on wavefunctions have been studied in a 2D disordered electron system with the usage of a Convolutional Neural Network (CNN) [21]. However, we focus on using an RBM architecture in this work. An RBM consists of a visible layer and a hidden layer with each neuron in the visible layer connected to all neurons in the hidden layer but the neurons within a layer are not connected to each other. The quantum state is ψ expanded in the basis $|x\rangle$:

$$|\psi\rangle = \sum \psi(x) \, |x\rangle \tag{4.13}$$

The Neural Network Quantum State (NQS) describes the wavefunction $\psi(x)$ to be written as $\psi(x;\theta)$, where θ represents the parameters of the RBM. $\psi(x;\theta)$ is now written in terms of the probability distribution that is obtained from the RBM as follows:

$$\psi(x;\theta) \propto \sum_{\{h\}} \mathrm{e}^{\frac{1}{2}\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z} + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z} h_{\mathrm{j}}} \tag{4.14}$$

where, σ_{i}^{z} is the Pauli z operator at i^{th} site, σ_{i}^{z} and h_{j} take values $\{+1, -1\}$, $\theta = \{a_{\mathrm{i}}, b_{\mathrm{j}}, w_{\mathrm{ij}}\}$ are the trainable bias and weight parameters of the RBM. Using stochastic optimization, the energy $E(\theta)$ is minimized.

This work was extended to obtain the ground states of the Bose-Hubbard model [22] and for the application of quantum state tomography [23].

With the rapid developments in the domains of ML and Quantum Computing (QC), the appetite for integrating ideas in both of these areas has been growing considerably. The last decade has seen a surge in the application of classical ML for quantum matter, wherein these methods have been adopted to benchmark, estimate and study the properties of quantum matter [24]–[27], with recently showing provable classification efficiency in classifying quantum states of matter [28]. RBM based ansätzes have been shown to capture entanglement transitions [29] and using an RBM with local sparse connectivity achieves higher accuracy compared to its dense counterpart when applied to disordered quantum Ising chains [30].

The protocols and algorithms related to ML implementable on a quantum system so called Quantum machine Learning [31] is expected to have the potential of changing the course of fundamental scientific research [32] along with industrial pursuit.

In lieu of today's Noisy Intermediate Scale Quantum (NISQ) devices, the ideas which utilize both classical and quantum resources, such that the part of the problem which has an exponential scaling is implemented on the quantum platform while the rest are dealt with classically, are being carefully investigated for various applications. Such algorithms are known as classical-quantum hybrid algorithms. In the work by Xia and Kais [33], a modified RBM with three layers was introduced, the third layer to account for the sign of the wavefunction, to solve for the ground state energies of molecules. Now, the parametrized wavefunction $\psi(x;\theta)$ is written as a function of $P(x)$ along with a sign function $s(x)$:

$$P(\mathbf{x}) = \frac{\sum_{\{h\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z}h_{\mathrm{j}}}}{\sum_{\mathbf{x}'}\sum_{\{h\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z'} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z'}h_{\mathrm{j}}}} \tag{4.15}$$

$$s(\mathbf{x}) = \tanh\left[(c + \sum_{\mathrm{i}} d_{\mathrm{i}}\sigma_{\mathrm{i}})\right] \tag{4.16}$$

The wavefunction ansatz in terms of the RBM can be expressed as[33]:

$$|\psi\rangle = \sum_{x} \sqrt{P(x)}s(x)\,|x\rangle \tag{4.17}$$

A quantum circuit comprising of a single-qubit (R_y) and multi-qubit y-rotation gates ($C1 - C2 - R_y$) are employed, to sample the Gibbs distribution. The utilization of R_y gates cater to the bias parameter of visible and hidden layers part of the distribution, while $C1 - C2 - R_y$ gates tend to the weights part of the distribution. In the work by Sureshbabu et al. [34], the implementation of such a circuit on IBM-Q devices were shown, wherein a new ancillary qubit is introduced to store the value corresponding to every $C1 - C2 - R_y$ gate (Fig. 4.3). The term n denotes the number of visible qubits and m denotes the number of hidden units. In this formalism, the number of ancillary qubits required are $n \times m$. Starting all the qubits from a $|0\rangle$, the R_y and $C1 - C2 - R_y$ rotations are performed, and a measurement is performed on all the qubits. If all the ancillary qubits are in $|1\rangle$, then the

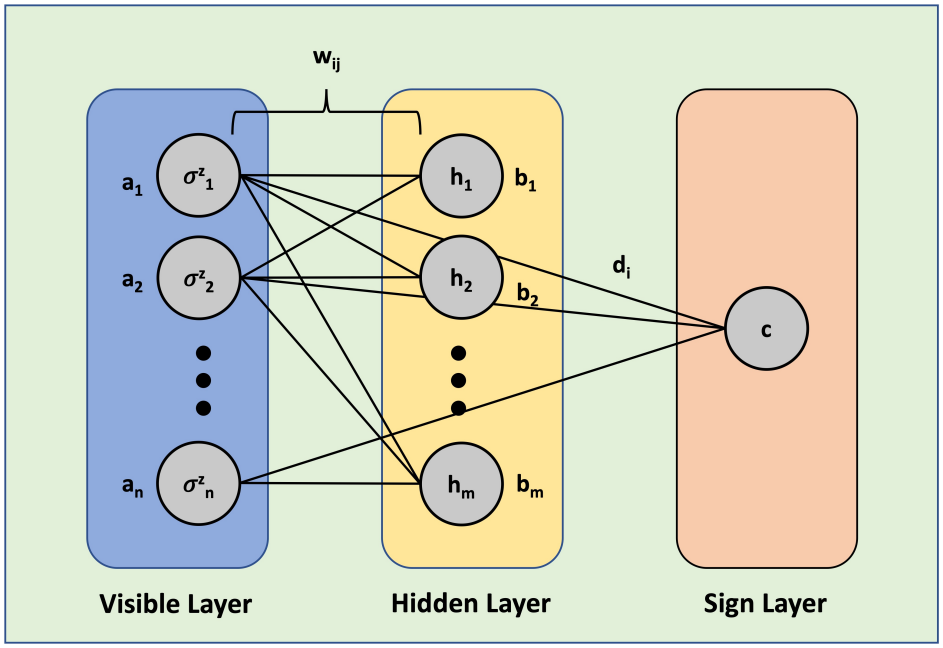

Figure 4.2. Restricted Boltzmann Machine architecture. The first layer is the visible layer with bias parameters denoted by a_{i}. The second layer is the hidden layer with bias parameters denoted by b_{j}. The third layer is the sign layer with bias parameters denoted by c. The weights associated with the connections between the visible neurons and the hidden neurons are designated by w_{ij}. The weights associated with the connections between the visible neurons and the neuron of the sign layer are designated by d_{i}.

sampling is deemed successful and the states corresponding to the first $m+n$ qubits provide the distribution $P(x)$. The joint probability distribution defined over the parameters of the circuit $\theta = \{a, b, w\}$ and a set of $y = \{\sigma^z, h\}$ is given by:

$$P(y,\theta) = \frac{\mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z}h_{\mathrm{j}}}}{\sum_{\{y\}} \mathrm{e}^{\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z'} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z'}h_{\mathrm{j}}}} \tag{4.18}$$

The probability of successful sampling can be improved by rewriting the distribution $P(y,\theta)$ as $Q(y,\theta)$ and setting $k = max(1, \frac{|w_{\mathrm{ij}}|}{2})$[33], [35]:

$$Q(y,\theta) = \frac{\mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z}h_{\mathrm{j}})}}{\sum_{\{y\}} \mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^{z'} + \sum_{\mathrm{j}} b_{\mathrm{j}}h_{\mathrm{j}} + \sum_{\mathrm{ij}} w_{\mathrm{ij}}\sigma_{\mathrm{i}}^{z'}h_{\mathrm{j}})}} \tag{4.19}$$

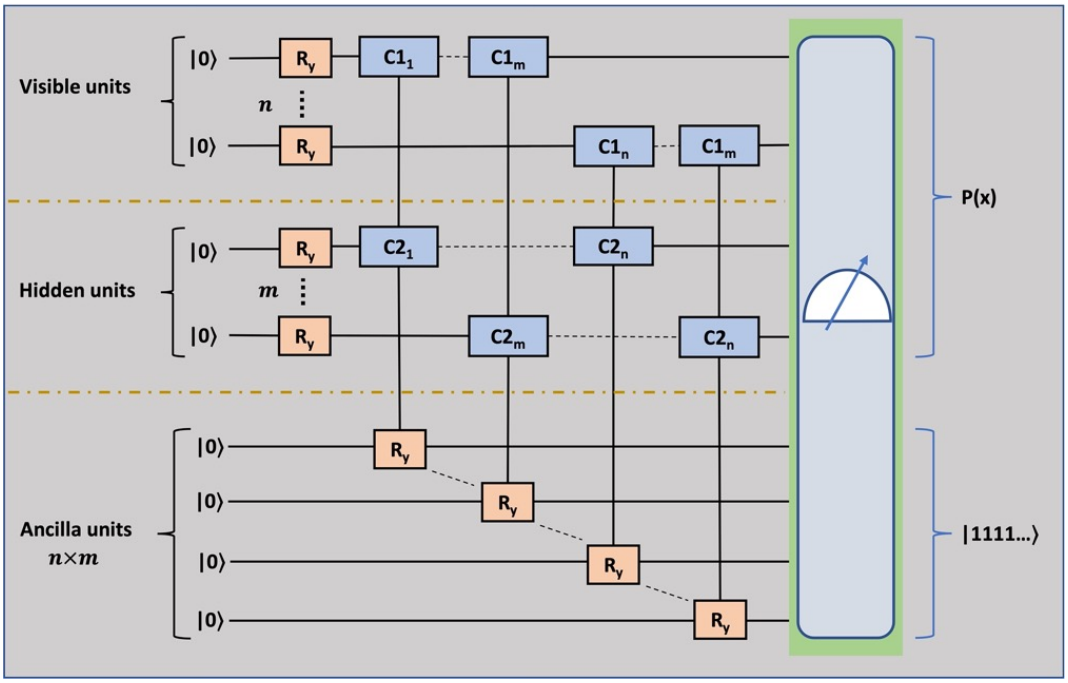

Figure 4.3. The quantum circuit to sample the Gibbs distribution. n is the number of qubits belonging to the visible layer and m is the number of qubits belonging to the hidden layer. There are $m \times n$ ancillary qubits.

Firstly, the QRBM is implemented classically, i.e, the quantum circuit is simulated on a classical computer. This execution caters to the ideal results that can be obtained through the QRBM algorithm. Then, the quantum circuit is implemented on the Digital Quantum Simulator, the *qasm* simulation backend. This simulator is part of the high-performance simulators from IBM-Q. The circuit is realized using IBM's Quantum Information Software Toolkit titled Qiskit [36]. Though no noise model was utilized, as a result of finite sampling, statistical fluctuations in the values of probabilities in observing the circuit in the measurement basis, are present in the obtained results.

Having obtained the distribution $Q(y, \theta)$, the probabilities are raised to the power of k, to get $P(y, \theta)$. Following this, the sign function is computed classically, thereby calculating $|\psi\rangle$. Then, the expectation value for the Hamiltonian H $[\langle\Psi| H |\Psi\rangle]$ is computed to get the energy, which is minimized using gradient descent to obtain the ground state eigenenergy of H.

The resource requirements demanded by this algorithm are quadratic. The number of qubits required are $(m+n)$ to encode the visible and hidden nodes, and $(m \times n)$ to account for the ancillary qubits. Hence, the number of qubits scales as $O(mn)$. The number of R_y gates required are $(m+n)$ and the number of $C1-C2-R_y$ gates required are $(m \times n)$. In addition, each $C1-C2-R_y$ gate requires $6n$ X-gates to account for all the states spanned by the control qubits. Therefore, the number of gates required also scales as $O(mn)$. Obtaining the ground states or minimum eigenvalues of a given matrix using exact diagonalization has a complexity of $\approx \mathrm{j}^3$, with j being the dimension of the column space for the given matrix [37].

4.3 Results

4.3.1 Exact Diagonalization

In this section, we demonstrate the calculation of the critical point of the Quantum Rabi model using the Finite-Size Scaling method. As discussed before, the phase transition in QRM occurs only in the limit $\Omega/\omega_0 \to \infty$. This limit is not straightforward to implement in H_{Rabi} given in Eq. (4.1). Instead, we have considered the effective low-energy Hamiltonians H_{np} and H_{sp} given in Eq. (4.2) and (4.3) respectively. In H_{np} and H_{sp}, Ω is involved only in a constant term which can be removed from the Hamiltonians and the limit $\Omega/\omega_0 \to \infty$ can then be easily imposed.

In H_{np} and H_{sp}, the degrees of freedom of the two-level system have been traced out and the only degrees of freedom we have are those of the bosonic mode. Let's first consider the *normal phase* Hamiltonian H_{np}. The Hilbert space for this Hamiltonian is spanned by the familiar harmonic oscillator number states $\{|0\rangle, |1\rangle, |2\rangle, \ldots\}$. We can truncate the full Hilbert space to an N-dimensional Hilbert space spanned by $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$ to apply the finite-size scaling analysis. In this restricted Hilbert space, the matrix form of $H_{np}^{(N)}$ can be found by using $a|m\rangle = \sqrt{m}\,|m-1\rangle$ and $a^\dagger|m\rangle = \sqrt{m+1}\,|m+1\rangle$. Once we have the matrix form, we can then use the exact diagonalization method to find the ground state of $H_{np}^{(N)}$ with energy $E_{np}^{(N)}$.

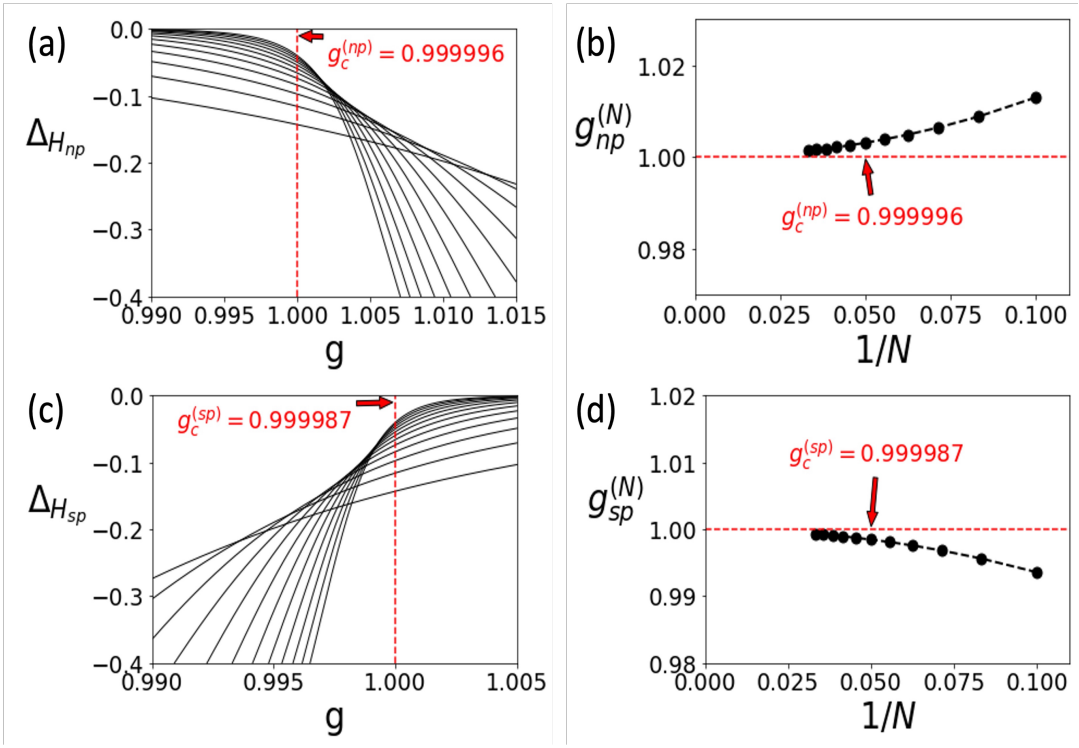

Figure 4.4. Finite-Size Scaling for Quantum Rabi model. We used $N = 8, 10, \ldots, 32$. (a) Graphs of $\Delta_{H_{np}}(g; 8, 10), \Delta_{H_{np}}(g; 10, 12), \ldots, \Delta_{H_{np}}(g; 30, 32)$ as a function of g. (b) Intersection points $g_{np}^{(N)}$ where $\Delta_{H_{np}}(g_{np}^{(N)}; N-4, N-2) = \Delta_{H_{np}}(g_{np}^{(N)}; N-2, N)$, as a function of $1/N$. As $N \to \infty$, $g_{np}^{(N)} \to 0.999996$. So, $g_c^{(np)} = 0.999996$. (c) Graphs of $\Delta_{H_{sp}}(g; 8, 10), \Delta_{H_{sp}}(g; 10, 12), \ldots, \Delta_{H_{sp}}(g; 30, 32)$ as a function of g. (d) Intersection points $g_{sp}^{(N)}$ where $\Delta_{H_{sp}}(g_{sp}^{(N)}; N-4, N-2) = \Delta_{H_{sp}}(g_{sp}^{(N)}; N-2, N)$, as a function of $1/N$. As $N \to \infty$, $g_{sp}^{(N)} \to 0.999987$. So, $g_c^{(sp)} = 0.999987$.

Consider the scaling law for the ground state energy in the vicinity of the critical point $g = g_c$,

$$E(g) \sim |g - g_c|^{\alpha}. \tag{4.20}$$

Here E is the ground state energy. We slightly modify the formula in Eq. (4.9) to take into account the difference in the signs of the exponents in Eq. (4.4) and (4.20). The new formula with $Q = E$ is,

$$\Delta_{H_{np}}(g; N, N') = \frac{\log(E_{np}^{(N)}(g)/E_{np}^{(N')}(g))}{\log(N'/N)}, \tag{4.21}$$

We plot the curves $\Delta_{H_{np}}(g; N, N+2)$ for $N = 8, 10, \ldots, 30$ in Fig. 4.4(a). We then plot the intersection points $g_{np}^{(N)}$ of the curves $\Delta_{H_{np}}(g; N-4, N-2)$ and $\Delta_{H_{np}}(g; N-2, N)$ as a function

of N as shown in Fig. 4.4(b). To find the limit of $g_{np}^{(N)}$ as $N \to \infty$, we used the Bulirsch-Stoer algorithm (see Appendix C.1). The limit was calculated to be $g_{np}^{(N)} \to 0.999996$. So $g_c^{(np)} = 0.999996$.

In a similar way, we then consider H_{sp}. The curves $\Delta_{H_{sp}}(g; N, N+2)$ are plotted in Fig. 4.4(c) for $N = 8, 10, \ldots, 30$ and the intersection points $g_{sp}^{(N)}$ are plotted in Fig. 4.4(d) as a function of N. In this case, the extrapolation to $N \to \infty$ gives the critical value $g_c^{(sp)} = 0.999987$. Both the calculated values of $g_c^{(np)}$ and $g_c^{(sp)}$ are very close to the exact value $g_c = 1$.

4.3.2 Quantum Restricted Boltzmann Machine

Now we illustrate the implementation of the FSS method using the QRBM algorithm. The results are shown in Fig. 4.5. Fig. 4.5(a) and Fig. 4.5(c) show the results for H_{np} and H_{sp} using the classical implementation of the algorithm respectively. Whereas, Fig. 4.5(b) and Fig. 4.5(d) correspond to the results for H_{np} and H_{sp} when the algorithm is implemented using the *qasm* simulator from IBM-Q respectively. The QRBM algorithm is run for $N = 8, 10, 12, 14, 16$.

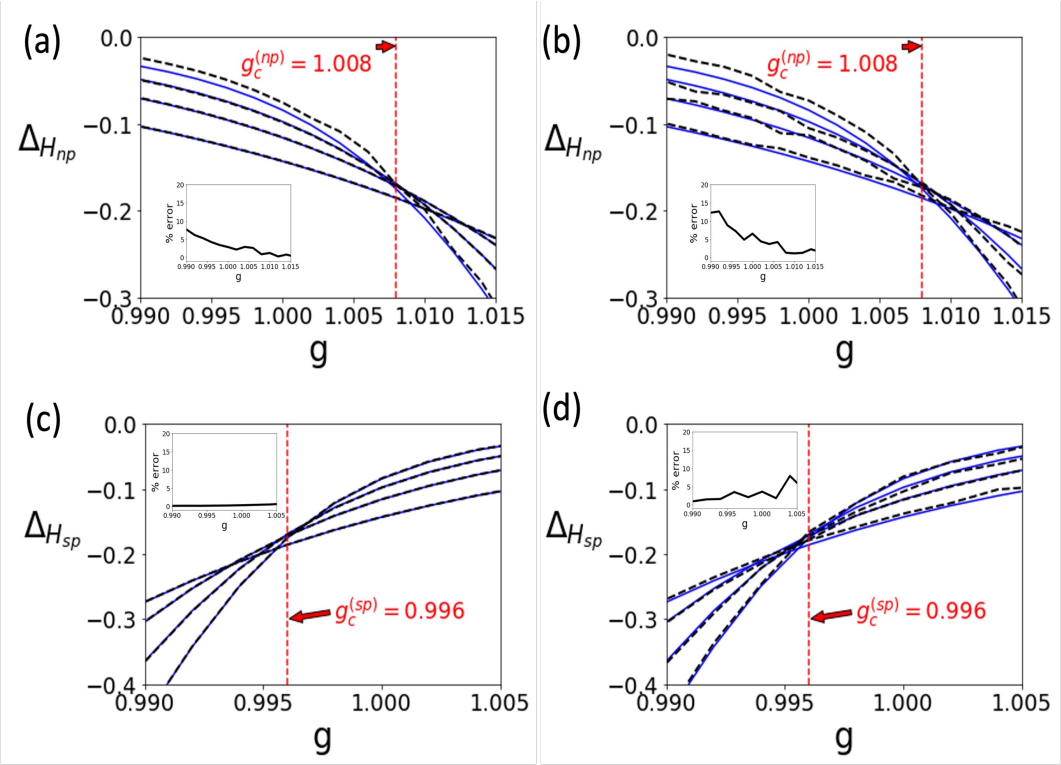

Figure 4.5. QRBM Implementation of FSS for QRM. The light blue line represents results obtained from exact diagonalization and dashed black line represents QRBM results. (a) Classical implementation of QRBM corresponding to normal phase, graphs of $\Delta_{H_{np}}(g;8,10)$, $\Delta_{H_{np}}(g;10,12)$, $\ldots,\Delta_{H_{np}}(g;14,16)$ as a function of g. (b) QRBM implemented on *qasm* simulator corresponding to normal phase, graphs of $\Delta_{H_{np}}(g;8,10)$, $\Delta_{H_{np}}(g;10,12)$, $\ldots$, $\Delta_{H_{np}}(g;14,16)$ as a function of g. The $g_c^{(np)}$ in both cases is calculated to be 1.008. (c) Classical implementation of QRBM corresponding to superradiant phase, graphs of $\Delta_{H_{sp}}(g;8,10)$, $\Delta_{H_{sp}}(g;10,12)$, $\ldots$, $\Delta_{H_{sp}}(g;14,16)$ as a function of g. (b) QRBM implemented on *qasm* simulator corresponding to superradiant phase, graphs of $\Delta_{H_{sp}}(g;8,10)$, $\Delta_{H_{sp}}(g;10,12)$, $\ldots$, $\Delta_{H_{sp}}(g;14,16)$ as a function of g. The $g_c^{(sp)}$ in both the cases is calculated to be 0.996. The inset plots display the mean percentage error between the exact diagonalization results and QRBM results.

For the case of N=8, the number of qubits associated with the visible nodes equal 3, the number of qubits associated with the hidden nodes equal 3, and 9 ancillary qubits were used. The quantum circuit consists of 6 R_y gates associated with the bias parameters, 9 $C1-C2-R_y$ gates associated with the weights. Since, each $C1-C2-R_y$ gate requires 6 X-gates, a total of 54 X-gates were used. For the case of N=10,..,16, the number of qubits associated with the visible nodes equal 4, the number of qubits associated with the hidden nodes equal 4, and 16 ancillary qubits were used. The quantum circuit consists of 8 R_y gates

associated with the bias parameters, 16 $C1-C2-R_y$ gates associated with the weights. Since, each $C1-C2-R_y$ gate requires 6 X-gates, a total of 96 X-gates were used.

Starting from random initialization, all parameters are updated via gradient descent. A learning rate of 0.01 was chosen and the algorithm is run for around 30,000 iterations. In order to assist with the convergence to the minimum eigenenergies, warm starting is employed. The method of warm starting is essentially initializing the parameters of the current point with the parameters of a previously converged point of calculation, which helps in avoiding the convergence to a local minimum.

The black curves plotted in the insets in Fig. 4.5 represent the deviation of the QRBM results (black dashed curves) from the exact diagonalization results (blue solid curves). They were calculated using the average of the quantity $\left|\Delta^{(ED)}(g)-\Delta^{(QRBM)}(g)/\Delta^{(ED)}(g)\right|\times 100$ over all the four curves. An enlarged version of the error plots can be found in the *Appendix C* section. For each case, the overall error close to $g=1.000$ is not more than $\sim 5\%$ which implies convergence to the right result. Moreover, for the case of H_{sp}, we notice that the error is very small for the classical implementation i.e. $\sim< 1\%$ throughout the range of the graph.

The critical point using H_{np} was found to be $g_c^{(np)}=1.008$ for both the classical and *qasm* implementations. Similarly, the critical point for the case of H_{sp} was found to be $g_c^{(sp)}=0.996$ for both the classical and *qasm* implementations. Here we notice that although, the convergence for the data obtained from both the classical and *qasm* implementations turns out to be the same for both H_{np} and H_{sp}, such a perfect match appears to be somewhat coincidental. In Appendix C.1, we have explained the Bulirsch-Stoer algorithm which sets the criteria used to deduce these convergence results. The convergence plots have been added to the *Appendix C* section.

4.4 Discussion and Outlook

In this chapter, Finite-Size Scaling in the Hilbert Space approach has been used to calculate the critical point of the Quantum Rabi Model. The low-energy effective Hamiltonians for both the normal and superradiant phases respectively have been utilized to show that

the critical point is $g_c \approx 1$. The original FSS approach in which the truncation is done in the physical space has been widely used to calculate critical points and critical exponents since its inception. However, that approach was not applicable to Quantum Phase Transitions which occur at a finite system size. With the rise in interest in QPTs occurring in these finite-size systems, our approach provides a natural extension of the original FSS method to study such phase transitions. To our knowledge, this is the first time this approach has been used to study a QPT in a light-matter interaction system.

A recipe for the implementation of this method on a universal quantum computer using the Quantum Restricted Boltzmann Machine algorithm has been provided. It was shown that results obtained from the classical gate simulation match those obtained from the IBM-Q's *qasm* simulator. Such an implementation scales quadratically while the exact diagonalization scales cubically in the best case and exponentially in the worst case. Looking forward, we are interested in applying this approach to other QPTs such as the QPT in anisotropic QRM. We would also like to use our method to calculate the critical exponents in addition to the critical points in these phase transitions. It would also be interesting to see if this approach can be used to predict any new phase transition for some other non-integrable model.

Another very promising research direction is to implement the FSS method for phase transitions in classically intractable many-body models such as exotic electronic and magnetic systems. These include general quantum materials, for example where Coulomb potential leads to a gapped spectrum in energy, including indirect band-gap semiconductors in the thermodynamic limit. Conventionally speaking, it might be necessary to resort to the original finite-size scaling in the physical space approach for these systems since they exhibit criticality only in the limit $N \to \infty$. However, the ground state of an appropriately truncated Hamiltonian could be deduced using the QRBM algorithm as shown in the paper towards efficient implementation on a digital quantum simulator. A simile can also be drawn between a many-body bulk gap separating a continuum of excited states from the ground state manifold to the gapped Rabi model discussed in this paper. Such an approach can be useful in emergent topological systems, such as in Weyl semimetals, 1-D Kitaev spin chains, quantum spin liquids, and others, on which there is a tremendous explosion of interest [38]–[43]. Topological phase transitions are devoid of any conventional order parameter and a

quantum solution deriving from the approach outlined in this paper can help us bypass resource and scaling limitations of DMRG and exact diagonalization approaches to calculate the critical point and the critical exponents.

References

[1] N. Goldenfeld, *Lectures on Phase Transitions and the Renormalization Group* (Frontiers in Physics). Addison-Wesley Publishing Company, 1992.

[2] M.-J. Hwang, R. Puebla, and M. B. Plenio, "Quantum phase transition and universal dynamics in the rabi model," *Phys. Rev. Lett.*, vol. 115, p. 180 404, 18 Oct. 2015. DOI: 10.1103/PhysRevLett.115.180404. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.115.180404.

[3] M.-J. Hwang and M. B. Plenio, "Quantum phase transition in the finite jaynes-cummings lattice systems," *Phys. Rev. Lett.*, vol. 117, p. 123 602, 12 Sep. 2016. DOI: 10.1103/PhysRevLett.117.123602. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.117.123602.

[4] M. Liu, S. Chesi, Z.-J. Ying, X. Chen, H.-G. Luo, and H.-Q. Lin, "Universal scaling and critical exponents of the anisotropic quantum rabi model," *Phys. Rev. Lett.*, vol. 119, p. 220 601, 22 Nov. 2017. DOI: 10.1103/PhysRevLett.119.220601. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.119.220601.

[5] M.-J. Hwang, P. Rabl, and M. B. Plenio, "Dissipative phase transition in the open quantum rabi model," *Phys. Rev. A*, vol. 97, p. 013 825, 1 Jan. 2018. DOI: 10.1103/PhysRevA.97.013825. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.97.013825.

[6] Y.-F. Xie, X.-Y. Chen, X.-F. Dong, and Q.-H. Chen, "First-order and continuous quantum phase transitions in the anisotropic quantum rabi-stark model," *Phys. Rev. A*, vol. 101, p. 053 803, 5 May 2020. DOI: 10.1103/PhysRevA.101.053803. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.053803.

[7] M.-L. Cai, Z.-D. Liu, W.-D. Zhao, *et al.*, "Observation of a quantum phase transition in the quantum Rabi model with a single trapped ion," *Nat Commun*, vol. 12, p. 1126, 2021. DOI: 10.1038/s41467-021-21425-8.

[8] P. Forn-Díaz, L. Lamata, E. Rico, J. Kono, and E. Solano, "Ultrastrong coupling regimes of light-matter interaction," *Rev. Mod. Phys.*, vol. 91, p. 025 005, 2 Jun. 2019. DOI: 10.1103/RevModPhys.91.025005. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.91.025005.

[9] B. Derrida and L. D. Seze, "Application of the phenomenological renormalization to percolation and lattice animals in dimension 2," in *Finite-Size Scaling*, ser. Current Physics–Sources and Comments, J. L. CARDY, Ed., vol. 2, Elsevier, 1988, pp. 275–283. DOI: https://doi.org/10.1016/B978-0-444-87109-1.50024-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444871091500248.

[10] J. P. Neirotti, P. Serra, and S. Kais, "Electronic structure critical parameters from finite-size scaling," *Phys. Rev. Lett.*, vol. 79, pp. 3142–3145, 17 Oct. 1997. DOI: 10.1103/PhysRevLett.79.3142. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.3142.

[11] P. Serra, J. P. Neirotti, and S. Kais, "Finite-size scaling approach for the schrödinger equation," *Phys. Rev. A*, vol. 57, R1481–R1484, 3 Mar. 1998. DOI: 10.1103/PhysRevA.57.R1481. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.57.R1481.

[12] P. Serra, J. P. Neirotti, and S. Kais, "Electronic structure critical parameters for the lithium isoelectronic series," *Phys. Rev. Lett.*, vol. 80, pp. 5293–5296, 24 Jun. 1998. DOI: 10.1103/PhysRevLett.80.5293. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.80.5293.

[13] P. Serra, J. P. Neirotti, and S. Kais, "Finite size scaling in quantum mechanics," *J. Phys. Chem. A*, vol. 102, pp. 9518–9522, 1998.

[14] S. Kais and P. Serra, "Quantum critical phenomena and stability of atomic and molecular ions," *International Reviews in Physical Chemistry*, vol. 19, no. 1, pp. 97–121, 2000. DOI: 10.1080/014423500229873. eprint: https://doi.org/10.1080/014423500229873. [Online]. Available: https://doi.org/10.1080/014423500229873.

[15] Q. Shi and S. Kais, "Finite size scaling for critical parameters of simple diatomic molecules," *Molecular Physics*, vol. 98, no. 19, pp. 1485–1493, 2000. DOI: 10.1080/00268970009483354. eprint: https://doi.org/10.1080/00268970009483354. [Online]. Available: https://doi.org/10.1080/00268970009483354.

[16] S. Kais and P. Serra, "Finite size scaling for atomic and molecular systems," *Advances in Chemical Physics, Volume 125*, pp. 1–100, 2003.

[17] A. Francis, D. Zhu, C. H. Alderete, *et al.*, "Many-body thermodynamics on quantum computers via partition function zeros," *Science Advances*, vol. 7, no. 34, eabf2447, 2021. DOI: 10.1126/sciadv.abf2447. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abf2447. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.abf2447.

[18] A. Keesling, A. Omran, H. Levine, *et al.*, "Quantum kibble–zurek mechanism and critical dynamics on a programmable rydberg simulator," *Nature*, vol. 568, pp. 207–211, 2019. DOI: 10.1038/s41586-019-1070-1.

[19] M. Dupont and J. E. Moore, "Quantum criticality using a superconducting quantum processor," *arXiv:2109.10909*, 2021.

[20] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[21] Z. Li, M. Luo, and X. Wan, "Extracting critical exponents by finite-size scaling with convolutional neural networks," *Physical Review B*, vol. 99, no. 7, p. 075 418, 2019.

[22] H. Saito, "Solving the bose–hubbard model with machine learning," *Journal of the Physical Society of Japan*, vol. 86, no. 9, p. 093 001, 2017.

[23] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography," *Nature Physics*, vol. 14, no. 5, pp. 447–450, 2018.

[24] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, pp. 431–434, 2017.

[25] D.-L. Deng, X. Li, and S. D. Sarma, "Machine learning topological states," *Physical Review B*, vol. 96, no. 19, p. 195 145, 2017.

[26] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, no. 7715, pp. 547–555, 2018.

[27] G. Carleo, I. Cirac, K. Cranmer, *et al.*, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, no. 4, p. 045 002, 2019.

[28] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, "Provably efficient machine learning for quantum many-body problems," *arXiv preprint arXiv:2106.12627*, 2021.

[29] R. Medina, R. Vasseur, and M. Serbyn, "Entanglement transitions from restricted boltzmann machines," *Physical Review B*, vol. 104, no. 10, p. 104 205, 2021.

[30] S. Pilati and P. Pieri, "Simulating disordered quantum ising chains via dense and sparse restricted boltzmann machines," *Physical Review E*, vol. 101, no. 6, p. 063 308, 2020.

[31] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, ISSN: 1476-4687. DOI: 10.1038/nature23474. [Online]. Available: http://dx.doi.org/10.1038/nature23474.

[32] M. Sajjan, J. Li, R. Selvarajan, *et al.*, "Quantum computing enhanced machine learning for physico-chemical applications," *arXiv preprint arXiv:2111.00851*, 2021.

[33] R. Xia and S. Kais, "Quantum machine learning for electronic structure calculations," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[34] S. H. Sureshbabu, M. Sajjan, S. Oh, and S. Kais, "Implementation of quantum machine learning for electronic structure calculations of periodic systems on quantum computing devices," *Journal of Chemical Information and Modeling*, 2021.

[35] M. Sajjan, S. H. Sureshbabu, and S. Kais, "Quantum machine-learning for eigenstate filtration in two-dimensional materials," *Journal of the American Chemical Society*, vol. 143, no. 44, pp. 18 426–18 445, 2021, PMID: 34705449. DOI: 10.1021/jacs.1c06246.

[36] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, *et al.*, "Qiskit: An open-source framework for quantum computing," *Accessed on: Mar*, vol. 16, 2019.

[37] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[38] K. J. Satzinger, Y.-J. Liu, A. Smith, *et al.*, "Realizing topologically ordered states on a quantum processor," *Science*, vol. 374, no. 6572, pp. 1237–1241, 2021. DOI: 10.1126/science.abi8378. eprint: https://www.science.org/doi/pdf/10.1126/science.abi8378. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.abi8378.

[39] A. Kitaev, "Anyons in an exactly solved model and beyond," *Annals of Physics*, vol. 321, no. 1, pp. 2–111, 2006, January Special Issue, ISSN: 0003-4916. DOI: https://doi.org/10.1016/j.aop.2005.10.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003491605002381.

[40] Y.-J. Liu, K. Shtengel, A. Smith, and F. Pollmann, "Methods for simulating string-net states and anyons on a digital quantum computer," *arXiv:2110.02020*, 2021.

[41] X. Xiao, J. K. Freericks, and A. F. Kemper, "Determining quantum phase diagrams of topological Kitaev-inspired models on NISQ quantum hardware," *Quantum*, vol. 5, p. 553, Sep. 2021, ISSN: 2521-327X. DOI: 10.22331/q-2021-09-28-553. [Online]. Available: https://doi.org/10.22331/q-2021-09-28-553.

[42] X.-G. Wen, "Colloquium: Zoo of quantum-topological phases of matter," *Rev. Mod. Phys.*, vol. 89, p. 041 004, 4 Dec. 2017. DOI: 10.1103/RevModPhys.89.041004. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.89.041004.

[43] M. Z. Hasan and C. L. Kane, "Colloquium: Topological insulators," *Rev. Mod. Phys.*, vol. 82, pp. 3045–3067, 4 Nov. 2010. DOI: 10.1103/RevModPhys.82.3045. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.82.3045.

5. PARAMETER SETTING IN QUANTUM APPROXIMATE OPTIMIZATION OF WEIGHTED PROBLEMS

The contents of this chapter are adapted from the article 'Sureshbabu, Shree Hari, et al. "Parameter Setting in Quantum Approximate Optimization of Weighted Problems." arXiv preprint arXiv:2305.15201 (2023)'.

5.1 Introduction

Quantum computers are widely believed to be able to provide computational speedups for various problems of relevance to science and industry [1], [2]. Combinatorial optimization is a domain that is very likely to benefit from quantum computing due to the ubiquity of hard optimization problems. Quantum Approximate Optimization Algorithm (QAOA) [3]–[5] is a leading candidate quantum heuristic algorithm for optimization. QAOA solves optimization problems by preparing a parameterized quantum state using a circuit consisting of layers of alternating operators, wherein each operator has a free parameter associated with it. The two operators are commonly referred to as the *phase operator* and *mixer operator*, respectively. QAOA has been shown to achieve better scaling than state-of-the-art classical solvers for finding exact solutions of k-SAT [6] and to achieve approximation ratios competitive with those of the best known classical algorithms for the unweighted MaxCut problem [7], [8]. When the mixer is different from the transverse field used in [3], [4], the algorithm is sometimes referred to as the Quantum Alternating Operator Ansatz [5]. Throughout the paper, we will use QAOA to refer to both without making a distinction.

One of the central challenges of applying QAOA to practically-relevant problems is the need to set the QAOA parameters. The parameter setting is particularly challenging for problems with objectives containing non-integer coefficients (weights) on the binary variables. The non-integer weights lead to the optimization landscape becoming non-periodic and in general hard to optimize [9]–[11]. While parameter setting schemes leveraging analytically-optimal QAOA parameters exist, they are only available in a limited number of cases. For example, in the infinite-size limit, optimal QAOA parameters are known for unweighted MaxCut on large-girth regular [7], Erdős-Rényi [12] and other [12] random graph ensembles,

as well as for the Sherrington-Kirkpatrick (SK) model [13]. While a similar parameter setting scheme has been proposed and numerically validated for weighted MaxCut [9], no analytical results are known for weighted MaxCut or other problems with non-integer eigenvalues.

In this work, we develop parameter setting heuristics for QAOA applied to a broad class of weighted problems. Our starting point is QAOA with the transverse-field mixer applied to the weighted MaxCut problem on large-girth regular graphs. We begin with $p = 1$ and derive globally-optimal parameters for QAOA applied to graphs with edge weights drawn i.i.d. from the exponential distribution for any graph size and from an arbitrary distribution in the infinite-size limit. Our analysis rigorously proves the folklore notion that for problems with non-periodic QAOA energy landscapes, the first local optimum near zero contains globally-optimal parameters in the average case [9]–[11]. We then analyze the case of $p \geq 1$ and connect QAOA on weighted MaxCut problems to QAOA on unweighted MaxCut by proving the following Theorem:

Theorem 3 (Informal)**.** *Consider QAOA with depth p and a regular graph G with girth $> 2p + 1$ and i.i.d. random edge weights drawn from $\mathbf{w}$. Then the QAOA objective for weighted MaxCut on G at parameters $\left(\boldsymbol{\beta}, \frac{\boldsymbol{\gamma}}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}\right)$ is, up to a global scaling factor, equal to the QAOA objective for the corresponding MaxCut problem on the unweighted version of the same graph at parameters $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ on average in the infinite-size limit. Here the average is taken over the random choice of edge weights, and $\mathbb{E}_{\mathbf{w}}[w^2]$ is the second moment of the distribution from which the edge weights are drawn.*

This result proves that the parameters that are optimal for unweighted MaxCut can be rescaled to be optimal for weighted MaxCut. As a consequence, it establishes a rule for setting parameters in QAOA for weighted MaxCut using the parameters for the unweighted case obtained previously in Ref. [7]. As MaxCut is deeply connected to the SK model [14], we briefly discuss a "weighted" modification of the SK model obtained by drawing couplings in the SK model from $\mathcal{N}(\mu, \sigma^2)$ instead of $\mathcal{N}(0, 1)$. Here μ may depend on the problem size N. We call this modification "biased SK" and show that it behaves trivially in the infinite-size limit, unless $\mu = \mu(N)$ decays to zero with increasing N.

We evaluate the parameter setting rule implied by Theorem 3 numerically outside of its theoretical assumptions by applying QAOA with $p \in \{1, 2, 3\}$ to MaxCut on a dataset of 34,701 weighted regular and non-regular graphs. We observe that our scheme outperforms the previously proposed approach of Ref. [9]. On average, across all graphs, values of p and edge-weight distributions, QAOA with parameters obtained using our scheme achieves solutions that are only 1.1 percentage points (p.p.) away from optimal, improving upon the 3.5 p.p. obtained using the technique presented in the prior work [9]. Moreover, the disparity from the solutions obtained using optimized parameters is reduced by a factor of three (from 3.6 p.p. to 1.0 p.p.) when the edge weights are drawn from the exponential distribution, and by a factor of ≈ 6 with the Cauchy distribution (from 20.7 to 3.3 p.p.).

5.2 Background

We begin by briefly reviewing the Quantum Approximate Optimization Algorithm, the parameter setting schemes for it, and the weighted MaxCut problem.

5.2.1 Quantum Approximate Optimization Algorithm

Consider the problem of optimizing some objective function $\mathcal{C}(\boldsymbol{x})$ defined on the n-dimensional Boolean cube that is encoded on n qubits by a diagonal Hamiltonian $C = \mathrm{diag}(\mathcal{C}(\boldsymbol{x}))$. Quantum Approximate Optimization Algorithm (QAOA)[3], [4] is a hybrid quantum-classical algorithm that approximately solves optimization problems by preparing a parameterized circuit such that upon measuring it, an approximate solution to the optimization problem is obtained. The QAOA circuit consists of layers of alternating unitaries, $\mathrm{e}^{-\mathrm{i}\gamma C}$ and $\mathrm{e}^{-\mathrm{i}\beta B}$, where C is the Hamiltonian corresponding to the optimization problem and B is the mixer Hamiltonian. Common choices of the mixer Hamiltonian B include the transverse field ($B = \sum_{\mathrm{j}} \mathrm{X}_{\mathrm{j}}$) for unconstrained problems and the XY mixer ($B = \frac{1}{2} \sum_{\mathrm{j},k} (\mathrm{X}_{\mathrm{j}}\mathrm{X}_k + \mathrm{Y}_{\mathrm{j}}\mathrm{Y}_k)$) for problems with an equality constraint on the Hamming weight.

The QAOA state with p layers is given by

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = \mathrm{e}^{-\mathrm{i}\beta_p B}\mathrm{e}^{-\mathrm{i}\gamma_p C} \ldots \mathrm{e}^{-\mathrm{i}\beta_1 B}\mathrm{e}^{-\mathrm{i}\gamma_1 C}|s\rangle \tag{5.1}$$

where $|s\rangle$ is the initial state and $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are free parameters chosen by a classical routine. We discuss the strategies for setting the parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ in Sec. 5.2.2 below.

The figure of merit that we use to evaluate the QAOA performance is the expected solution quality given by the "QAOA energy":

$$\langle C(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle := \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle = \sum_{\boldsymbol{z} \in \{0,1\}^n} \mathcal{C}(\boldsymbol{z}) \Pr(\boldsymbol{z}), \tag{5.2}$$

where $\Pr(\boldsymbol{z})$ is the probability of observing $\boldsymbol{z}$ when measuring all qubits of $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$.

5.2.2 Parameter setting strategies for QAOA

Multiple techniques have been proposed for obtaining high-quality parameters for QAOA. While the parameters can be obtained by direct optimization of the objective (5.2) using a preferred optimization method [15]–[21], this procedure is typically computationally expensive [22]–[25]. The cost of finding parameters can be significantly reduced by leveraging the apparent problem-instance independence of the optimal QAOA parameters [26], [27]. More straightforwardly, optimized parameters from one instance can be used directly as high-quality parameters for another instance from the same problem class [9], [18], [28]–[30]. A machine learning model can be trained that would leverage the concentration to accurately predict the parameters [31]–[36]. Optimal parameters can be derived exactly in certain analytically tractable cases, such as triangle-free regular graphs at $p = 1$ [37].

In certain cases, i.e. in the infinite-size limit of a given problem, a closed-form iteration can be derived for the QAOA objective, Equation (5.2), at constant p. Then parameters can be optimized in the infinite-size limit and used for finite-size instances. This has been demonstrated for the Sherrington-Kirkpatrick model [38] and for MaxCut on random graphs [7], [12]. The goal of this work is to extend these results to weighted problems.

5.2.3 MaxCut problem

For an undirected graph $G = (V, E)$ with weights $w_{uv} = w_{\{u,v\}}$ assigned to edges $\{u, v\} \in E$, the goal of MaxCut is to partition the set of nodes V into two disjoint subsets, such that

the total sum of weights of the edges spanning both partitions is maximized. We refer to this problem as weighted MaxCut in the general case and as unweighted MaxCut when $w_{uv} = 1$ for all $\{u, v\} \in E$.

For the weighted MaxCut problem the objective function is given by

$$\mathcal{C}(\boldsymbol{z}) = \frac{1}{2} \sum_{\{u,v\} \in E} w_{uv}(1 - z_u z_v), \tag{5.3}$$

where $z_u \in \{-1, 1\}$ are the variables to be optimized and w_{uv} are sampled from the desired probability distribution. The MaxCut objective is encoded on qubits by the Hamiltonian

$$C = \frac{1}{2} \sum_{\{u,v\} \in E} w_{uv}(\mathrm{I} - \mathrm{Z}_u \mathrm{Z}_v), \tag{5.4}$$

where Z_u and Z_v are Pauli-Z operators applied to the uth and vth qubits, respectively.

For unweighted graphs, the cut fraction is defined as the ratio between the number of edges in a cut and the total number of edges in the graph. For a random unweighted $(D + 1)$-regular graph, the optimal cut fraction is, with high probability, given by

$$\frac{1}{2} + \frac{\Pi_*}{\sqrt{D}} + o\left(\frac{1}{\sqrt{D}}\right), \tag{5.5}$$

where $\Pi_* \approx 0.7632$ is the Parisi value [14].

5.3 Parameter setting scheme for QAOA on weighted problems

Our parameter setting scheme is motivated by the observation, formalized in Sec. 5.4, that in many cases the QAOA energy landscape for weighted MaxCut can be rescaled to match that of unweighted MaxCut for arbitrary p. In the case of weighted MaxCut, this gives an explicit parameter setting rule. In the case of a general objective, we use the same observation to propose a rescaling rule that makes the QAOA energy landscape easier to optimize. We validate our scheme numerically for both cases in Section 5.6.

5.3.1 Weighted MaxCut

The proposed procedure is as follows. First, rescale the edge weights in the graph following

$$w_{uv} \to \frac{w_{uv}}{\sqrt{\frac{1}{|E|}\sum_{\{u,v\}\in E} w_{uv}^2}} \tag{5.6}$$

Second, use the parameter setting rule for the corresponding unweighted graph.

As an example of parameter setting rule for unweighted graphs to be used in the second step, one can use the parameters $\boldsymbol{\beta}^{\text{inf}}, \boldsymbol{\gamma}^{\text{inf}}$ optimized for large-girth regular graphs in the infinite-size limit [7, Tables 4 and 5] and follow the rescaling procedure therein, which we include here for completeness: $\boldsymbol{\beta} = \boldsymbol{\beta}^{\text{inf}}$, $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{\text{inf}}/\sqrt{D}$. Here D is the average degree of the graph. Alternatively, the procedure from Ref. [39] can be used. For small D and p, higher quality results may be obtained by taking inspiration from the explicit formula of Ref. [37] and setting $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{\text{inf}} \arctan \frac{1}{\sqrt{D-1}}$.

As an optional third step, the quality of the parameters can be improved further by running a local optimizer with a small initial step from the parameters obtained in the second step.

5.3.2 General objective

For a general objective function and QAOA with an arbitrary mixer (e.g., constraint-preserving), analytical results are not available. At the same time, we can use the intuition from MaxCut to rescale the QAOA objective to make the geometry of the landscape more amenable to optimization. Specifically, if the objective f is given by a degree-k polynomial over spins $\boldsymbol{z} \in \{-1, 1\}^n$:

$$f(\boldsymbol{z}) = \sum_{\{u_1,\ldots,u_k\}} w^{(k)}_{u_1\ldots u_k} z_{u_1} \ldots z_{u_k} + \ldots + \sum_u w^{(1)}_u z_u, \tag{5.7}$$

our first step is to divide the objective by

$$\sqrt{\frac{1}{|E_k|}\sum_{\{u_1,\ldots,u_k\}} (w^{(k)}_{u_1,\ldots,u_k})^2 + \ldots + \frac{1}{|E_1|}\sum_u (w^{(1)}_u)^2}, \tag{5.8}$$

where E_{i} is the set of i-way hyperedges, i.e. the number of terms of order i.

In the second step, parameter optimization is performed as usual.

This scaling is inspired by the observation that our results on weighted MaxCut generalize to problems with higher-order (higher than quadratic) terms; see Remark 3. In Section 5.6 we demonstrate the power of this simple procedure using the example of mean-variance portfolio optimization with a budget constraint enforced by the XY-mixer.

5.4 Analytical results for QAOA on weighted MaxCut

We now present the analytical results for QAOA applied to weighted MaxCut on large-girth regular graphs with i.i.d. edge weights. We begin by analyzing $p = 1$ in Section 5.4.1. QAOA energy for $p = 1$ is given by a simple trigonometric formula derived in [40, Theorem 7]. We use this formula to derive globally-optimal QAOA parameters. The parameters we derive are optimal in expectation, with the expectation taken over the distribution of the edge weights.

We first consider weights sampled from the exponential distribution and obtain optimal parameters for any graph size (Theorem 5.4.1). We analyze the exponential distribution separately as it allows us to derive globally-optimal parameters for finite-sized graphs. Then we consider the infinite-size limit, which enables us to relax the assumption on the distribution and obtain optimal parameters for graphs with weights sampled from an arbitrary distribution (Theorem 5.4.2). We then consider $p \geq 1$ in Section 5.4.2. We extend the techniques of [7] to relate the QAOA objective landscape for weighted MaxCut to that for unweighted MaxCut (Theorem 5.4.3) and the SK model (Corollary 5.4.4.1).

5.4.1 Globally-optimal parameters for QAOA with $p = 1$

According to [40, Theorem 7], the expected QAOA performance for MaxCut on triangle-free graphs can be expressed in closed form as:

$$\langle C(\gamma,\beta)\rangle = \frac{\sum_{\{u,v\}\in E} w_{uv}}{2} + \frac{\sin(4\beta)}{4} \sum_{\{u,v\}\in E} w_{uv}\sin(w_{uv}\gamma) \left(\prod_{k\in \mathrm{nbhd}(u)/\{v\}} \cos(w_{uk}\gamma) + \prod_{t\in \mathrm{nbhd}(v)/\{u\}} \cos(w_{tv}\gamma)\right), \tag{5.9}$$

where $\mathrm{nbhd}(u)$ is the neighborhood function that gives the set of vertices adjacent to u. The above is always maximized at $\beta = \frac{\pi}{8}$. Thus, with a slight abuse of notation, we define $\langle C(\gamma)\rangle = \langle C(\gamma, \frac{\pi}{8})\rangle$.

We are considering the expected QAOA energy over the edge weights, i.e. $\mathbb{E}_{\mathbf{w}}[\langle C(\gamma)\rangle]$. In the sections that follow $\mathbb{E}_{\mathbf{w}}[\,\cdot\,]$ denotes the expectation over the graph weights, w_{uv}, that are all drawn independently from the distribution $\mathbf{w}$. Thus, for $(D+1)$-regular graphs with i.i.d. edge weights, this expectation simplifies as follows

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}}[\langle C(\gamma)\rangle] \\
&= \frac{\sum_{\{u,v\}\in E} \mathbb{E}_{\mathbf{w}}[w_{uv}]}{2} \\
&+ \frac{1}{2} \sum_{\{u,v\}\in E} \mathbb{E}_{\mathbf{w}}\left[w_{uv}\sin(w_{uv}\gamma)\right] \left(\mathbb{E}_{\mathbf{w}}\left[\cos(w_{uv}\gamma)\right]\right)^D \\
&= \frac{N(D+1)}{2} \\
&\cdot \left[\frac{\mathbb{E}_{\mathbf{w}}[w]}{2} + \frac{1}{2}\mathbb{E}_{\mathbf{w}}\left[w\sin(w\gamma)\right] \left(\mathbb{E}_{\mathbf{w}}\left[\cos(w\gamma)\right]\right)^D\right],
\end{aligned} \tag{5.10}$$

where we drop the subscript on w since the edge weights are i.i.d. and use the fact that

$$|E| = \frac{N(D+1)}{2}. \tag{5.11}$$

We now consider edge weights distributed identically and independently according to the exponential distribution with parameter $\lambda > 0$, which has as its probability density

function $f(x) = \lambda e^{-\lambda x}$ if $x > 0$ or $f(x) = 0$ otherwise. The mean and standard deviation are $\mu = \sigma = \frac{1}{\lambda}$.

Theorem 5.4.1 ($p = 1$, exponential distribution, finite size)**.** *Let $\mathbb{E}_{exp(\lambda)}[\langle C(\gamma)\rangle]$ denote the expected QAOA objective with $p = 1$ over instances of the weighted MaxCut problem on a given triangle-free $(D+1)$-regular graph with edge weights, w, drawn i.i.d. from an exponential distribution with parameter λ. Then $\mathbb{E}_{exp(\lambda)}[\langle C(\gamma)\rangle]$ has a global maximum at $\gamma* = \frac{1}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}\sqrt{D+\frac{3}{2}}}$.*

Proof. To obtain the optimal parameters, we start with Equation (5.10) and use the following identities

$$\mathbb{E}_{\mathbf{w}}[w \sin(w\gamma)] = \frac{2\gamma\lambda^2}{(\lambda^2+\gamma^2)^2}, \tag{5.12}$$

$$\mathbb{E}_{\mathbf{w}}[\cos(w\gamma)] = \frac{\lambda^2}{\lambda^2+\gamma^2}, \tag{5.13}$$

which give

$$\begin{aligned}\mathbb{E}_{\exp(\lambda)}[\langle C(\gamma)\rangle] &= \frac{N(D+1)}{2} \\ &\quad \cdot \left[\frac{1}{2\lambda} + \frac{1}{2}\frac{2\gamma\lambda^2}{(\lambda^2+\gamma^2)^2}\left(\frac{\lambda^2}{\lambda^2+\gamma^2}\right)^D\right] \\ &= \frac{N(D+1)}{2}\left[\frac{1}{2\lambda} + \frac{\gamma\lambda^{2D+2}}{(\lambda^2+\gamma^2)^{D+2}}\right].\end{aligned} \tag{5.14}$$

Taking the derivative with respect to γ, we obtain

$$\frac{d}{d\gamma}\mathbb{E}_{\exp(\lambda)}[\langle C(\gamma)\rangle] = c\frac{\lambda^2 - (2D+3)\gamma^2}{(\lambda^2+\gamma^2)^{D+3}}, \tag{5.15}$$

where c is a positive and γ-independent constant. Setting the derivative to zero gives

$$\gamma = \frac{\pm\lambda}{\sqrt{2D+3}} = \frac{\pm 1}{\sqrt{\mathbb{E}_{\exp(\lambda)}[w^2]}\sqrt{D+\frac{3}{2}}}. \tag{5.16}$$

From Equation (5.14), we can see that

$$\mathbb{E}_{\exp(\lambda)}[\langle C(-\infty)\rangle] = \mathbb{E}_{\exp(\lambda)}[\langle C(\infty)\rangle] = \frac{N(D+1)}{4\lambda},$$

so the global maximum is

$$\gamma^* = \frac{1}{\sqrt{\mathbb{E}_{\exp(\lambda)}[w^2]}\sqrt{D+\frac{3}{2}}}. \tag{5.17}$$

□

Note that unlike the following Theorems, this proof does not rely on any assumptions on D.

We now consider a graph with edge weights drawn from an arbitrary distribution with mean value μ and standard deviation σ. To study the infinite-size limit, we define a quantity that tends to a constant as $D \to \infty$. Specifically, we consider the following quantity

$$\frac{\mathbb{E}_{\mathbf{w}}[\langle C(\gamma)\rangle]}{\mathbb{E}_{\mathbf{w}}\left[\sum_{\{u,v\}\in E} w_{uv}\right]}, \tag{5.18}$$

which reduces to the cut fraction if the graph is unweighted, i.e. $w_{uv} = 1;\ \forall\{u,v\} \in E$.

Using Equation (5.11) followed by Equation (5.10), we can write

$$\begin{aligned}
&\frac{\mathbb{E}_{\mathbf{w}}[\langle C(\gamma)\rangle]}{\mathbb{E}_{\mathbf{w}}\left[\sum_{\{u,v\}\in E} w_{uv}\right]} \\
&\quad= \frac{2}{N(D+1)\mu}\mathbb{E}_{\mathbf{w}}\left[\langle C(\gamma)\rangle\right] \\
&\quad= \frac{1}{2} + \frac{1}{2\mu}\mathbb{E}_{\mathbf{w}}\left[w\sin(w\gamma)\right]\left(\mathbb{E}_{\mathbf{w}}\left[\cos(w\gamma)\right]\right)^D \qquad (5.19)\\
&\quad= \frac{1}{2} + \frac{\vartheta_1(D,\gamma)}{\sqrt{D}}, \qquad (5.20)
\end{aligned}$$

where we introduce $\vartheta_1(D,\gamma)$ to match Π_* in Equation (5.5). We will now show that $\vartheta_1(D,\gamma)$ tends to a D-independent quantity as $D \to \infty$ when $\gamma = \Theta(D^{-1/2})$, and use the resulting limit to derive the optimal value γ^* in the limit of infinite-sized graphs.

The assumption of $\gamma = \Theta(D^{-1/2})$ is inspired by the numerical observation that the optimal γ for unweighted MaxCut is $\Theta(D^{-1/2})$ (see, e.g. [12, Figure 1b]). Furthermore, we prove that $\vartheta_1(D, \gamma)$

has a local maximum at a value $\gamma = \Theta(D^{-1/2})$ for sufficiently large D. In the limit of $D \to \infty$, we prove that this local maximum is also the global maximum. This motivates the definition of the following limiting quantity

$$\vartheta_1(\gamma) = \lim_{D\to\infty} \vartheta_1(D, \gamma/\sqrt{D}). \tag{5.21}$$

Theorem 5.4.2 ($p = 1$, infinite size)**.** *Consider weighted MaxCut on a given triangle-free $(D+1)$-regular graph with edge weights, w, drawn i.i.d. from a distribution* $\mathbf{w}$ *with finite second moment. Then for sufficiently large D, the function $\vartheta_1(D, \gamma)$ associated with QAOA for $p = 1$ has a local maximum at a γ that is $\Theta(D^{-1/2})$. Moreover, the limiting quantity $\vartheta_1(\gamma)$ attains its global maximum at $\gamma^* = \frac{1}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}$.*

Proof. The assumption of finite second moment along with Jensen's inequality implies that $\mathbb{E}_{\mathbf{w}}[|w|]$ is also finite. Thus, since the derivatives of the functions inside the expectations taken in Equation (5.19) are dominated in γ, i.e. $|w^2 \cos(w\gamma)| \leq w^2$ and $|w \sin(w\gamma)| \leq |w|$, the dominated convergence theorem and the mean-value theorem [41, Section 7.2.2] ensure that the operations of differentiation w.r.t. γ and expectation over w can be interchanged. This gives

$$\begin{aligned}\frac{d}{d\gamma}\vartheta_1(D, \gamma) &= \frac{\sqrt{D}}{2\mu}(\mathbb{E}_{\mathbf{w}}[\cos(w\gamma)])^{D-1} \\ &\cdot \left(\mathbb{E}_{\mathbf{w}}[w^2 \cos(w\gamma)]\mathbb{E}_{\mathbf{w}}[\cos(w\gamma)] - D(\mathbb{E}_{\mathbf{w}}[w \sin(w\gamma)])^2\right).\end{aligned} \tag{5.22}$$

Substituting $\gamma = \frac{\alpha}{\sqrt{D}}$, for α independent of D and using the Taylor series expansions of the trigonometric functions, we get

$$\begin{aligned}&\frac{d}{d\gamma}\vartheta_1(D, \gamma)\Big|_{\gamma=\alpha/\sqrt{D}} \\ &\quad = \frac{\sqrt{D}}{2\mu} \mathrm{e}^{-\frac{\mathbb{E}_{\mathbf{w}}[w^2]\alpha^2}{2}} [\mathbb{E}_{\mathbf{w}}[w^2] - (\mathbb{E}_{\mathbf{w}}[w^2])^2\alpha^2] + O(D^{-1/2}),\end{aligned} \tag{5.23}$$

where the implicit exchange of infinite series and expectation over w is justified by the finiteness of the second moment and Fubini's theorem [42, Theorem 8.8]. Here we use the observation that

$$\begin{aligned}
&\cos(x/\sqrt{D})^{D-1} \\
&= \left(1 - \frac{x^2}{2D}\right)^{D-1} + O(D^{-1}) \\
&= \mathrm{e}^{-\frac{x^2}{2}} + O(D^{-1}),
\end{aligned} \tag{5.24}$$

for x that is bounded by a constant independent of D. [‡]

For sufficiently large D, both

$$\frac{d}{d\gamma}\vartheta_1(D,\gamma)\Big|_{\gamma=0} = \frac{\sqrt{D}}{2\mu}\mathbb{E}_{\mathbf{w}}[w^2] + O(D^{-1/2}) > 0 \tag{5.25}$$

and

$$\frac{d}{d\gamma}\vartheta_1(D,\gamma)\Big|_{\gamma=\alpha^*/\sqrt{D}} \leq 0 \tag{5.26}$$

for some sufficiently large constant α^* independent of D. Thus, by Darboux's theorem [43, Theorem 5.12], γ has a local maxima in the interval $(0, \alpha^*/\sqrt{D})$ for each triangle-free $(D+1)$-regular graph. We now consider the limiting value of ϑ_1 in the regime of small γ. With $\gamma = \frac{\gamma'}{\sqrt{D}}$ for some D-independent γ', we get

$$\begin{aligned}
&\vartheta_1(D, \gamma'/\sqrt{D}) \\
&= \frac{\sqrt{D}}{2\mu}\mathbb{E}_{\mathbf{w}}[w\sin(w\gamma'/\sqrt{D})](\mathbb{E}_{\mathbf{w}}[\cos(w\gamma'/\sqrt{D})])^D \\
&= \frac{\mathbb{E}_{\mathbf{w}}[w^2]}{2\mu}\gamma'\mathrm{e}^{-\frac{\mathbb{E}_{\mathbf{w}}[w^2]\gamma'^2}{2}} + O(D^{-1}),
\end{aligned}$$

[‡]↑Note that the second equality follows from $(1-\frac{x^2}{2D})^{D-1} = (1-\frac{x^2}{2D})^D(1-O(D^{-1})) = \left[\sum_{k=0}^{D}\frac{(-x^2/2)^k}{k!}\right](1-O(D^{-1}))^2$ and works in the limit due to absolute convergence.

where we use Equation (5.24), and the implicit exchange of infinite series and expectation is justified by Fubini's theorem. Now taking the limit in D, we obtain

$$\begin{aligned}\vartheta_1(\gamma') &:= \lim_{D\to\infty} \vartheta_1(D, \gamma'/\sqrt{D}) \\ &= \frac{\mathbb{E}_{\mathbf{w}}[w^2]}{2\mu}\gamma' \mathrm{e}^{-\frac{\mathbb{E}_{\mathbf{w}}[w^2]\gamma'^2}{2}}.\end{aligned} \tag{5.27}$$

Now, consider the derivative,

$$\frac{d}{d\gamma'}\vartheta_1(\gamma') = \frac{\mathbb{E}_{\mathbf{w}}[w^2]\mathrm{e}^{-\frac{\mathbb{E}_{\mathbf{w}}[w^2]\gamma'^2}{2}}}{2\mu}(1 - \mathbb{E}_{\mathbf{w}}[w^2]\gamma'^2). \tag{5.28}$$

It can be easily seen that the function $\vartheta_1(\gamma')$ is always decreasing to the right of the local maximum at $\gamma^* = \frac{1}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}$, and the function is negative to the left of zero. Thus this is in fact a global optima. □

Remark 1. *To see the correspondence between Theorem 5.4.2 and Theorem 5.4.1, i.e. that the latter is a special case of the former, rescale $\gamma \to \gamma/\sqrt{D}$ and note that the constant in the denominator in Theorem 5.4.1 has no effect on the limiting value as $D \to \infty$.*

5.4.2 Correspondence between QAOA on weighted and unweighted graphs with $p \geq 1$

To derive a parameter scaling rule for arbitrary p, we extend the techniques developed in Ref. [7] for MaxCut on large-girth, regular unweighted graphs to large-girth, regular graphs with i.i.d. edge weights. Without a subscript, $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ will refer to the p-layer QAOA state for a random weighted instance of MaxCut on a given $(D+1)$-regular graph with weights, w, drawn from a distribution $\mathbf{w}$.

Note that

$$\langle C(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \frac{1}{2}\sum_{\{u,v\}\in E} w_{uv}(1 - \langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\mathrm{Z}_u\mathrm{Z}_v|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle). \tag{5.29}$$

We start by proving the following result about the above quantity that is valid for any p.

Lemma 1. *If a* $(D+1)$ *regular graph has girth* $> 2p+1$ *and i.i.d. edge weights,* w*, drawn from* $\mathbf{w}$*, then the QAOA objective for weighted MaxCut on this graph satisfies*

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}[\langle C(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle] &= \frac{N(D+1)\mu}{4} \\ &- \frac{N(D+1)}{4}\mathbb{E}_{\mathbf{w}}[w\langle\boldsymbol{\gamma},\boldsymbol{\beta}|Z_L Z_R|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle],\end{aligned} \tag{5.30}$$

for any edge $\{L,R\}$*.*

Proof. The locality of the p-layer QAOA combined with the regularity and girth $> 2p+1$ assumptions implies that the quantity $w_{LR}\langle\boldsymbol{\gamma},\boldsymbol{\beta}|\mathrm{Z}_L\mathrm{Z}_R|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle$ can only depend on vertices that lie in a D-ary tree of diameter $2p+1$. This is the QAOA "light cone" for the term $\langle\boldsymbol{\gamma},\boldsymbol{\beta}|\mathrm{Z}_L\mathrm{Z}_R|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle$. The tree subgraph, without weights, for $p=2$ and $D+1=4$ is shown in Figure 5.1. Thus, it follows from [7, Equations A.2-A.5] that

$$\begin{aligned}w_{LR}\langle\boldsymbol{\gamma},\boldsymbol{\beta}|\mathrm{Z}_L\mathrm{Z}_R|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle &= \sum_{\{\boldsymbol{z}_u\}} w_{LR} z_L^{[0]} z_R^{[0]} \\ &\cdot \left[\exp\left(-\mathrm{i}\sum_{\{u,v\}\in E_{LR}} w_{uv}\boldsymbol{\Gamma}\cdot(\boldsymbol{z}_u\boldsymbol{z}_v)\right)\prod_{v\in V_{LR}} g(\boldsymbol{z}_v)\right],\end{aligned} \tag{5.31}$$

where (V_{LR}, E_{LR}) denotes the vertex and edge sets corresponding to the tree subgraph seen from the edge $\{L,R\}$. Note that Ref. [7] has an extra factor of $\frac{1}{\sqrt{D}}$ in the exponential since the cost function in Equation (5.29) is therein defined with that extra factor. In addition, $\boldsymbol{\Gamma}$ is a $(2p+1)$-component vector with entries $\boldsymbol{\Gamma}_r = \gamma_r$, $\boldsymbol{\Gamma}_{-r} = -\gamma_r$, $\boldsymbol{\Gamma}_0 = 0$ and $1 \le r \le p$. Also for node u, the vector $\boldsymbol{z}_u = (z_u^{[1]},\ldots,z_u^{[r]},z_u^{[0]},z_u^{[-r]},\ldots,z_u^{[-1]}) \in \{-1,1\}^{2p+1}$. The notation $\{\boldsymbol{z}_u\}$ is short for the collection $\{\boldsymbol{z}_u | u \in V_{LR}, \boldsymbol{z}_u \in \{-1,1\}^{2p+1}\}$. Lastly $(\boldsymbol{z}_u\boldsymbol{z}_v)$ denotes an element-wise product, and

$$\begin{aligned}g(\boldsymbol{z}_u) &= \left[\langle z_u^{[p]}|\mathrm{e}^{\mathrm{i}\beta_p \mathrm{X}}|z_u^{[0]}\rangle\langle z_u^{[0]}|\mathrm{e}^{-\mathrm{i}\beta_p \mathrm{X}}|z_u^{[-p]}\rangle\right] \\ &\cdot \frac{1}{2}\prod_{r=1}^{p-1}\langle z_u^{[-(r+1)]}|\mathrm{e}^{-\mathrm{i}\beta_r \mathrm{X}}|z_u^{[-r]}\rangle\langle z_u^{[r]}|\mathrm{e}^{\mathrm{i}\beta_r \mathrm{X}}|z_u^{[r+1]}\rangle.\end{aligned} \tag{5.32}$$

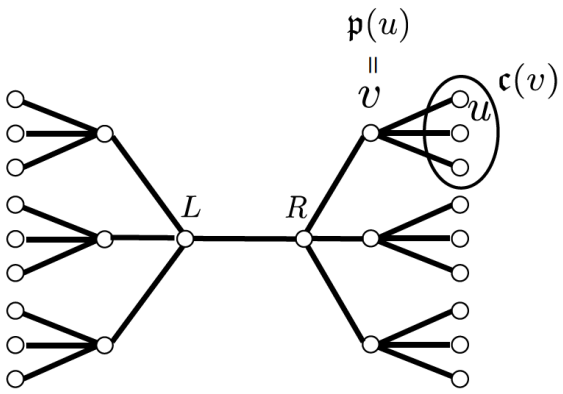

Figure 5.1. Unweighted tree subgraph seen by QAOA from the edge $\{L, R\}$ with $p = 2$ on a four-regular graph. The operation $\mathfrak{p}$ produces the parent of a node, and the operation $\mathfrak{c}$ produces the set of a node's immediate children.

The authors of [7] noted that Equation (5.31) can be computed recursively by traversing, from leaves to roots, the left and right branches simultaneously. This effectively "factors" the right-hand side of Equation (5.31). For simplicity we will only do this for $p = 2$, however, the generalization is straightforward. Define $H_D^{(0)} = 1$. We start by summing over the configuration of an arbitrary leaf in either branch of Figure 5.1:

$$\sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u) \exp\left(-\mathrm{i} w_{u\mathfrak{p}(u)} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_{\mathfrak{p}(u)})\right), \tag{5.33}$$

where $\mathfrak{p}(u)$ is the parent of u in the tree subgraph. Since the left and right branches are D-ary trees, the expression for any node v in the second level of either branch is

$$\sum_{\boldsymbol{z}_v} g(\boldsymbol{z}_v) H_D^{(1)}(\boldsymbol{z}_v) \exp\left(-\mathrm{i} w_{v\mathfrak{p}(v)} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_v \boldsymbol{z}_{\mathfrak{p}(v)})\right), \tag{5.34}$$

where

$$\begin{aligned} H_D^{(1)}(\boldsymbol{z}_v) &:= \prod_{u \in \mathfrak{c}(v)} \sum_{\boldsymbol{z}_u} H_D^{(0)}(\boldsymbol{z}_u) g(\boldsymbol{z}_u) \exp\left(-\mathrm{i} w_{uv} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v)\right) \\ &= \prod_{u \in \mathfrak{c}(v)} \sum_{\boldsymbol{z}_u} H_D^{(0)} g(\boldsymbol{z}_u) \cos\left(w_{uv} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v)\right). \end{aligned} \tag{5.35}$$

The second equality follows from the even parity of $g \cdot H_D^{(r)}$, implied by [7, Claims A.14 and A.15] and $\mathfrak{c}(v)$ denotes the set of D immediate children of v.

This can again be done for the next level, i.e. the roots L or R, producing the quantity $H_D^{(2)}$. Lastly, we combine the results from the two branches by summing over the configurations of the left and right roots:

$$\begin{aligned} w_{LR}\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \mathrm{Z}_u \mathrm{Z}_v | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle &= \sum_{\boldsymbol{z}_L, \boldsymbol{z}_R} w_{LR} z_L^{[0]} z_R^{[0]} \\ &\cdot \left[g(\boldsymbol{z}_L) g(\boldsymbol{z}_R) H_D^{(2)}(\boldsymbol{z}_L) H_D^{(2)}(\boldsymbol{z}_R) \exp\left(-\mathrm{i} w_{LR} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)\right) \right] \\ &= -\mathrm{i} \sum_{\boldsymbol{z}_L, \boldsymbol{z}_R} z_L^{[0]} z_R^{[0]} g(\boldsymbol{z}_L) g(\boldsymbol{z}_R) \\ &\cdot \left[H_D^{(2)}(\boldsymbol{z}_L) H_D^{(2)}(\boldsymbol{z}_R) w_{LR} \sin\left(w_{LR} \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)\right) \right], \end{aligned} \tag{5.36}$$

where the last equality follows again from the even parity of $g \cdot H_D^{(r)}$. The general iteration for the random quantity $w_{LR}\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \mathrm{Z}_L \mathrm{Z}_R | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle$ follows by induction.

Note that in the previous recursion, each edge was only counted once. Since all of the edges are i.i.d., the expectation operation commutes with all products that appear in the right-hand side of Equation (5.36). More specifically, we have the following for general p:

$$\begin{aligned} \mathbb{E}_{\mathbf{w}}[w_{LR}\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \mathrm{Z}_L \mathrm{Z}_R | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle] &= -\mathrm{i} \sum_{\boldsymbol{z}_L, \boldsymbol{z}_R} z_L^{[0]} z_R^{[0]} g(\boldsymbol{z}_L) g(\boldsymbol{z}_R) \\ &\cdot \left[\bar{H}_D^{(p)}(\boldsymbol{z}_L) \bar{H}_D^{(p)}(\boldsymbol{z}_R) \mathbb{E}_{\mathbf{w}}[w \sin\left(w \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)\right)] \right], \end{aligned} \tag{5.37}$$

where subscripts have been dropped from the weights due to the i.i.d. assumption, and

$$\begin{aligned} \bar{H}_D^{(r)}(\boldsymbol{z}_v) &:= \mathbb{E}_{\mathbf{w}}[H_D^{(r)}(\boldsymbol{z}_v)] \\ &= \prod_{u \in \mathfrak{c}(v)} \sum_{\boldsymbol{z}_u} \bar{H}_D^{(r-1)}(\boldsymbol{z}_u) g(\boldsymbol{z}_u) \mathbb{E}_{\mathbf{w}}[\cos\left(w \boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v)\right)] \end{aligned} \tag{5.38}$$

for $1 \leq r \leq p$.

Since all randomness has been removed by the expectation operation, it is evident that this quantity only depends on the graph structure and not the sampled weights, since they

are i.i.d. The non-random quantities in the iteration for $w_{LR}\langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\mathrm{z}_L\mathrm{z}_R|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ only depend on the local graph structure that QAOA sees from a given edge. As argued earlier this graph structure is always two D-ary trees joined at their roots. Thus, the quantity $\mathbb{E}_{\mathbf{w}}[w\langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\mathrm{z}_L\mathrm{z}_R|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle]$ is independent of the chosen edge $\{L, R\}$, which is analogous to the unweighted case.

The result of the lemma follows from Equation (5.29) and the linearity of expectation. □

By the previous lemma, we can, analogously to the $p = 1$ case, define $\vartheta_p(D, \boldsymbol{\gamma}, \boldsymbol{\beta})$ as follows:

$$\begin{aligned}\frac{\mathbb{E}_{\mathbf{w}}[\langle C(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle]}{\mathbb{E}_{\mathbf{w}}\left[\sum_{\{u,v\}\in E} w_{uv}\right]} &= \frac{\mathbb{E}_{\mathbf{w}}[\langle C(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle]}{\frac{N(D+1)\mu}{2}} \\ &= \frac{1}{2} - \frac{1}{2\mu}\mathbb{E}_{\mathbf{w}}[w\langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\mathrm{zz}|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle] \\ &= \frac{1}{2} + \frac{\vartheta_p(D, \boldsymbol{\gamma}, \boldsymbol{\beta})}{\sqrt{D}},\end{aligned} \tag{5.39}$$

and

$$\begin{aligned}\vartheta_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) &:= \lim_{D\to\infty} \vartheta_p(D, \boldsymbol{\gamma}/\sqrt{D}, \boldsymbol{\beta}) \\ &= \lim_{D\to\infty} -\frac{\sqrt{D}}{2\mu}\mathbb{E}_{\mathbf{w}}\left[w\left\langle \frac{\boldsymbol{\gamma}}{\sqrt{D}}, \boldsymbol{\beta}\right|\mathrm{zz}\left|\frac{\boldsymbol{\gamma}}{\sqrt{D}}, \boldsymbol{\beta}\right\rangle\right],\end{aligned} \tag{5.40}$$

where the subscripts on w and z have been dropped since they can be arbitrary by the previous lemma.

The quantity considered in [7, Equation A.19] for the unweighted case is the following

$$\nu_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lim_{D\to\infty} -\frac{\sqrt{D}}{2}\left\langle \frac{\boldsymbol{\gamma}}{\sqrt{D}}, \boldsymbol{\beta}\right|\mathrm{zz}\left|\frac{\boldsymbol{\gamma}}{\sqrt{D}}, \boldsymbol{\beta}\right\rangle_{\mathrm{u}}, \tag{5.41}$$

where the subscript "u" indicates that the parameterized state is prepared by a p-layer QAOA for the corresponding unweighted problem on the same graph. Our main result below shows that $\boldsymbol{\gamma}$ can be scaled to make these two quantities equal up to a global scaling factor.

Theorem 5.4.3 ($p \geq 1$, infinite size)**.** *If the girth* $> 2p+1$*, and the edge-weight distribution,* $\mathbf{w}$*, has finite second moment, then for all parameters* $\boldsymbol{\gamma}, \boldsymbol{\beta}$ *the following holds*

$$\nu_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{\mu}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}} \vartheta_p \left(\frac{\boldsymbol{\gamma}}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}, \boldsymbol{\beta} \right). \tag{5.42}$$

Proof. We implicitly assume that $\boldsymbol{\gamma} \mapsto \frac{\boldsymbol{\gamma}}{\sqrt{D}}$ and thus $\boldsymbol{\Gamma} \mapsto \frac{\boldsymbol{\Gamma}}{\sqrt{D}}$ in Equation (5.37). By the product rule for limits, we can evaluate the limits of the terms $\bar{H}_D^{(p)}(\boldsymbol{z}_L)$, $\bar{H}_D^{(p)}(\boldsymbol{z}_R)$, and the one involving the sin separately, since we will show they individually exist.

Note that for any sum inside of the product of Equation (5.38):

$$\begin{aligned}
&\sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u) \bar{H}_D^{(p-1)}(\boldsymbol{z}_u) \mathbb{E}_{\mathbf{w}} \left[\cos\left(w \frac{\boldsymbol{\Gamma}}{\sqrt{D}} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v) \right) \right] \\
&= \left(1 - \frac{\mathbb{E}_{\mathbf{w}}[w^2]}{2D} \sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u) \bar{H}_D^{(p-1)}(\boldsymbol{z}_u) (\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v))^2 \right. \\
&\left. + O(D^{-2}) \right),
\end{aligned} \tag{5.43}$$

where the implicit exchange of the expectation operator and infinite series expansion of trig functions is justified by Fubini's theorem and the assumption of the weight distribution having finite second moment, like in Section 5.4.1. In addition, we have used the following generalization of [7, Equation (A.23)], where for any r:

$$\sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u) H_D^{(r-1)}(\boldsymbol{z}_u) = 1. \tag{5.44}$$

After taking expectations, it follows that $\sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u)\bar{H}_D^{(r-1)}(\boldsymbol{z}_u) = 1$. By the i.i.d. assumption, Equation (5.43) is the same for every $u \in \mathfrak{c}(v)$, and thus

$$\begin{aligned}
&\bar{H}_D^{(p)}(\boldsymbol{z}_v) \\
&= \left(1 - \frac{\mathbb{E}_{\mathbf{w}}[w^2]}{2D} \sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u)\bar{H}_D^{(p-1)}(\boldsymbol{z}_u)(\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v))^2\right)^D \\
&+ O(D^{-1}) \\
&= \exp\left(-\frac{\mathbb{E}[w^2]}{2} \sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u)\bar{H}_D^{(p-1)}(\boldsymbol{z}_u)(\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v))^2\right) \\
&+ O(D^{-1}),
\end{aligned} \tag{5.45}$$

where we use Equation (5.24).

Along with continuity, the previous result implies

$$\begin{aligned}
\bar{H}^{(p)}(\boldsymbol{z}_v) &:= \lim_{D\to\infty} \bar{H}_D^{(p)}(\boldsymbol{z}_v) \\
&= \exp\left(-\frac{\mathbb{E}_{\mathbf{w}}[w^2]}{2} \sum_{\boldsymbol{z}_u} g(\boldsymbol{z}_u)\bar{H}^{(p-1)}(\boldsymbol{z}_u)(\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_u \boldsymbol{z}_v))^2\right).
\end{aligned} \tag{5.46}$$

The limit can then be propagated down to the lowest level of the recursion. Similarly for the term involving:

$$\begin{aligned}
&\lim_{D\to\infty} \sqrt{D}\mathbb{E}_{\mathbf{w}}\left[w \sin\left(w\frac{\boldsymbol{\Gamma}}{\sqrt{D}} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)\right)\right] \\
&= \mathbb{E}_{\mathbf{w}}[w^2]\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)
\end{aligned} \tag{5.47}$$

Putting this altogether, for arbitrary $\{L, R\} \in E$, we have

$$\begin{aligned}
\vartheta_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) &= \frac{\mathrm{i}\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}{2\mu} \sum_{\boldsymbol{z}_L, \boldsymbol{z}_R} z_L^{[0]} z_R^{[0]} g(\boldsymbol{z}_L) g(\boldsymbol{z}_R) \\
&\cdot \left[\bar{H}^{(p)}(\boldsymbol{z}_L)\bar{H}^{(p)}(\boldsymbol{z}_R)\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}\boldsymbol{\Gamma} \cdot (\boldsymbol{z}_L \boldsymbol{z}_R)\right]
\end{aligned} \tag{5.48}$$

and comparing with the unweighted case [7, Equation A.26] reveals that the equality in Equation (5.42) holds. □

Remark 2. *Note that in the i.i.d. case the square of the denominator of Equation (5.6) is an unbiased estimator of* $\mathbb{E}_{\mathbf{w}}[w^2]$*. The i.i.d. case is the inspiration for the more general scaling rule.*

Remark 3. *By [7, Section 6], which extends the iteration presented earlier for unweighted regular graphs to unweighted regular k-uniform hypergraphs, Theorem 5.4.3 can be trivially extended to weighted regular k-uniform hypergraphs.*

This result implies a relationship in the infinite-size limit between QAOA's objective value for weighted MaxCut and the SK model. Let

$$C^{\mathrm{SK}} = \sum_{1 \leq u < v \leq N} \frac{J_{uv}}{\sqrt{N}} \mathrm{Z}_u \mathrm{Z}_v, \tag{5.49}$$

where $J_{uv} \sim \mathcal{N}(0, 1)$, and

$$V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) := \lim_{n \to \infty} \mathbb{E}_J[{}_J\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C^{\mathrm{SK}}/N | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle_J], \tag{5.50}$$

where the subscript J of the state signifies that the state was prepared by a p-layer QAOA with the SK objective as the phase operator.

Theorem 5.4.4 (Restated from [7])**.** *For all* p *and all parameters* $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ *the following holds*

$$V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \nu_p(\boldsymbol{\gamma}, \boldsymbol{\beta}). \tag{5.51}$$

Trivially, this in combination with Theorem 5.4.3 leads to the following corollary.

Corollary 5.4.4.1. *If the edge-weight distribution has finite second moment, then for all* p *and all parameters* $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ *the following holds*

$$V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{\mu}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}} \vartheta_p \left(\frac{\boldsymbol{\gamma}}{\sqrt{\mathbb{E}_{\mathbf{w}}[w^2]}}, \boldsymbol{\beta} \right). \tag{5.52}$$

Thus the performance of QAOA on SK, MaxCut on large-girth, regular graphs and weighted MaxCut on large-girth, regular graphs are equivalent in the infinite limit.

Remark 4. *By [44, Theorem 3] and Remark 3, one can trivially extend Corollary 5.4.4.1 to connect QAOA's performance on weighted MaxCut on regular k-uniform hypergraphs to its performance on pure k-spin models, generalizing SK.*

5.5 Observations about biased SK model

As presented in Corollary 5.4.4.1, there is a deep connection between arbitrarily-weighted MaxCut and the SK model. The SK model is given in Equation (5.49) and has couplings $J_{uv} \sim \mathcal{N}(0,1)$. A natural generalization to consider is a model which has couplings $J_{uv} \sim \mathcal{N}(\mu, \sigma^2)$ with arbitrary μ and σ. More generally, we can allow for the bias to be a function of the number of spins, i.e. $\mu(N)$. When $\mu(N) \neq 0$, we call this the biased SK model, and when $\mu(N) = 0$, we call it the standard SK model. Unfortunately, this natural generalization does not lead to interesting behavior. Specifically, we show that unless $\mu(N) \to 0$, the biased SK problem is trivial in the thermodynamic limit.

The performance of QAOA for arbitrary p on standard SK, specifically an iteration for the quantity V_p, was originally established in [13] using different techniques than those of Section 5.4.2. However, it is not clear how these techniques can be generalized to non-symmetric distributions. In this section, we use a different set of elementary techniques to determine the limiting optimal value of different versions of the biased SK model. Our goal is to find an analog to the Parisi value for the biased model. The following is based on [45].

For $\boldsymbol{z} \in \{-1, 1\}^N$, let

$$G(\boldsymbol{z}) = -\sum_{1 \leq \mathrm{i} < \mathrm{j} \leq N} J_{\mathrm{ij}}^{(N)} z_{\mathrm{i}} z_{\mathrm{j}}, \tag{5.53}$$

where $J_{\mathrm{ij}}^{(N)} \sim \mathcal{N}(\mu(N), \sigma^2)$. The optimization problem is $\max_{\boldsymbol{z}} G(\boldsymbol{z})$. Note that unlike the standard SK, which is symmetric around zero, here we must keep track of the signs of the couplings.

When $\mu(N) = 0$, we know that $\mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})] = O(N^{3/2})$. In the standard SK model, the weights are scaled by $N^{-1/2}$ to ensure that expected maximum instead grows linearly with N. This is the reasoning for the scaling of the standard SK objective presented in Equation (5.49). For even N, let

$$h(N) = \begin{cases} \frac{N^2}{4}, & \text{if } \mu(N) > 0, \\ -\binom{N}{2}, & \text{if } \mu(N) < 0, \\ 0, & \text{if } \mu(N) = 0, \end{cases}$$

which equals $\frac{\max_{\boldsymbol{z}} \mathbb{E}_J[G(\boldsymbol{z})]}{\mu(N)}$ when $\mu(N) \neq 0$. More specifically, when $\mu(N) > 0$, the problem $\max_{\boldsymbol{z}} \mathbb{E}_J[G(\boldsymbol{z})]$ reduces to MaxCut on a complete graph with all edge weights equal to $\mu(N)$, and thus the optimal cut value is $\mu(N)\frac{N^2}{4}$, i.e. set half of the $z_{\mathrm{i}} = 1$. When $\mu(N) < 0$, the optimal value is obtained when all $z_{\mathrm{i}} = 1$, and results in an objective function value of $\mu(N)\binom{N}{2}$. Note that when N is odd, the factor in the denominator is the same for all cases, and thus we can restrict to even N, wlog.

We have the following by the convexity of max:

$$\mu(N)h(N) = \max_{\boldsymbol{z}} \mathbb{E}_J[G(\boldsymbol{z})] \leq \mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})] \tag{5.54}$$

and

$$\begin{aligned} \mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})] &= \frac{1}{\alpha} \log \exp \mathbb{E}_J[\alpha \max_{\boldsymbol{z}} G(\boldsymbol{z})] \\ &\leq \frac{1}{\alpha} \log \mathbb{E}_J[\exp(\alpha \max_{\boldsymbol{z}} G(\boldsymbol{z}))] \\ &\leq \frac{1}{\alpha} \log \sum_{\boldsymbol{z}} \mathbb{E}_J[\exp(G(\boldsymbol{z}))] \\ &= \frac{1}{\alpha} \log \sum_{\boldsymbol{z}} \exp\left(\frac{\alpha^2\sigma^2}{2}\binom{N}{2} - \alpha\mu(N)\sum_{\mathrm{i}<\mathrm{j}} z_{\mathrm{i}} z_{\mathrm{j}}\right) \\ &\leq \frac{1}{\alpha} \log 2^N \exp\left(\frac{\alpha^2\sigma^2}{2}\binom{N}{2} + \alpha\mu(N)h(N)\right) \\ &= \frac{N\log(2)}{\alpha} + \frac{\alpha\sigma^2}{2}\binom{N}{2} + \mu(N)h(N). \end{aligned} \tag{5.55}$$

Minimizing over $\alpha > 0$, gives

$$\alpha^2 = \frac{N \log(4)}{\sigma^2 \binom{N}{2}}, \tag{5.56}$$

and thus

$$\begin{aligned} \mu(N)h(N) &\leq \mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})] \\ &\leq \sigma \sqrt{\log(4) N \binom{N}{2}} + \mu(N)h(N). \end{aligned} \tag{5.57}$$

Note, the right-hand side of the last inequality now involves only σ instead of σ^2, and thus is invariant under any scaling before or after the expectation in $\mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})]$.

When $\mu_N = 0$ and $h(N) = 0$, we recover the scaling mentioned for SK, i.e. $O(N^{3/2})$. Thus,

$$\lim_{N\to\infty} \frac{\mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})]}{N^{3/2}} = \sigma \Pi^*, \tag{5.58}$$

which is a multiple of the Parisi value Π^*.

While the standard SK model typically has $\sigma = 1$, we note that, when $\mu(N) = 0$, the limiting behavior for SK with a distribution with any $\sigma \neq 1$ can be obtained by simple rescaling. Specifically for $\mathcal{N}(0, \sigma^2)$, the quantity V_p in Equation (5.50) scales as $V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = V_{p,1}(\boldsymbol{\gamma}, \boldsymbol{\beta}) = V_{p,\sigma}(\frac{\gamma}{\sigma}, \boldsymbol{\beta})$. Therefore for the remainder of this Section we focus on the case where $\mu(N) \neq 0$.

If $\mu(N) \to \mu$ as $N \to \infty$ for some nonzero constant μ, the term involving $\mu(N)$ dominates, and the expected maximum is $\Theta(N^2)$. Thus, the limiting quantity is

$$\lim_{N\to\infty} \frac{\mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})]}{N^2} = \frac{\mu}{2(1 + \text{sign}(\mu))}, \tag{5.59}$$

where we define sign(x) to be 1 when $x \geq 0$ and 0 when $x < 0$. The solution to the above problem is trivial in the infinite limit: set all $z_{\mathrm{i}} = 1$, when $\mu < 0$, and set half of the z_{i} to 1 when $\mu > 0$.

However, to compensate for the maximum growing $\Theta(\sqrt{N})$ faster when $\mu(N)$ is not always zero, we could consider defining the biased SK model to have $\mu(N) = O(N^{-1/2})$. Then letting $\mu = \lim_{N\to\infty} \mu(N)\sqrt{N}$ we get

$$\lim_{N\to\infty} \frac{\mathbb{E}_J[\max_{\boldsymbol{z}} G(\boldsymbol{z})]}{N^{3/2}} = \sigma\Pi^* + \frac{\mu}{2(1+\mathrm{sign}(\mu))}. \tag{5.60}$$

In this regime, the "biased SK" model appears non-trivial and is distinct from the standard SK model.

5.6 Numerical results

Numerical investigation of the proposed parameter setting rule has been performed on a dataset of weighted graphs from Ref. [9], available through QAOAKit [46]. The dataset consists of a total of 34,701 weighted graphs with up to 20 nodes and contains both regular and non-regular graphs. The graphs have edge weights drawn i.i.d. from four different distributions, namely Uniform over $[0, 1]$ ("Uniform+"), Uniform over $[-1, 1]$ ("Uniform $\pm$"), Exponential (with $\lambda = 0.2$), and Cauchy.

For the numerical study, we investigate two proposed parameter setting rules, which are variants of Equation (5.6):

$$\text{(i)} \quad \boldsymbol{\gamma}* = \frac{\boldsymbol{\gamma}^{\mathrm{inf}}}{\sqrt{\frac{D-1}{|E|}\sum_{\{u,v\}\in E} w_{uv}^2}}, \tag{5.61}$$

$$\text{(ii)} \quad \boldsymbol{\gamma}* = \frac{\boldsymbol{\gamma}^{\mathrm{inf}}}{\sqrt{\frac{1}{|E|}\sum_{\{u,v\}\in E} w_{uv}^2}} \arctan\left(\frac{1}{\sqrt{D-1}}\right), \tag{5.62}$$

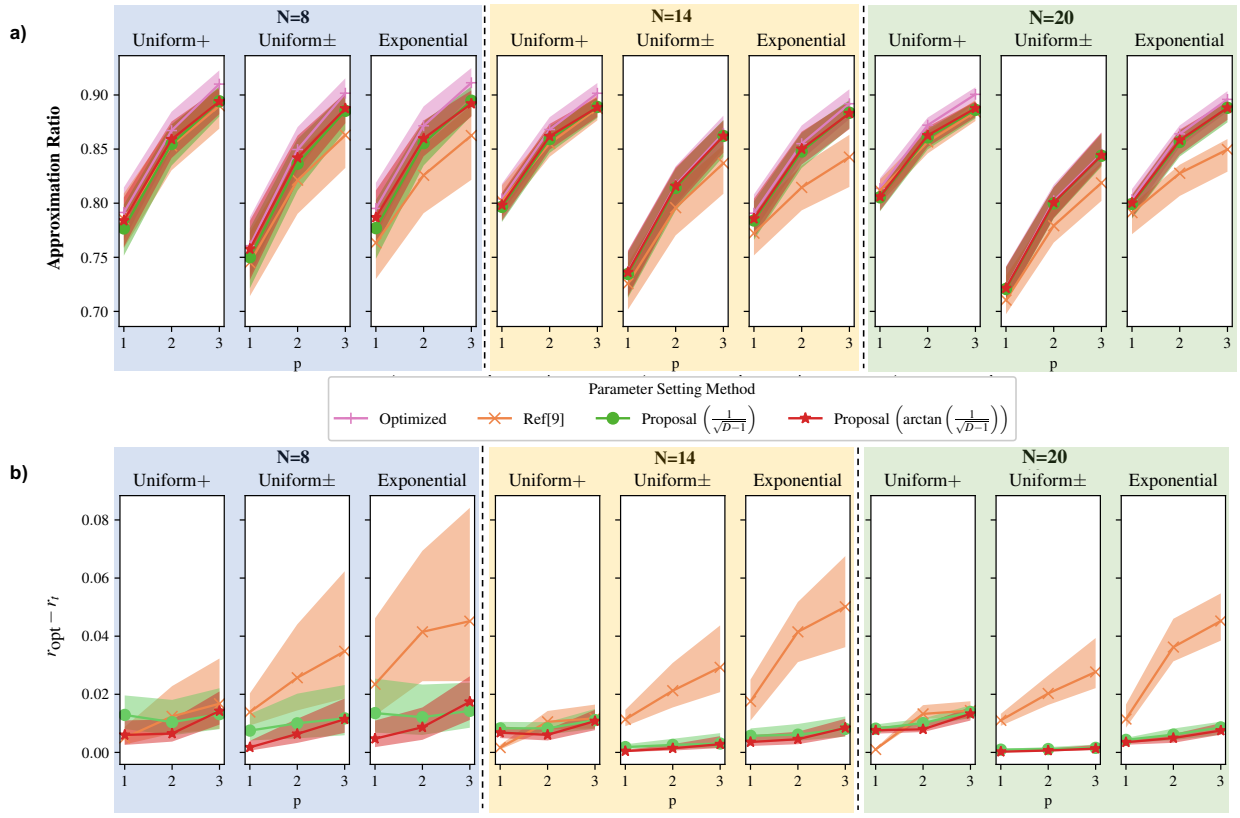

Figure 5.2. a) The approximation ratios obtained with directly optimized parameters, parameter setting method of Ref. [9], and parameter setting methods presented in this work. b) The gap between the approximation ratios with optimized parameters and with parameter setting methods of Ref. [9], (i) and (ii). The proposed parameter setting methods perform better when compared to the prior work, as indicated by the reduced gap from the objective obtained with the optimized parameters.

where the parameters $\boldsymbol{\beta}^{\text{inf}}, \boldsymbol{\gamma}^{\text{inf}}$ are the optimized parameters for large-girth, regular graphs in the infinite-size limit from [47, Table 4], and D is the average degree. Our baseline is the parameter scheme of Ref. [9], given by:

$$\boldsymbol{\gamma}* = \frac{\boldsymbol{\gamma}^{\text{median}}}{\frac{1}{|E|}\sum_{\{u,v\}\in E}|w_{uv}|}\arctan\left(\frac{1}{\sqrt{D-1}}\right), \tag{5.63}$$

where $\boldsymbol{\gamma}^{\text{median}}$ is a median taken over optimized parameters for all 261,080 nonisomorphic connected 9-node graphs. The key difference between our scaling and that of Ref. [9] is the choice of the denominator. Since $\boldsymbol{\gamma}^{\text{median}}$ is close in value to $\boldsymbol{\gamma}^{\text{inf}}$, the nominator is similar in both schemes.

We refer to the parameter setting procedure described in Equation 5.61 as method (i) and that in Equation 5.62 as method (ii). The first method is inspired directly by the analytical results described in Section 5.4.

We observe that in the case of small p, better results are obtained when the formula for $p = 1$ from [37] is considered, which motivates the second rule. We note that while we have derived the exact formula for graphs with weights sampled from the exponential distribution, we do not use it in the numerical experiments. Our goal for numerics is to simulate the practical setting, wherein one does not know the distribution from which the weights are sampled.

We analyze the performance of the proposed parameter setting rules across multiple weight distributions, values of p and values of N. Herein, we denote the median approximation ratio with directly optimized parameters by r_{opt}, with the parameter setting scheme from Ref. [9] by $r_{[9]}$ and with the two proposed methods as r_D and $r_{\arctan}$ respectively. We refer to the difference between the approximation ratio of a given parameter setting scheme and r_{opt} as the optimality gap. The results are presented in Figure 5.2.

Our techniques lead to lower optimality gaps as compared to Ref. [9] in all cases except $p = 1$ with weights sampled uniformly from $[0, 1]$. We note that the gap between the methods (i) and (ii) reduces as p increases. For example, for $N = 8$ the optimality gap drops from 0.0111 on average for $p = 1$, to 0.0062 for $p = 2$, and eventually to 0.0005 for $p = 3$.

The median difference in approximation ratios for all considered p and weight distributions is 1.8 p.p. for method (i) and 1.45 p.p. for method (ii). Specifically, for the cases of exponential and Cauchy distributions, the median differences in approximation ratios from our method (i) are 1.3 p.p. and 3.8 p.p. respectively, and those from method (ii) remain a mere 1.0 p.p. and 3.3 p.p. respectively. For comparison, the previous proposal [9] obtains median differences of 3.6 p.p. and 20.7 p.p. for the weights drawn from exponential and Cauchy distributions respectively. As can be seen in Figure 5.3, for the case when the edge weights are drawn from a Cauchy distribution, the improvement over Ref. [9] is the largest, with an $8\times$ reduction in optimality gap at $p = 3$.

Table 5.1. The optimality gaps achieved by the proposed parameter setting rules and the rule in [9]. Proposed method leads to lower optimality gaps in all cases except $p = 1$ for the Uniform+ distribution.

Weight Distribution	p	$r_{opt} - r_{[9]}$	$r_{opt} - r_D$	$r_{opt} - r_{\arctan}$
Uniform+	1	0.5	1.3	0.6
	2	1.2	1.0	0.6
	3	1.6	1.3	1.4
Uniform±	1	1.4	0.7	0.2
	2	2.6	1.1	0.6
	3	3.5	1.3	1.1
Exponential	1	2.3	1.3	0.5
	2	4.1	1.2	0.8
	3	4.5	1.4	1.7
Cauchy	1	17.5	4.8	3.3
	2	20.8	4.1	3.9
	3	24.0	3.2	3.0

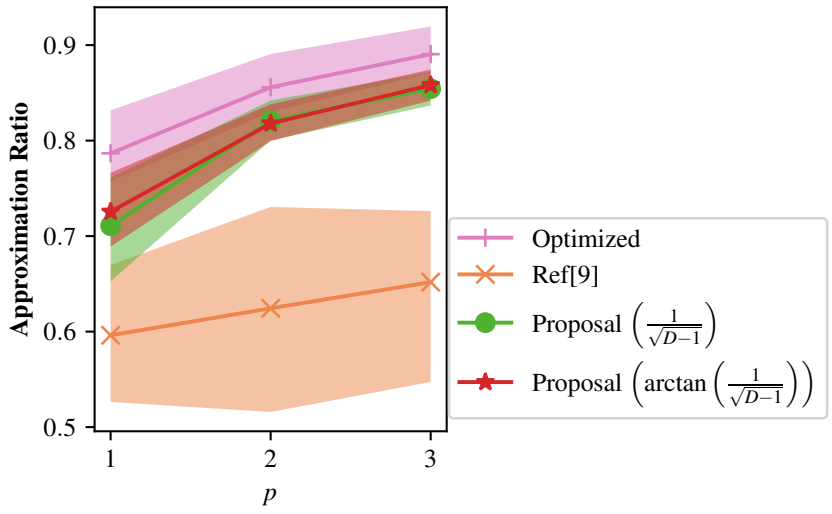

Figure 5.3. Approximation ratio for the graphs with edge weights drawn from a Cauchy distribution for $N = 14$. The proximity to the optimized parameter scenario, especially for large p, indicates the power of the suggested parameter setting strategies and shows a clear improvement over the earlier work. Our methods reduce the optimality gap by a factor of 8 for $p = 3$ as compared to Ref. [9].

Table 5.2. The median quality of solution for median variance across different distributions and for multiple p achieved through our introduced methods when compared with the previous study[9] for $n = 14$. As variance, s^2, grows, the optimality gaps achieved by our presented techniques are almost an order of magnitude better in comparison to the method in Ref. [9].

s^2	p	$r_{opt} - r_{[9]}$	$r_{opt} - r_D$	$r_{opt} - r_{\arctan}$
0.5697 [Uniform+]	1	0.00166	0.00828	0.0068
	2	0.01053	0.00827	0.00604
	3	0.01172	0.01112	0.01077
0.9557 [Exponential]	1	0.00572	0.00265	0.00357
	2	0.04149	0.00611	0.00455
	3	0.05012	0.0081	0.00843
1.1364 [Uniform±]	1	0.00183	0.00160	0.00045
	2	0.02124	0.00258	0.00142
	3	0.02928	0.00306	0.00282
2.4811 [Cauchy]	1	0.17524	0.04775	0.03344
	2	0.20788	0.04098	0.03867
	3	0.23958	0.03238	0.02967

The optimality gaps obtained by considering our presented parameters setting rules are of comparable values for each p. However, the performance of the method of Ref. [9] deteriorates as p increases.

From the values shown in Table. 5.2, it can be observed that when the variance is small, the performance obtained using the method in the prior work [9] is comparable to that of our methods. However, as variance increases the solution qualities achieved by the methods introduced in this work beat those of the previous work [9].

5.7 Discussion

In this chapter, we propose heuristic parameter setting rules for QAOA, inspired by a formal connection between weighted and unweighted MaxCut on regular graphs. For $p = 1$,

we derive explicit expressions for the parameter γ that maximizes the cost function in the weighted case. Our analysis of MaxCut at $p = 1$ rigorously proves that the globally-optimal γ are small, providing additional justification for this commonly used assumption [6], [7], [10], [12]. For $p \geq 1$, we show explicitly how the energy landscape and, consequently, the optimal parameters scale between the weighted and unweighted cases. As we prove the concentration of the QAOA objective, our results apply with high probability to any random weighted MaxCut instance.

Additionally, we consider the biased SK problem and rigorously show that it has a trivial solution in the infinite-size limit, unless the mean of the weight distribution falls sufficiently fast with the number of vertices. This investigation was inspired by the connection between SK and MaxCut on regular graphs, and the observation that the closed-form iterations that we use for QAOA do not apply to complete graphs. However, it appears that, unlike standard SK, the analysis of QAOA performance is unlikely to lead to significant insights when the weights are biased.

Our observation that QAOA parameters $\boldsymbol{\gamma}$ have to decrease with problem size is an instantiation of a broader principle, namely that parameterized quantum circuits are not scale-independent. Similar results have been observed for quantum kernel methods [48], [49] and quantum neural network initialization [50]. A unification of these observations into a general theory of parameterized quantum circuits is a tempting prospect, though it would require the development of novel mathematical techniques.

References

[1] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, 2002.

[2] D. Herman, C. Googin, X. Liu, *et al.*, "A survey of quantum computing for finance," *arXiv:2201.02773*, 2022. DOI: 10.48550/ARXIV.2201.02773. [Online]. Available: https://arxiv.org/abs/2201.02773.

[3] T. Hogg and D. Portnov, "Quantum optimization," *Information Sciences*, vol. 128, no. 3-4, pp. 181–197, 2000. DOI: 10.1016/s0020-0255(00)00052-9. [Online]. Available: https://doi.org/10.1016/s0020-0255(00)00052-9.

[4] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv:1411.4028*, 2014.

[5] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, 2019.

[6] S. Boulebnane and A. Montanaro, "Solving boolean satisfiability problems with the quantum approximate optimization algorithm," *arXiv:2208.06909*, 2022.

[7] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, "The quantum approximate optimization algorithm at high depth for maxcut on large-girth regular graphs and the sherrington-kirkpatrick model," en, *Proceedings of the Conference on the Theory of Quantum Computation, Communication and Cryptography*, vol. 7, pp. 1–21, 2022. DOI: 10.4230/LIPICS.TQC.2022.7. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/16514/.

[8] M. B. Hastings, "A classical algorithm which also beats $\frac{1}{2}+\frac{2}{\pi}\frac{1}{\sqrt{D}}$ for high girth max-cut," *arXiv:2111.12641*, 2021. eprint: arXiv:2111.12641.

[9] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, "Parameter transfer for quantum approximate optimization of weighted MaxCut," *ACM Transactions on Quantum Computing*, vol. 4, no. 3, pp. 1–15, Apr. 2023. DOI: 10.1145/3584706. [Online]. Available: https://doi.org/10.1145/3584706.

[10] S. Boulebnane, X. Lucas, A. Meyder, S. Adaszewski, and A. Montanaro, "Peptide conformational sampling using the quantum approximate optimization algorithm," *arXiv:2204.01821*, 2022.

[11] S. Brandhofer, D. Braun, V. Dehn, *et al.*, "Benchmarking the performance of portfolio optimization with QAOA," *Quantum Information Processing*, vol. 22, no. 1, Dec. 2022. DOI: 10.1007/s11128-022-03766-5. [Online]. Available: https://doi.org/10.1007/s11128-022-03766-5.

[12] S. Boulebnane and A. Montanaro, "Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit," *arXiv:2110.10685*, 2021. eprint: arXiv:2110.10685.

[13] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, "The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size," *Quantum*, vol. 6, p. 759, Jul. 2022. DOI: 10.22331/q-2022-07-07-759. [Online]. Available: https://doi.org/10.22331/q-2022-07-07-759.

[14] A. Dembo, A. Montanari, and S. Sen, "Extremal cuts of sparse random graphs," *The Annals of Probability*, vol. 45, no. 2, Mar. 2017. DOI: 10.1214/15-aop1084. [Online]. Available: https://doi.org/10.1214/15-aop1084.

[15] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," *arXiv:1811.08419*, 2018.

[16] M. Streif and M. Leib, "Training the quantum approximate optimization algorithm without access to a quantum processing unit," *Quantum Science and Technology*, vol. 5, no. 3, p. 034 008, 2020.

[17] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Physical Review X*, vol. 10, p. 021 067, 2020. DOI: 10.1103/PhysRevX.10.021067.

[18] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *IEEE High Performance Extreme Computing Conference*, 2019, pp. 1–8. DOI: 10.1109/hpec.2019.8916288.

[19] X. Lee, Y. Saito, D. Cai, and N. Asai, "Parameters fixing strategy for quantum approximate optimization algorithm," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, Oct. 2021. DOI: 10.1109/qce52317.2021.00016. [Online]. Available: https://doi.org/10.1109/qce52317.2021.00016.

[20] S. H. Sack and M. Serbyn, "Quantum annealing initialization of the quantum approximate optimization algorithm," *Quantum*, vol. 5, p. 491, 2021.

[21] O. Amosy, T. Danzig, E. Porat, G. Chechik, and A. Makmal, "Iterative-free quantum approximate optimization algorithm using neural networks," *arXiv:2208.09888*, 2022.

[22] D. Lykov, R. Schutski, A. Galda, V. Vinokur, and Y. Alexeev, "Tensor network quantum simulator with step-dependent parallelization," *arXiv:2012.02430*, 2020.

[23] M. Medvidović and G. Carleo, "Classical variational simulation of the quantum approximate optimization algorithm," *npj Quantum Information*, vol. 7, no. 101, 2021. DOI: 10.1038/s41534-021-00440-z.

[24] R. Shaydulin and S. M. Wild, "Exploiting symmetry reduces the cost of training QAOA," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–9, 2021. DOI: 10.1109/tqe.2021.3066275.

[25] R. Shaydulin and Y. Alexeev, "Evaluating quantum approximate optimization algorithm: A case study," in *Tenth International Green and Sustainable Computing Conference*, 2019. DOI: 10.1109/IGSC48788.2019.8957201.

[26] F. G. S. L. Brandão, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," *arXiv:1812.04170*, 2018.

[27] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, "Parameter concentrations in quantum approximate optimization," *Physical Review A*, vol. 104, no. 1, Jul. 2021. DOI: 10.1103/physreva.104.l010401. [Online]. Available: https://doi.org/10.1103/physreva.104.l010401.

[28] P. C. Lotshaw, T. S. Humble, R. Herrman, J. Ostrowski, and G. Siopsis, "Empirical performance bounds for quantum approximate optimization," *Quantum Information Processing*, vol. 20, no. 12, p. 403, 2021. DOI: 10.1007/s11128-021-03342-3.

[29] A. Galda, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, "Transferability of optimal QAOA parameters between random graphs," *arXiv:2106.07531*, 2021.

[30] X. Lee, N. Xie, D. Cai, Y. Saito, and N. Asai, "A depth-progressive initialization strategy for quantum approximate optimization algorithm," *arXiv:2209.11348*, 2022.

[31] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to optimize variational quantum circuits to solve combinatorial problems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2367–2375, 2020. DOI: 10.1609/aaai.v34i03.5616.

[32] G. Verdon, M. Broughton, J. R. McClean, *et al.*, "Learning to learn with quantum neural networks via classical neural networks," *arXiv:1907.05415*, 2019.

[33] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Reinforcement-learning-based variational quantum circuits optimization for combinatorial problems," *arXiv:1911.04574*, 2019. eprint: arXiv:1911.04574.

[34] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, "Reinforcement-learning-assisted quantum optimization," *Physical Review Research*, vol. 2, no. 3, Sep. 2020. DOI: 10.1103/physrevresearch.2.033446. [Online]. Available: https://doi.org/10.1103/physrevresearch.2.033446.

[35] M. Alam, A. Ash-Saki, and S. Ghosh, "Accelerating quantum approximate optimization algorithm using machine learning," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, Mar. 2020. DOI: 10.23919/date48585.2020.9116348. [Online]. Available: https://doi.org/10.23919/date48585.2020.9116348.

[36] J. Yao, L. Lin, and M. Bukov, "Reinforcement learning for many-body ground-state preparation inspired by counterdiabatic driving," *Physical Review X*, vol. 11, no. 3, Sep. 2021. DOI: 10.1103/physrevx.11.031070. [Online]. Available: https://doi.org/10.1103/physrevx.11.031070.

[37] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Physical Review A*, vol. 97, no. 2, 2018. DOI: 10.1103/physreva.97.022304.

[38] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, "The quantum approximate optimization algorithm and the Sherrington–Kirkpatrick model at infinite size," *arXiv:1910.08187*, 2020.

[39] J. Wurtz and D. Lykov, "The fixed angle conjecture for QAOA on regular MaxCut graphs," *arXiv:2107.00677*, 2021.

[40] S. A. Hadfield, *Quantum algorithms for scientific computing and approximate optimization*. Columbia University, 2018.

[41] P. Glasserman, *Monte Carlo methods in financial engineering*. Springer, 2004, vol. 53.

[42] W. Rudin, *Real and Complex Analysis*. McGraw-Hill, 1974.

[43] W. Rudin, *Principles of mathematical analysis*. McGraw-hill, 1976.

[44] J. Basso, D. Gamarnik, S. Mei, and L. Zhou, "Performance and limitations of the QAOA at constant levels on large sparse hypergraphs and spin glass models," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, Oct. 2022. DOI: 10.1109/focs54457.2022.00039.

[45] T. Dominguez, "Introduction to the SK model," 2021. [Online]. Available: https://www.math.toronto.edu/joaqsan/Resources/tomas_april2.pdf.

[46] R. Shaydulin, K. Marwaha, J. Wurtz, and P. C. Lotshaw, "QAOAKit: A toolkit for reproducible study, application, and verification of QAOA," in *Second International Workshop on Quantum Computing Software*, 2021. DOI: 10.1109/QCS54837.2021.00011.

[47] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, "The quantum approximate optimization algorithm at high depth for maxcut on large-girth regular graphs and the sherrington-kirkpatrick model," *arXiv:2110.14206*, 2021.

[48] R. Shaydulin and S. M. Wild, "Importance of kernel bandwidth in quantum machine learning," *Physical Review A*, vol. 106, no. 4, Oct. 2022. DOI: 10.1103/physreva.106.042407. [Online]. Available: https://doi.org/10.1103/physreva.106.042407.

[49] A. Canatar, E. Peters, C. Pehlevan, S. M. Wild, and R. Shaydulin, "Bandwidth enables generalization in quantum kernel models," *arXiv:2206.06686*, 2022.

[50] K. Zhang, L. Liu, M.-H. Hsieh, and D. Tao, "Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 612–18 627, 2022.

6. TEXNET: A DEEP NEURAL NETWORK FOR THE DECOMPOSITION OF HYPERSPECTRAL THERMAL IMAGES

The contents of this chapter are adapted from the article 'F. Bao, X. Wang, S. H. Sureshbabu, et al., "Heat-assisted detection and ranging," *Nature*, Forthcoming, 2023. DOI: 10.1038/s41586-023-06174-6'.

6.1 Introduction

Thermography, the detection and analysis of thermal radiation has emerged as an important domain of research especially due to its feature of witnessing the surrounding environment without any visible illumination. This has propelled applications in security, environmental monitoring, medical imaging to name a few. The state-of-the-art machine perception utilizing active sensors like sonar, radar and LiDAR to enhance camera vision is not a viable methodology as the number of intelligent agents scales up. *Quasi*-passive approaches like cameras are an alternative but they rely on ambient illumination. Furthermore, cameras cannot compete with human perception even though important strides [1] have been made recently based on deep learning [2], [3]. It causes phenomena like phantom braking [4] in automated vehicles due to the visual ambiguity and lack of physical context in perception. Exploiting omnipresent heat signals could be a new frontier for scalable perception and thermal imaging [5]–[8] has well-known advantages, *e.g.*, to see through the darkness or solar glare as well as bad weather [9]. Physical attributes of the scene, namely, temperature (T, physical status), emissivity (e, material fingerprint) and texture (X, surface geometry) are mixed in photon streams, as constantly emit and scatter thermal radiation leading to texture-less images popularly known as the 'ghosting effect'. Ghosting limits thermal imaging only to night vision enhancement without any specificity even when combined with AI algorithms.

A method titled *HADAR*[10] overcomes the ghosting effect by decomposing the heat signal into temperature, emissivity, and texture (TeX decomposition). The goal of this chapter is to introduce a technique that performs the aforementioned decomposition. A

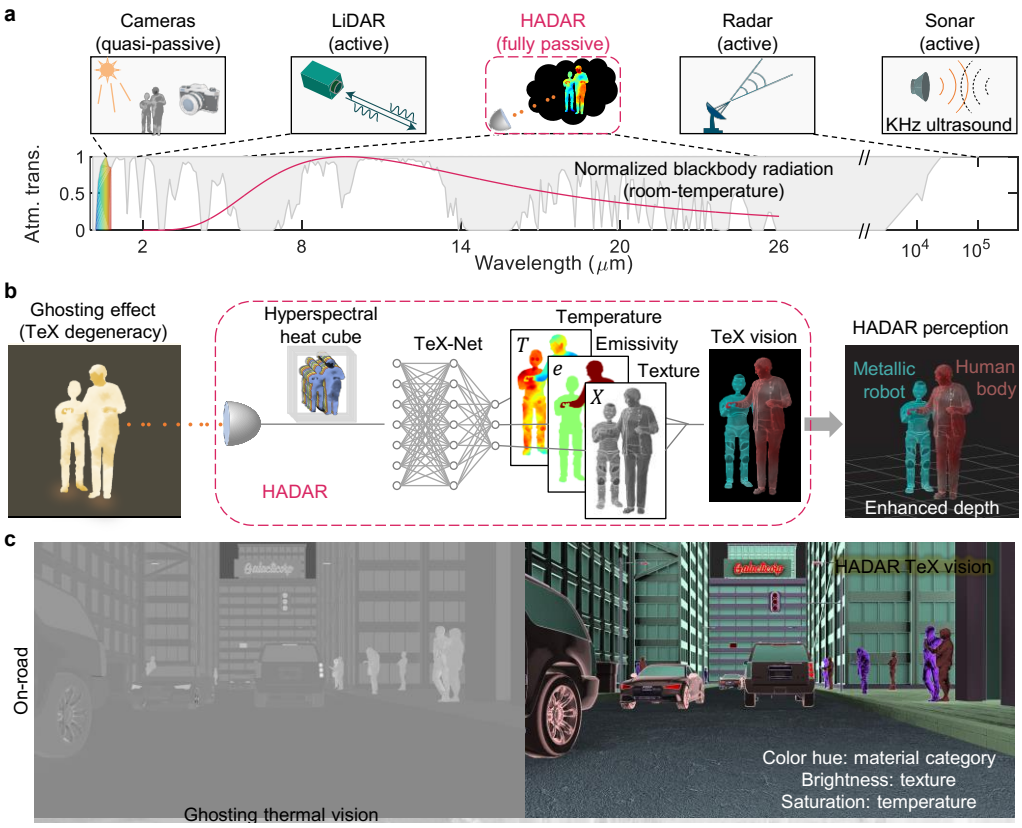

Figure 6.1. a) Fully passive HADAR makes use of heat signals, as opposed to active sonar, radar, LiDAR, and quasi-passive cameras. Atmospherical transmittance window (white area) and temperature of the scene determine the working wavelength of HADAR. b) HADAR takes thermal photon streams as input, records hyperspectral-imaging heat cubes, breaks the TeX degeneracy through TeX-Net, and generates TeX vision for improved detection and ranging. c) TeX vision demonstrated on our HADAR database indicates that HADAR perceives textures through the darkness with a comprehensive understanding of the scene.

physics-driven semantic segmentation is performed to achieve improved performance against AI-enhanced thermal sensing.

6.2 TeX decomposition and TeX vision

We address the ghosting effect with an approach we call TeX decomposition, which vividly recovers the texture from cluttered heat signal and also accurately disentangles temperature and emissivity at the Cramér-Rao bound. Representing these decluttered TeX attributes in HSV color space (Hue $= e$, Saturation $= T$, Brightness $= X$) leads to a paradigm shift of TeX vision with physical context for machine perception (Fig. 6.1b). TeX vision empowers

AI algorithms to reach information-theoretic bounds, which has thus far been elusive for traditional RGB or thermal vision. Fig. 6.1c shows TeX vision for on- and off-road scenes at night overcoming the ghosting effect. Our demonstrations of HADAR include detection and ranging based on TeX vision, for both real-world level HADAR database and outdoor experiments.

For intuitive clarity, we first explain the origin of the ghosting effect using an example of thermal radiation (visible) from a light bulb. Fig. 6.2 shows Monte Carlo path tracing simulations of rays emanating from a bulb, with the reflection of environmental emission taken into account. Geometric textures on the bulb surface can be seen only when the bulb is off. We emphasize that this texture revealed by reflection is completely lost in direct emission when the bulb is switched on, a familiar scenario from daily experience. Since every object in a complex scene emits and scatters thermal radiation, they are thermal light sources with no texture like a shining bulb. The total heat signal leaving an object α has two additive contributions,

$$S_{\alpha\nu} = \mathrm{e}_{\alpha\nu} B_\nu(T_\alpha) + [1 - \mathrm{e}_{\alpha\nu}] X_{\alpha\nu}, \tag{6.1}$$

where the first term is direct thermal emission (textureless), and the second term carrying texture is the environmental emission entering the detector after scattering from the object. Here ν in the subscript denotes wavenumber (spectrum) dependence. The key difference with a shining bulb is that blackbody radiation B_ν is fundamentally governed by Planck's law and cannot be switched off. Textureless thermal imaging is thus widely regarded as impossible to use for quantitative insight about a scene. The environmental thermal illumination on object α from all other objects β is given by $X_{\alpha\nu} = \sum_{\beta\neq\alpha} V_{\alpha\beta} S_{\beta\nu}$, with $V_{\alpha\beta}$ being the thermal lighting factor. Ghosting effect is exacerbated for high emissivity materials in nature such as skin and plants ($\mathrm{e} \approx 1$) as the total collected signal consists of dominant direct emission and weak scattered signal. We note that $S_{\alpha\nu}$ is invariant under joint transformations of temperature T, emissivity e and texture X, which we address as TeX degeneracy. In addition to the ghosting effect, this TeX degeneracy renders the separation of temperature-emissivity as a major roadblock [11] to quantitative thermal sensing.

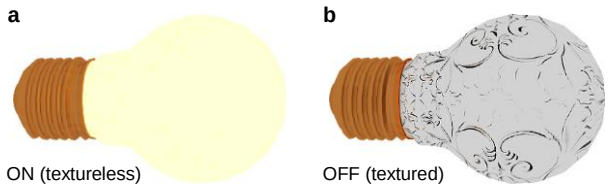

Figure 6.2. Monte Carlo path tracing simulation of a light bulb to explain the 'ghosting effect'. Geometric texture on a light bulb can only be seen when the bulb is off whereas this texture is completely missing when it is glowing. The blackbody radiation can never be turned off leading to loss of texture for thermal images. This ghosting effect presents the long-standing roadblock for heat-assisted machine perception.

We recover the texture by breaking TeX degeneracy and discretizing spectral emissivity $\mathrm{e}_{\alpha\nu}$ into $\mathrm{e}_\nu(m_\alpha)$ in a material library, $\mathcal{M} = \{\mathrm{e}_\nu(m)|m = 1, 2, \cdots, M\}$, that contains all possible spectral emissivity in the scene. This opportunity of dimensional reduction is available naturally in smart applications where materials usually have industrial standards [12]. Our approach of TeX-Net uses Eq. (6.1) to design physics-based loss and uses a 3D convolutional neural network to learn spatio-spectral features, in recovering texture X, temperature T and emissivity e.

6.3 Inverse Mapping in Applications

Recall that the heat signal leaving object α is $S_{\alpha\nu} = \mathrm{e}_{\alpha\nu}B_\nu(T_\alpha) + [1 - \mathrm{e}_{\alpha\nu}]X_{\alpha\nu}$, with $X_{\alpha\nu} = \sum_{\beta\neq\alpha} V_{\alpha\beta}S_{\beta\nu}$. Starting with T_α, $\mathrm{e}_{\alpha\nu}$, and $V_{\alpha\beta}$ for all compact and finite objects, Monte Carlo path tracing can solve $S_{\alpha\nu}$ asymptotically with the l-th order scattering-cutoff solution $\tilde{S}^l_{\alpha\nu}$. The residual error $\delta_{\alpha l} \equiv |\tilde{S}^l_{\alpha\nu} - S_{\alpha\nu}| \to 0$ when l increases. Let k denote the maximum number of significant environmental objects considered in the scene, whose spectral emissivity must be one out of M curves in the material library $\mathcal{M} = \{\mathrm{e}_\nu(m)|m = 1, 2, \cdots, M\}$. The parameter set $\{klM\}$ determines the complexity of the inverse problem and also controls the accuracy of the solution of $T_\alpha, \mathrm{e}_\nu(m_\alpha)$ and $X_{\alpha\nu}$ for given observed $S_{\alpha\nu}$. Note that $S_{\beta\nu}$ in texture $X_{\alpha\nu} = \sum_{\beta\neq\alpha} V_{\alpha\beta}S_{\beta\nu}$ is partially observed as $S_{\alpha\nu}$. We downsample $S_{\alpha\nu}$ into k spectra to approximately describe k most significant environmental objects. For example, each heat cube in the HADAR-Street dataset has dimension of $\mathrm{H} \times \mathrm{W} \times \mathrm{C} = 1080 \times 1920 \times 54$, H being height, W being width, and C being channel (number of wavenumbers). In our

demonstration, we considered $k = 2$ environmental objects. To do so, we spatially split images H $\times$ W into 2×1 quadrants, each quadrant having dimension of $540 \times 1920 \times 54$. Then we spatially average each quadrant into a spectrum of length 54, *i.e.*, for each of 54 channels, we average the 540×1920 sub-image and get its mean value. These 2 spectra, denoted as $S_{1\nu}$ and $S_{2\nu}$, are equivalent objects of the environment, and now the texture is given by

$$X_{\alpha\nu} = V_{\alpha 1} S_{1\nu} + V_{\alpha 2} S_{2\nu} + \delta_{\alpha\nu,2}, \tag{6.2}$$

where the residue $\delta_{\alpha\nu,k}$ is the summation of all sub-leading contributions,

$$\delta_{\alpha\nu,k} \equiv \sum_{\beta \neq 1,2,\cdots,k} V_{\alpha\beta} S_{\beta\nu}, \tag{6.3}$$

and we have $\delta_{\alpha\nu,k} \to 0$ as k increases. Spectral radiance of external objects beyond the view can also be provided in addition to the down-sampled spectra in Eq. (6.2). For example, the sky is usually a significant environmental object in open areas but may not be captured in the image.

The part of the scattered signal that people are familiar with in daily experience is originated only from sky illumination, and hence texture distillation is necessary to recast X. The distillation process is to turn off radiation of other environmental objects than the sky in Eq. (6.2) and then evaluate the HADAR constitutive equation in a forward manner without direct emission. Due to the cutoff on the number of environmental objects, $\delta_{\alpha\nu,k}$ also contains textures, and hence the final texture is a fusion of the distilled $\bar{X}$ and the residue $\delta_{\alpha\nu,k}$.

By substituting down-sampled $S_{\alpha\nu}$ into $S_{\beta\nu}$, we have taken into account infinite scattering ($l = \infty$). The number of environmental objects k is restricted by the number of channels C, $k - 1 + 2 \leq \mathrm{C}$, in order to have a determined solution (number of variables is no more than the number of equations). With the texture model Eq. (6.2) ignoring the residue, HADAR identifiability and material estimation theory in the last section can be readily generalized to any number of objects and infinite scattering bounces. The unknown parameter set to be estimated becomes $\{g, T_\alpha, V_{\alpha 1}, V_{\alpha 2}, \cdots, V_{\alpha k}\}$.

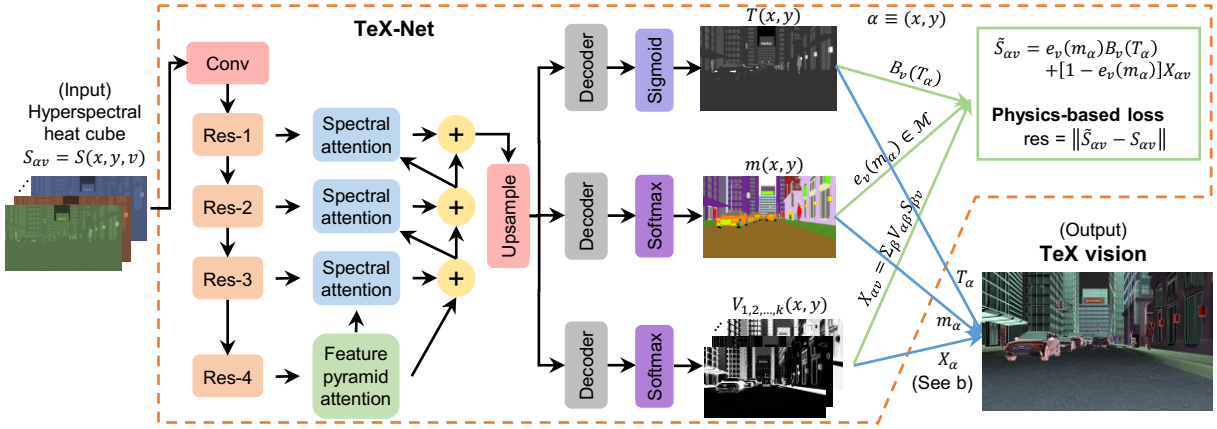

Figure 6.3. Architecture of TeX-Net for inverse TeX decomposition. TeX-Net can either be trained with ground truth T, m, and V in supervised learning, or alternatively, with material library $\mathcal{M}$, Planck's law $B_\nu(T_\alpha)$, and the mathematical structure of $X_{\alpha\nu}$ in unsupervised learning. In supervised learning, the loss function is a combination of individual losses with regularization hyper-parameters. In unsupervised learning, the loss function defined on the re-constructed heat cube is based on physics models of the heat signal. In practice, a hybrid loss function with T, e, V contributions (50%) in addition to the physics-based loss (50%) is used. The sigmoid function before the temperature is to set lower and upper bounds of temperature. Res-1/2/3/4 are Res-Net with downsampling. The plus symbol is addition operation followed by upsampling.

6.4 TeXNet

The architecture of the proposed TeXNet is described in Fig. 6.3.

TeX-Net is physics-inspired for three aspects. Firstly, TeX decomposition of heat cubes relies on both spatial patterns and spectral thermal signatures. This inspires the adoption of spectral and pyramid (spatial) attention layers [13] in the UNet model. Secondly, due to TeX degeneracy, the mathematical structure, $X_{\alpha\nu} = \sum_\beta V_{\alpha\beta} S_{\beta\nu}$, has to be specified to ensure the uniqueness of inverse mapping, and hence it is essential to learn thermal lighting factors V instead of texture X. That is, TeX-Net cannot be trained end-to-end. X_α is constructed with V and $S_{\beta\nu}$ indirectly, where $S_{\beta\nu}$ is the down-sampled $S_{\alpha\nu}$ to approximate k most significant environmental objects. Thirdly, the material library $\mathcal{M}$ and its dimension are key to the network.

6.4.1 Training data and training strategy

Our TeX-Net was trained on the HADAR database (https://github.com/FanglinBao/HADAR). The HADAR database includes dissimilar scenes like Crowded Street, Highway, Suburb, Countryside, Indoor, Forest, Desert, etc., covering most common road conditions that HADAR may find applications in. The 11^{th} dataset is a real-world off-road scene with heat cube dimension Height $\times$ Width $\times$ Channel $= 260 \times 1500 \times 49$, while the first 10 scenes are synthetic with heat cube dimension Height $\times$ Width $\times$ Channel $= 1080 \times 1920 \times 54$. The channels in the real-world scene correspond to the $5^{\text{th}} \sim 53^{\text{rd}}$ channels of the synthetic scenes. The HADAR database mimics self-driving situations, with the HADAR sensor(s) either mounted at the positions of headlights, or on the top of the automated vehicles, or on robot helpers. Each scene has 5 frames for each camera, and there are 30 different kinds of materials in total in the HADAR database. For the Street, Suburb, Rocky Terrain, and the Real-World Off-Road scenes, TeX, RGB and IR images are provided for the purpose of ranging. The Street scene has a long animation version (100 frames, 12 channels). For the real-world experimental scene, HADAR sensor is a pushbroom hyperspectral imager that can produce 256 spectral bands. The heat cubes have been interpolated into 49 channels to match the channels in synthetic scenes. Only 49 channels of all the scenes are used to train TeX-Net. Full technical details about the HADAR database, such as, ray depth, field of view, material properties, and so on, are available in the readme file along with the database.

We split the HADAR database (11 scenes) into training set (80% data) + validation set (20% data) to train the TeX-Net with 5-fold cross validation. Due to limited experimental data, we manually split the database, instead of randomly splitting, to ensure the same diversity of the validation set and training set. Explicitly, in each fold, one frame per view of each scene was selected for validation. We used a hybrid loss with half supervised loss and half physics loss, and we trained TeX-Net for 40K epochs. Since the real-world scene (260*1500) has a different image size with the synthetic scenes (1080*1920), we used random crop (256*256) in training. The network was trained using the number of workers of 8 and a batch size of 20. The learning rate started at 0.001 and dropped by a factor of 10 at 30000 and 37000 epochs. ADAM optimizer was used with the default momentum parameters. The

used ResNet50 model was pre-trained on the ImageNet dataset. For synthetic scenes, ground truth temperature and material are synthesized along with the heat cubes. Thermal lighting factors are solved out with least-squares fitting as the ground truth.

6.4.2 Saliency maps

Saliency map shows the relevant region that is used to predict the desired quantity (material classification). The Saliency map for material classification e(m) in TeX-Net is evaluated by Grad-CAM [14] and given in Fig. 6.4.

6.4.3 Performance and training loss

The supervised training loss and performance of TeX-Net on Street-Long-Animation are shown in Fig. 6.5.

In unsupervised learning with physics-based loss, TeX-Net searches the best matching TeX for a given signal S. As in practice, standard materials still bear small amount of variations in property, the material library is an approximation of the scene into several material classes. Therefore, the number of materials in the library affects the overall accuracy of TeX decomposition. For the HADAR-Street dataset which consists of 20 materials, we show the role of material library, by approximating the scene into much fewer material classes and analyzing the overall physics-based loss. For example, in using 3 materials in the library, we only keep the most distinct emissivities of glass and brass, and approximate all other materials as blackbody. This approximation will surely lead to biased temperature and texture, but as the number of materials increases, the loss will decrease. The analysis is given in Fig. 6.6. With increasing materials, TeX-Net is trained from the beginning, and training convergence is not significantly slower.

The TeX-Net performance on the HADAR database is shown in Fig. 6.7. Training loss curves and the TeX-Net codes are available along with the HADAR database.

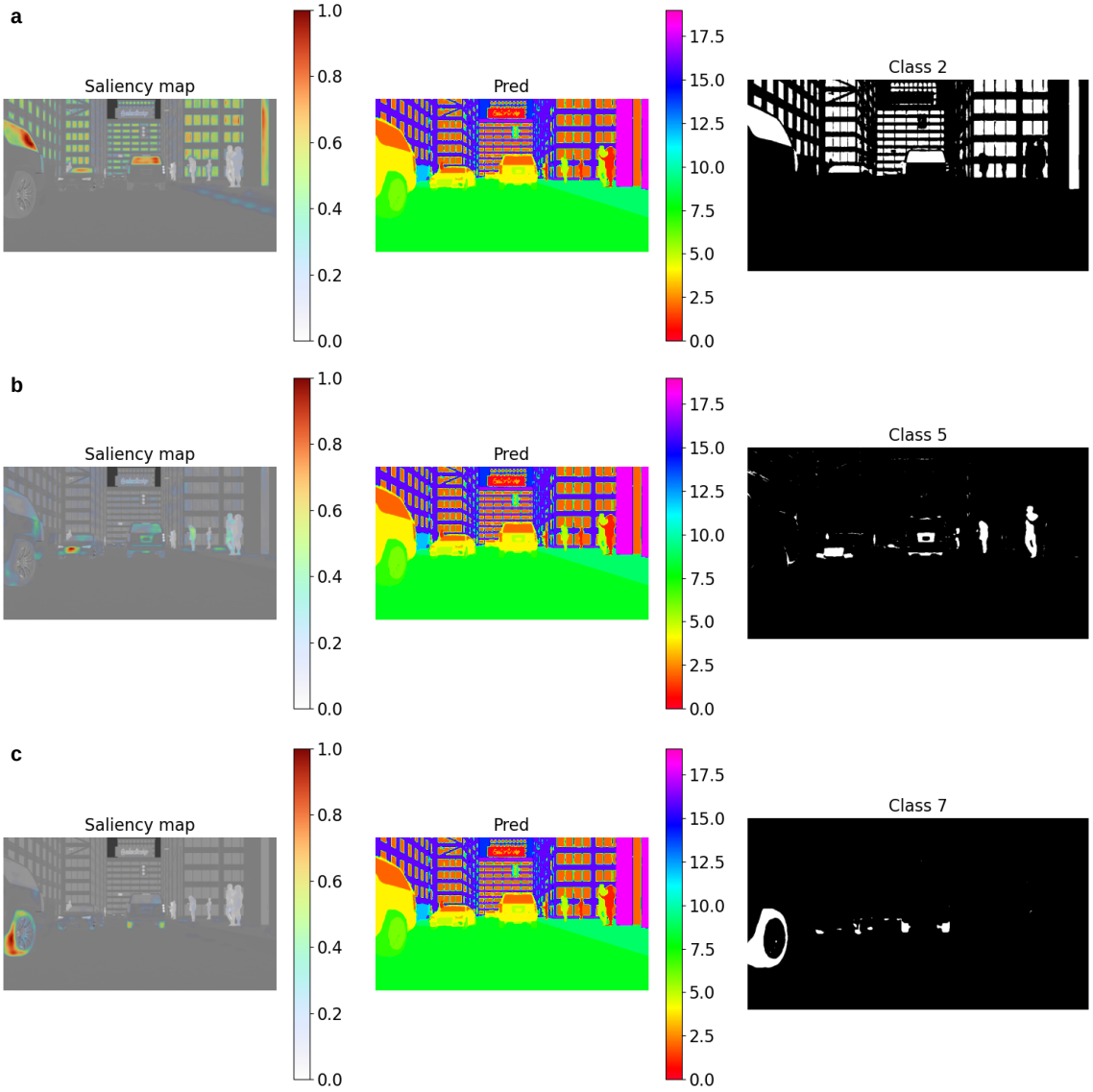

Figure 6.4. Saliency map of TeX-Net in supervised learning. The active region in Saliency maps is localized and highly correlated with the corresponding material region (last column), indicating that TeX-Net has properly learnt spatial and spectral features for material classification. 3 samples out of 20 materials are shown. a, Saliency map for class 2, window glass. b, Saliency map for class 5, aluminum. c, Saliency map for class 7, tire. Pred: material index prediction of TeX-Net.

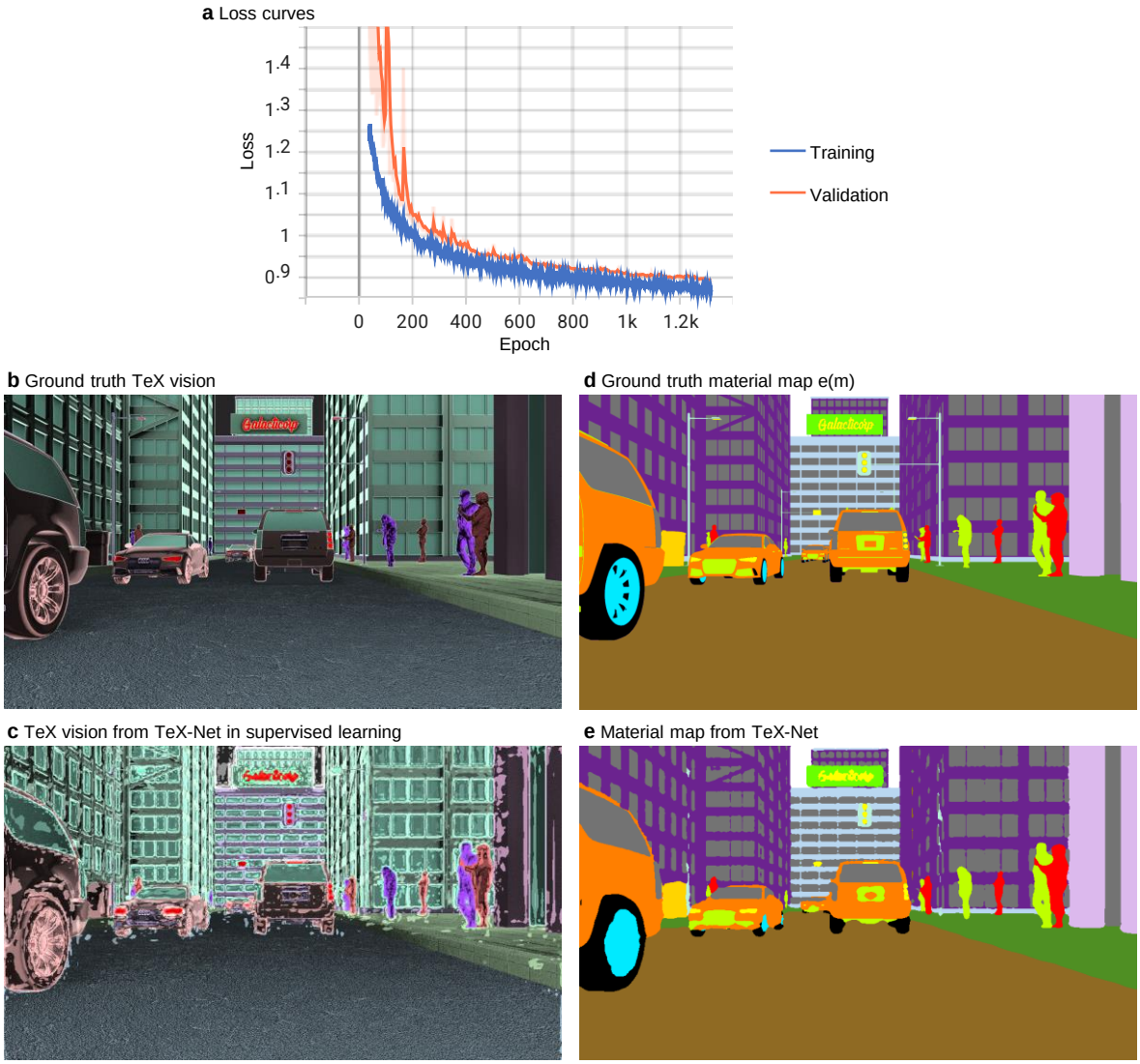

Figure 6.5. a, Loss curves in supervised learning showing the convergence of TeX-Net training. b, Ground truth TeX vision. c, Output of TeX-Net. d, Ground truth material map. e, Material map from TeX-Net. The comparisons of TeX-Net output with the ground truth show that TeX-Net is indeed able to do TeX decomposition. Small prediction errors in temperature lead to texture error in brightness, and hence there are some noisy spots observed in c. This can be improved by imposing sophisticated smooth constraint on temperature and harder training in the future. This training was done on the Street Long-Animation dataset in the HADAR database.

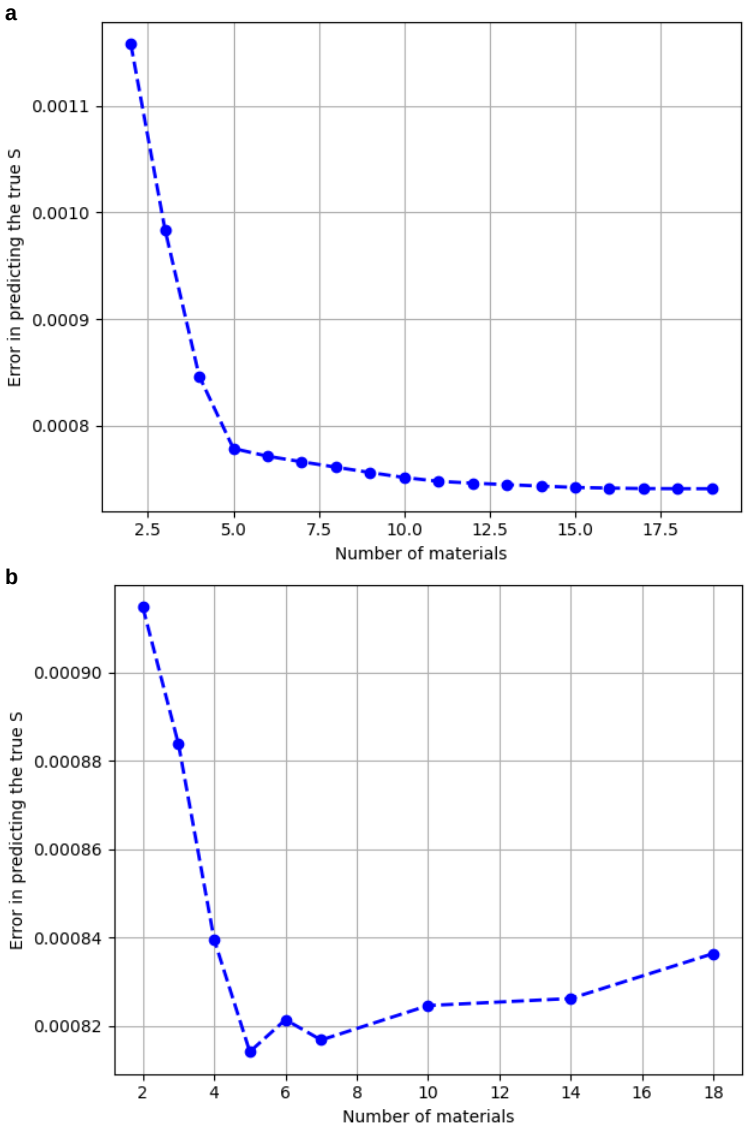

Figure 6.6. Physics-based loss decreases as the number of materials in the library increases. a, materials are added into the library with a greedy approach, and pixels are classified into those material classes based on visual similarity. Temperature and thermal lighting factors are solved out accordingly. b, Pixels are classified into material classes with a neural network (TeX-Net). TeX-Net finds more accurate TeX decomposition, and again, we can see that with more materials in the library, the physics-based loss is lower. The error in (b) after 5 materials is noise.

Figure 6.7. TeX vision comparison between the ground truth and TeX-Net output. TeX-Net was trained with hybrid loss, an equal-weight combination of supervised loss, and physics-based loss. The HADAR database was split into a training set (80% data) and a validation set(20% data) for 5-fold cross-validation. The TeX-Net was trained with 40K epochs.

6.5 Conclusion

We propose HADAR for fully-passive and physically-aware machine perception. We develop a deep neural network that can take advantage of the spectral and spatial resolution of hyperspectral thermal images to decompose into the corresponding physical attributes of the scene, which successfully tackles the ghosting effect observed in thermal imaging. Though practical challenges exist, such as library collection, on-fly calibration, real-time data acquisition, and functionality-cost optimization, we believe HADAR will lead to a new chapter in the Fourth Industrial Revolution with applications in autonomous navigation, healthcare, agriculture, wildlife monitoring, geosciences, and defense industry.

References

[1] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, 2018, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2018.05.018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, p. 436, May 2015. DOI: 10.1038/nature14539.

[3] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015, ISSN: 0036-8075. DOI: 10.1126/science.aaa8415.

[4] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20, Virtual Event, USA: Association for Computing Machinery, 2020, pp. 293–308, ISBN: 9781450370899. DOI: 10.1145/3372297.3423359.

[5] R. Gade and T. B. Moeslund, "Thermal cameras and applications: A survey," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 245–262, Jan. 2014, ISSN: 1432-1769. DOI: 10.1007/s00138-013-0570-5.

[6] K. Tang, K. Dong, C. J. Nicolai, *et al.*, "Millikelvin-resolved ambient thermography," *Sci. Adv.*, vol. 6, no. 50, eabd8688, 2020. DOI: 10.1126/sciadv.abd8688.

[7] M. Henini and M. Razeghi, *Handbook of infrared detection technologies*. Elsevier, 2002.

[8] A. Haque, A. Milstein, and F.-F. Li, "Illuminating the dark spaces of healthcare with ambient intelligence," *Nature*, vol. 585, no. 7824, pp. 193–202, Sep. 2020, ISSN: 1476-4687. DOI: 10.1038/s41586-020-2669-y.

[9] K. Beier and H. Gemperlein, "Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation," *Aerosp. Sci. Technol.*, vol. 8, no. 1, pp. 63–71, 2004, ISSN: 1270-9638. DOI: 10.1016/j.ast.2003.09.002.

[10] F. Bao, X. Wang, S. H. Sureshbabu, *et al.*, "Heat-assisted detection and ranging," *Nature*, Forthcoming, 2023. DOI: 10.1038/s41586-023-06174-6.

[11] A. Gillespie, S. Rokugawa, T. Matsunaga, J. S. Cothern, S. Hook, and A. B. Kahle, "A temperature and emissivity separation algorithm for advanced spaceborne thermal emission and reflection radiometer (aster) images," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 4, pp. 1113–1126, Jul. 1998, ISSN: 1558-0644. DOI: 10.1109/36.700995.

[12] A. Baldridge, S. Hook, C. Grove, and G. Rivera, "The aster spectral library version 2.0," *Remote Sens. Environ.*, vol. 113, no. 4, pp. 711–715, 2009, ISSN: 0034-4257. DOI: 10.1016/j.rse.2008.11.007.

[13] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," in *British Machine Vision Conference 2018, Newcastle, UK*, 2018, p. 285. [Online]. Available: http://bmvc2018.org/contents/papers/1120.pdf.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

7. CONCLUSION

The promise of AI and quantum technology has ushered the development of several hardware and software methodologies, that have pushed the boundaries of science. Furthering the development of computational techniques relevant to the physical sciences has been the driving force of this thesis.

To summarize, we have presented purely classical and quantum-classical hybrid techniques to tackle complex problems that require state-of-the-art classical and quantum hardware.

We show through the utilization of a unique ansatz that resembles a classical ML model, how any eigenstate of choice can be extracted on quantum hardware. The emphasis was laid on two-dimensional materials and their corresponding band diagrams were extracted to display how the proposed technique can have a profound impact on the study of electronics.

Then, the previously presented method was extended to capture quantum phase transitions, a fundamentally many-body phenomenon. A simulation was performed to calculate the critical point of the Quantum Rabi Model on quantum hardware, thereby extending the domain of quantum simulation to a light-matter interacting system.

The above approaches involved performing a variational calculation, wherein the parameters, required to construct a unitary that can be placed on a quantum circuit to perform the relevant computation, were optimized on classical computers. This optimization might become intensive for certain problems. Therefore, we proposed a parameter-setting strategy for a quantum heuristic algorithm known as QAOA, applied to a general class of combinatorial optimization problems.

Finally, we looked at a classical deep learning-based approach to extract the relevant components of a thermal image. The importance of solving this inverse problem became apparent in the case of autonomous navigation during the night.

In essence, we have tried to present new methods in the near term to solve some of the pressing problems in physics and chemistry with the state-of-the-art available hardware. In the case of the classical neural network, a more extensive dataset with thermal images can help improve the quality of solutions and in turn, the way we approach autonomous

navigation. For quantum techniques presented in this thesis, a reliable physical quantum computer can help propose new algorithms and improve upon the existing ones to discover more impactful quantum applications.

A QUANTUM MACHINE LEARNING FOR VALENCE BAND CALCULATIONS [APPENDIX]

A.1 General Tight-Binding Hamiltonian for a system of two sublattices

We begin our discussion with a general tight-binding (TB) Hamiltonian for a system of consisting of unit cells of two conjoint sublattices (say A and B) with one atom per sublattice. For simplification, we shall consider the case where each atom contributes only one orbital even though this restriction can be relaxed in a straight-forward extension. Our TB Hamiltonian in the basis of the contributing orbitals is:

$$\begin{aligned} \mathrm{H} &= \sum_{m,n,\sigma,\sigma'} \epsilon^{A}_{m,n,\sigma,\sigma'} a^{\dagger}_{m\sigma} a_{n\sigma'} + \sum_{m,n,\sigma,\sigma'} \epsilon^{B}_{m,n,\sigma,\sigma'} b^{\dagger}_{m\sigma} b_{n\sigma'} \\ &+ \sum_{m,n,\sigma,\sigma'} t_{m,n,\sigma,\sigma'} (a^{\dagger}_{m,\sigma} b_{n,\sigma'} + h.c.) \end{aligned} \tag{A1}$$

where $\epsilon^{A}_{m,n,\sigma,\sigma'}$, $\epsilon^{B}_{m,n,\sigma,\sigma'}$ are the interaction matrix elements within each of the respective sublattices (either A or B) and $t_{m,n,\sigma,\sigma'}$ (assumed to be real) denotes the hopping interaction between the two-sublattices. $a^{\dagger}_{m\sigma}$ creates an electron in the mth atom (also mth orbital) with spin σ in sublattice A. Similar definition holds also for $b^{\dagger}_{m\sigma}$ except it caters to the B sublattice. The following properties of these operators will be very useful later

$$\{a^{\dagger}_{m\sigma}, a_{n\sigma'}\} = \delta_{m,n,\sigma,\sigma'} \tag{A2}$$

$$\{b^{\dagger}_{m\sigma}, b_{n\sigma'}\} = \delta_{m,n,\sigma,\sigma'} \tag{A3}$$

$$\{a^{\dagger}_{m\sigma}, a^{\dagger}_{n\sigma'}\} = \{b^{\dagger}_{m\sigma}, b^{\dagger}_{n\sigma'}\} = 0 \tag{A4}$$

$$\{b^{\dagger}_{m\sigma}, a^{\dagger}_{n\sigma'}\} = \{b^{\dagger}_{m\sigma}, a_{n\sigma'}\} = 0 \tag{A5}$$

$$a_{n\sigma}|0\rangle = 0 \tag{A6}$$

$$b_{n\sigma}|0\rangle = 0 \tag{A7}$$

Using Eq.A5 is equivalent to assuming that the overlap metric between the sublattices A and B is identity. Now since Eq.A1 is banded, to afford dimensionality reduction and ease of diagonalization let us define Fourier transform of the operators $a^{\dagger}_{n\sigma'}$ and $b^{\dagger}_{n\sigma'}$ as follows:

$$c^{\dagger}_{k\sigma} = \frac{1}{\sqrt{N}} \sum_{m} \mathrm{e}^{\mathbf{ik\cdot R_{mA}}} a^{\dagger}_{m\sigma} \tag{A8}$$

$$c^{\dagger}_{k\sigma'} = \frac{1}{\sqrt{N}} \sum_{m} \mathrm{e}^{\mathbf{ik\cdot R_{mA}}} a^{\dagger}_{m\sigma'} \tag{A9}$$

$$d^{\dagger}_{k\sigma} = \frac{1}{\sqrt{N}} \sum_{m} \mathrm{e}^{\mathbf{ik\cdot R_{mB}}} b^{\dagger}_{m\sigma} \tag{A10}$$

$$d^{\dagger}_{k\sigma'} = \frac{1}{\sqrt{N}} \sum_{m} \mathrm{e}^{\mathbf{ik\cdot R_{mB}}} b^{\dagger}_{m\sigma'} \tag{A11}$$

where $\mathbf{R_{mB}}$ and $\mathbf{R_{mA}}$ are real-space lattice vectors of the two sublattices and $\mathbf{k}$ is the wavevector that belongs to the 1st Brillouin zone of the corresponding reciprocal lattice. Using Eq.A8, A9, A10, A11 and properties listed in Eq.A2, A3, A4, A5, A6, A7, it is now possible to construct matrix elements of the following forms:

- $\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$

$$\begin{aligned}
\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle &= \frac{1}{N} \sum_{m,n,\sigma_1,\sigma_2} \sum_{p.q} \mathrm{e}^{\mathbf{ik\cdot(R_{qA}-R_{pA})}} \langle a_{p\sigma'} a^{\dagger}_{m\sigma_1} a_{n\sigma_2} a^{\dagger}_{q\sigma}\rangle \epsilon^{A}_{m,n,\sigma_1,\sigma_2} \\
&= \frac{1}{N} \sum_{m,n,\sigma_1,\sigma_2} \sum_{p.q} \mathrm{e}^{\mathbf{ik\cdot(R_{qA}-R_{pA})}} \delta_{nq}\delta_{\sigma\sigma_2}\delta_{mp}\delta_{\sigma'\sigma_1} \epsilon^{A}_{m,n,\sigma_1,\sigma_2} \\
&= \frac{1}{N} \sum_{p,q} \mathrm{e}^{\mathbf{ik\cdot(R_{qA}-R_{pA})}} \epsilon^{A}_{p,q,\sigma,\sigma'}
\end{aligned} \tag{A12}$$

- $\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$

$$\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle = \frac{1}{N} \sum_{p,q} \mathrm{e}^{\mathbf{ik\cdot(R_{qB}-R_{pB})}} \epsilon^{B}_{p,q,\sigma,\sigma'} \tag{A13}$$

- $\langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$

$$\begin{aligned}\langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle &= \frac{1}{N}\sum_{m,n,\sigma_1,\sigma_2}\sum_{p.q} \mathrm{e}^{\mathbf{ik}\cdot(\mathbf{R_{qB}}-\mathbf{R_{pA}})}\langle a_{p\sigma'}a^{\dagger}_{m\sigma_1}b_{n\sigma_2}b^{\dagger}_{q\sigma}\rangle t_{m,n,\sigma_1,\sigma_2} \\ &= \frac{1}{N}\sum_{m,n,\sigma_1,\sigma_2}\sum_{p.q} \mathrm{e}^{\mathbf{ik}\cdot(\mathbf{R_{qB}}-\mathbf{R_{pA}})}\delta_{nq}\delta_{\sigma\sigma_2}\delta_{mp}\delta_{\sigma'\sigma_1}t_{m,n,\sigma_1,\sigma_2} \\ &= \frac{1}{N}\sum_{p,q} \mathrm{e}^{\mathbf{i\vec{k}}\cdot(\mathbf{R_{qB}}-\mathbf{R_{pA}})}t_{p,q,\sigma,\sigma'} \end{aligned} \tag{A14}$$

With these matrix elements, we can cast the Hamiltonian in the basis of operators defined in Eq.A8, A9, A10, A11. To proceed further we need to now specialize to the exact geometry of the lattice which shall be discussed in the next section.

A.2 Honeycomb lattices: Graphene and h-BN

Using the matrix elements derived in Eq.A12, A13, A14 we can now deduce the Hamiltonian used in this work for graphene and h-BN upto third nearest neighbor interaction. Both graphene and h-BN possesses the similar lattice structure, a representative prototype of which is given in Fig.A1. The real-space lattice unit vectors are $\mathbf{a_1}$, $\mathbf{a_2}$ are also displayed. The primitive vectors of the real lattice of are given by $\mathbf{a}_1 = a\left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$, $\mathbf{a}_2 = a\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$ where $a = |\mathbf{a}_1| = |\mathbf{a}_2| = 2.47$ Å is the lattice constant for h-BN and $a = 2.55$ Å for monolayer graphene.

Nearest-neighbor interaction

For nearest-neighbor interaction only in hexagonal honeycomb lattices, it is easy to appreciate from the geometry in Fig.A1(a) that atoms in A sublattice share a vertex with those at sublattice B only and vice versa. So the following substitutions need to be made

- $\epsilon^{A}_{p,q,\sigma,\sigma'} = \epsilon^{A}_{p}\delta_{pq}\delta_{\sigma\sigma'}$

  Substituting in Eq.A12 we get

$$\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle = \sum_{\mathrm{p}}\frac{\epsilon^{\mathrm{A}}_{\mathrm{p}}}{\mathrm{N}}\delta_{\sigma,\sigma'} \tag{A15}$$

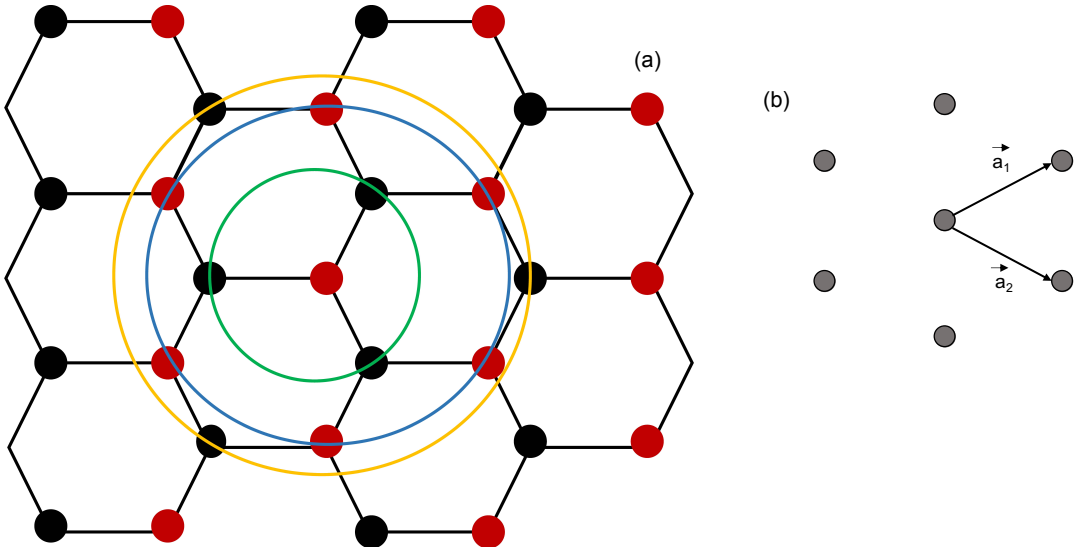

Figure A1. (a) Structure of the honeycomb lattice. The green circle shows the nearest neighbors (three in this case), the blue circle shows the second-nearest neighbors (six in this case), and the orange circle shows the third-nearest neighbors (three in this case). (b) The unit vectors $\mathbf{a_1}$, $\mathbf{a_2}$ of the real space lattice are indicated.

- $\epsilon^B_{p,q,\sigma,\sigma'} = \epsilon^B_p \delta_{pq} \delta_{\sigma\sigma'}$

  Substituting in Eq.A13 we get

$$\langle 0|d_{k\sigma'} \mathrm{H} \mathrm{d}^\dagger_{\mathrm{k}\sigma}|0\rangle = \sum_{\mathrm{p}} \frac{\epsilon^{\mathrm{B}}_{\mathrm{p}}}{\mathrm{N}} \delta_{\sigma,\sigma'} \tag{A16}$$

- $t_{p,q,\sigma,\sigma'} = t_1$ if $\mathbf{R_{qB}} = \mathbf{R_{pA}} + \mathbf{d_1}, \mathbf{R_{pA}} + \mathbf{d_2}, \mathbf{R_{pA}} + \mathbf{d_3}$

  Substituting in Eq.A14 we get

$$\begin{aligned} \langle 0|c_{k\sigma'} \mathrm{H} \mathrm{d}^\dagger_{\mathrm{k}\sigma}|0\rangle &= \delta_{\sigma\sigma'} \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(-\mathbf{d_1})} t_1 (1 + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(\mathbf{d_2}-\mathbf{d_1})} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(\mathbf{d_3}-\mathbf{d_1})}) \\ &= \delta_{\sigma\sigma'} \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(-\mathbf{d_1})} t_1 (1 + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}}) \end{aligned}$$

This form of the matrix elements has actually been deduced before in [1], [2]. From geometry the nearest neighbor length vectors are $\mathbf{d_1} = (\mathbf{0}, \frac{\mathbf{1}}{\sqrt{\mathbf{3}}})\mathbf{a}, \mathbf{a_2} = \mathbf{d_2} - \mathbf{d_1}, \mathbf{a_3} = \mathbf{d_3} - \mathbf{d_1}$.

Second-nearest neighbor interaction

From Fig.A1(a) it is clear that every next nearest neighbor of atom A is also only atom A and vice-versa of atoms of B sublattice as well. Inclusion of second nearest neighbor interaction thus only modifies the matrix elements $\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ and $\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ only. Let us look at each of them

- $\epsilon^{A}_{p,q,\sigma,\sigma'} = \epsilon^{A}_{p}\delta_{pq}\delta_{\sigma\sigma'}$
  $= t_2$ if $\mathbf{R_{qA}} = \mathbf{R_{pA}} \pm \mathbf{a_1}$,$\mathbf{R_{qA}} = \mathbf{R_{pA}} \pm \mathbf{a_2}$, $\mathbf{R_{qA}} = \mathbf{R_{pA}} \pm (\mathbf{a_1} - \mathbf{a_2})$.
  Substituting in Eq.A12 we get

$$\begin{aligned}\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle &= \sum_{p}(\frac{\epsilon^{A}_{p}}{N})\delta_{\sigma,\sigma'} + t_2(\mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}} \\ &+ \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(\mathbf{a_1}-\mathbf{a_2})} + \mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot(\mathbf{a_1}-\mathbf{a_2})})\delta_{\sigma,\sigma'} \qquad \text{(A17)}\end{aligned}$$

- $\epsilon^{B}_{p,q,\sigma,\sigma'} = \epsilon^{B}_{p}\delta_{pq}\delta_{\sigma\sigma'}$
  $= \tilde{t_2}$ if $\mathbf{R_{qB}} = \mathbf{R_{pB}} \pm \mathbf{a_1}$,$\mathbf{R_{qB}} = \mathbf{R_{pB}} \pm \mathbf{a_2}$, $\mathbf{R_{qB}} = \mathbf{R_{pB}} \pm (\mathbf{a_1} - \mathbf{a_2})$.
  Substituting in Eq.A13 we get

$$\begin{aligned}\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle &= \sum_{p}(\frac{\epsilon^{B}_{p}}{N})\delta_{\sigma,\sigma'} + \tilde{t_2}(\mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}} \\ &+ \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(\mathbf{a_1}-\mathbf{a_2})} + \mathrm{e}^{-\mathbf{i}\mathbf{k}\cdot(\mathbf{a_1}-\mathbf{a_2})})\delta_{\sigma,\sigma'} \qquad \text{(A18)}\end{aligned}$$

- $\langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle = \delta_{\sigma\sigma'}\mathrm{e}^{\mathbf{i}\mathbf{k}\cdot(-\mathbf{d_1})}\mathrm{t}_1(1 + \mathrm{e}^{\mathbf{i}\tilde{\mathbf{k}}\cdot\tilde{\mathbf{a_1}}} + \mathrm{e}^{\mathbf{i}\mathbf{k}\cdot\mathbf{a_2}})$

The matrix elements of the type $\langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ do not change at all and is equal to the value obtained in the nearest-neighbor case.

Third-nearest neighbor interaction

It is evident from Fig.A1(a) that third-nearest neighbor interaction only interconnects of the atoms in A and B sublattices only and hence matrix elements of the kind $\langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ will be exclusively changed while elements of the kind $\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ and $\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ will involve participation upto second nearest neighbor only.

- $t_{p,q,\sigma,\sigma'} = t_1$ if $\mathbf{R_{qB}} = \mathbf{R_{pA}} + \mathbf{d_1}, \mathbf{R_{pA}} + \mathbf{d_2}, \mathbf{R_{pA}} + \mathbf{d_3}$ and
  $= t_3$ if $\mathbf{R_{qB}} = \mathbf{R_{pA}} \pm (\mathbf{a_1} - \mathbf{a_2}), \mathbf{R_{pA}} + \mathbf{a_2} + \mathbf{a_1}$
  Substituting these in Eq.A14 we get

$$\begin{aligned} \langle 0|c_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle &= t_1(1 + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a_1}} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a_2}}) + t_3(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)} \\ &+ \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1+\mathbf{a}_2)}) \end{aligned} \tag{A19}$$

The two other matrix elements i.e. $\langle 0|c_{k\sigma'}\mathrm{Hc}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$, $\langle 0|d_{k\sigma'}\mathrm{Hd}^{\dagger}_{\mathrm{k}\sigma}|0\rangle$ remain the same as the second-nearest neighbor case.

Now we are in a position to construct all the matrix elements of h-BN and monolayer graphene using interactions up to the third nearest neighbor.

h-BN

$$\begin{aligned} \sum_p \frac{\epsilon_p^A}{N}\delta_{\sigma,\sigma'} &= t_b \\ \sum_p \frac{\epsilon_p^B}{N}\delta_{\sigma,\sigma'} &= t_N \\ \mathrm{H}_1 &= t_1\left(1 + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}\right) \\ \mathrm{H}_2 &= t_2\left(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)} + \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1} + \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}\right) \\ \tilde{\mathrm{H}}_2 &= \tilde{t_2}\left(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)} + \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1} + \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}\right) \\ \mathrm{H}_3 &= t_3\left(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)} + \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1+\mathbf{a}_2)}\right) \end{aligned}$$

$$\mathrm{H} = \begin{bmatrix} t_b + \mathrm{H}_2 & 0 & 0 & \mathrm{H}_1 + \mathrm{H}_3 \\ 0 & t_b + \mathrm{H}_2 & \mathrm{H}_1 + \mathrm{H}_3 & 0 \\ 0 & \mathrm{H}_1^{\dagger} + \mathrm{H}_3^{\dagger} & t_n + \tilde{\mathrm{H}}_2 & 0 \\ \mathrm{H}_1^{\dagger} + \mathrm{H}_3^{\dagger} & 0 & 0 & t_n + \tilde{\mathrm{H}}_2 \end{bmatrix},$$

Table 1. Tight binding parameters for h-BN

t_b (eV)	t_n (eV)	t_1 (eV)	t_2 (eV)	t_3 (eV)
2.46	-2.55	2.16	0.04	0.08

Monolayer graphene

$$\begin{aligned}
\sum_p \frac{\epsilon_p^A}{N}\delta_{\sigma,\sigma'} &= t_C \\
\mathrm{H}_1 &= t_1\left(1+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}\right) \\
\mathrm{H}_2 &= t_2\left(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)}+\mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_1}+\mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{a}_2}\right) \\
\mathrm{H}_3 &= t_3\left(\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1-\mathbf{a}_2)}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_2-\mathbf{a}_1)}+\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot(\mathbf{a}_1+\mathbf{a}_2)}\right)
\end{aligned}$$

$$\mathrm{H}=\begin{bmatrix}
t_C+\mathrm{H}_2 & 0 & 0 & \mathrm{H}_1+\mathrm{H}_3 \\
0 & t_C+\mathrm{H}_2 & \mathrm{H}_1+\mathrm{H}_3 & 0 \\
0 & \mathrm{H}_1^\dagger+\mathrm{H}_3^\dagger & t_C+\mathrm{H}_2 & 0 \\
\mathrm{H}_1^\dagger+\mathrm{H}_3^\dagger & 0 & 0 & t_C+\mathrm{H}_2
\end{bmatrix},$$

In the case of graphene, we also model electronic interaction between opposite spins through a Hubbard Hamiltonian with the repulsion parameter being denoted by U. Since the average number of electrons with spin-up is taken as 1 and the average number of electrons with spin-down is taken as 0. Therefore, U enters the Hamiltonian only on the down-spin diagonal terms.

$$\mathrm{H} = \begin{bmatrix} t_C + \mathrm{H}_2 & 0 & 0 & \mathrm{H}_1 + \mathrm{H}_3 \\ 0 & t_C + \mathrm{H}_2 + U & \mathrm{H}_1 + \mathrm{H}_3 & 0 \\ 0 & \mathrm{H}_1^\dagger + \mathrm{H}_3^\dagger & t_C + \mathrm{H}_2 + U & 0 \\ \mathrm{H}_1^\dagger + \mathrm{H}_3^\dagger & 0 & 0 & t_C + \mathrm{H}_2 \end{bmatrix} \tag{A20}$$

Table 2. Hubbard model parameters for graphene

t_C (eV)	t_1 (eV)	t_2 (eV)	t_3 (eV)	U (eV)
1.994	2.86	-0.236	0.252	9.3

A.3 Scaling

After the single qubit rotations (R_y) and before the Controlled-Controlled Rotations ($C-C-R$), the probability distribution corresponding to a specific σ^z, h can be written as:

$$\frac{\mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}}{\sum_{\sigma^z,h} \mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}} \tag{A21}$$

Once the $C-C-R$ is applied for the first time, the probability distribution with the corresponding ancilla qubit being in $|1\rangle$ is:

$$\frac{\mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}}{\sum_{\sigma^z,h} \mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}} \times \frac{\mathrm{e}^{\frac{1}{k}(w_{\mathrm{ij}}\sigma_{\mathrm{i}}^z h_{\mathrm{j}})}}{\mathrm{e}^{\frac{1}{k}|w_{\mathrm{ij}}|}} \tag{A22}$$

After all the $C-C-R$ are applied, then the probability distribution with all the ancilla qubits being in $|1\rangle$ is:

$$\frac{\mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}}{\sum_{\sigma^z,h} \mathrm{e}^{\frac{1}{k}(\sum_{\mathrm{i}} a_{\mathrm{i}}\sigma_{\mathrm{i}}^z + \sum_{\mathrm{j}} b_{\mathrm{j}} h_{\mathrm{j}})}} \times \prod_{\mathrm{i,j}} \frac{\mathrm{e}^{\frac{1}{k}(w_{\mathrm{ij}}\sigma_{\mathrm{i}}^z h_{\mathrm{j}})}}{\mathrm{e}^{\frac{1}{k}|w_{\mathrm{ij}}|}}$$

$$= \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j)}}{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j)}} \times \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij} \sigma_i^z h_j)}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}}$$

$$= \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j + \sum_{i,j} w_{ij} \sigma_i^z h_j)}}{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j)} e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \tag{A23}$$

By summing all possible states, the probability of getting all ancilla qubits to be in $|1\rangle$ is given by:

$$\frac{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j + \sum_{i,j} w_{ij} \sigma_i^z h_j)}}{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j)} e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \tag{A24}$$

Since, $e^{w_{ij}\sigma_i^z h_j} \geq e^{-abs w_{ij}}$, the term $e^{w_{ij}\sigma_i^z h_j}$ in Eq.(A24) can be replaced with $e^{-|w_{ij}|}$.

This results in the successful probability P to be:

$$P = \frac{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j + \sum_{i,j} w_{ij} \sigma_i^z h_j)}}{\sum_{\sigma^z,h} e^{\frac{1}{k}(\sum_i a_i \sigma_i^z + \sum_j b_j h_j)} e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \geq \frac{e^{\frac{-1}{k}(\sum_{ij}|w_{ij}|)}}{e^{\frac{1}{k}(\sum_{ij}|w_{ij}|)}} = \frac{1}{e^{\frac{1}{k}(\sum_{ij} 2|w_{ij}|)}} \tag{A25}$$

By choosing $\max(\sum_{ij} \frac{|w_{ij}|}{2}, 1)$, the lower bound of the probability for successful sampling becomes a constant equal to e^{-4}.

A.4 Implementation Details

1. If n is the number of visible units and m is the number of hidden units, then,

   (a) The number of qubits required are:

   - 2 qubits for visible units (n)
   - 2 qubits for hidden units (m)
   - 4 ancilla qubits (n+m)

   (b) The number of gates used are:

   - 4 single qubit rotations (n+m)
   - 4 Controlled-Controlled rotations (n×m)
   - 24 X(bit-flip) gates (6×n×m)

2. The parameter are updated through gradient descent with a learning rate equal to 0.01.

3. The number of measurements

$$= \text{number of iterations} = \begin{cases} \approx 30000, & \text{for classical and } qasm \text{ simulations without warm start} \\ \approx 500, & \text{for classical and } qasm \text{ simulations with warm start} \\ \approx 500, & \text{for IBM-Q implementation} \end{cases}$$

B QUANTUM MACHINE LEARNING FOR EIGENSTATE FILTRATION [APPENDIX]

B.1 Proof of Theorem 2.1

The proof of feasibility of general penalty functions is known in optimization theory. Since both our objective function and penalty term are quadratic forms herein we construct an original, formal and a simple proof for Theorem 2.1.

Let us recollect the cost function $\mathrm{F}(\lambda, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle)$ defined in text:

$$\mathrm{F}(\lambda, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle) = \langle\psi|\hat{\mathrm{H}}|\psi\rangle + \lambda\langle\psi|(\hat{\mathrm{O}} - \omega)^2|\psi\rangle \tag{B1}$$

B.2 Definitions

We re-iterate the following definitions as considered in the main text

1. $\hat{H} \in \mathbb{C}^{d\times d}$ and $\hat{H} = \hat{H}^\dagger$. This is the Hamiltonian operator in the problem and we denote the spectrum of $\hat{H}$ by $\vec{\sigma} = [\sigma_0, \sigma_1, ...\sigma_n]^T$ where $\sigma_0 \leq \sigma_1 \leq ....\sigma_n$. We shall assume that the entries of $\hat{H}$ in the chosen basis is finite.

2. $\hat{O} \in \mathbb{C}^{d\times d}$ and $\hat{O} = \hat{O}^\dagger$ is an user-defined operator for the problem. ω is an eigenvalue of operator $\hat{O}$. We denote the spectrum of operator $(\hat{O} - \omega)^2$ as $\vec{\eta} = [\eta_0, \eta_1, \eta_2....\eta_n]^T$ where $\eta_0 \leq \eta_1 \leq \eta_2.... \leq \eta_n$. Further $\eta_{\mathrm{i}} \geq 0 \ \forall$ i as $(\hat{O} - \omega)^2 \succeq 0$ (positive-semidefinite by construction)

3. $Null(\hat{A}) = \{|x\rangle|\ \hat{A}|x\rangle = 0,\ \forall\ |x\rangle \in \mathbb{C}^d\}$ where $\hat{A}$ is any arbitrary operator $\in \mathbb{C}^{d\times d}$.

4. $\lambda \in \mathbb{R}_{++}$ is a penalty parameter

5. $|\psi\rangle \in \mathbb{C}^d$ is the state-vector of the system sought from the minimization scheme by training the neural network.

6. $\{\lambda_{\mathrm{i}}\}_{\mathrm{i}=1}^{\infty}$ is a sequence in the penalty parameter such that $\lambda_1 \leq \lambda_2 \leq \lambda_3 ..... \lambda_\infty \to \infty$

7. $P = \{|\psi_{\mathrm{i}}\rangle\}_{\mathrm{i}=1}^{\infty}$ such that $\forall\ |\psi_{\mathrm{i}}\rangle \in P$ the following is true.

$$|\psi_{\mathrm{i}}\rangle = \arg\min_{\psi} \mathrm{F}(\lambda_{\mathrm{i}}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle) \tag{B2}$$

   In other words P is the set of minimizers for Eq. B1 for each penalty parameter $\lambda \in \{\lambda_{\mathrm{i}}\}_{\mathrm{i}=1}^{\infty}$.

8. $|\psi^*\rangle = \lim_{\mathrm{i}\to\infty} |\psi_{\mathrm{i}}\rangle$ be the limit point of a convergent sequence in P

9. All vectors $\in \mathbb{C}^d$ discussed below will be considered normalized unless otherwise stated

Using the definitions above we construct the following lemmas.

Lemma 1. *For any* $|\psi\rangle \in \mathbb{C}^d$, $\langle\psi|\hat{A}|\psi\rangle \leq \sqrt{Tr(\hat{A}^\dagger \hat{A})}$ *where* $\hat{A}$ *is any arbitrary hermitian operator* $\in \mathbb{C}^{d\times d}$.

Proof. Let us denote the variance of operator $\hat{A}$ as $Var(\hat{A})$ evaluated in an arbitrary state $|\psi\rangle \in \mathbb{C}^d$. $Var(\hat{A})$ by definition is always non-negative. From this we can claim

$$\begin{aligned} Var(\hat{A}) = \langle\psi|\hat{A}^2|\psi\rangle - (\langle\psi|\hat{A}|\psi\rangle)^2 \geq 0 \quad \because (\text{by definition}) \\ \langle\psi|\hat{A}^2|\psi\rangle \geq (\langle\psi|\hat{A}|\psi\rangle)^2 \\ \langle\psi|\hat{A}^\dagger\hat{A}|\psi\rangle \geq (\langle\psi|\hat{A}|\psi\rangle)^2 \quad \because (\hat{A} = \hat{A}^\dagger) \end{aligned} \tag{B3}$$

Now let us consider a complete set of eigenvectors of $\hat{A}^\dagger\hat{A}$ denoted as $\{|s_{\mathrm{i}}\rangle\}_{\mathrm{i}=1}^{d}$ with corresponding eigenvalues $\{s_{\mathrm{i}}\}_{\mathrm{i}=1}^{d}$ which are non-negative as $\hat{A}^\dagger\hat{A} \succeq 0$. One can resolve the state $|\psi\rangle$ in the basis $\{|s_{\mathrm{i}}\rangle\}_{\mathrm{i}=1}^{d}$ as follows

$$|\psi\rangle = \sum_{\mathrm{i}=1}^{d} \langle s_{\mathrm{i}}|\psi\rangle |s_{\mathrm{i}}\rangle \tag{B4}$$

Using Eq. B4 to express $\langle\psi|\hat{A}^\dagger\hat{A}|\psi\rangle$ we have

$$\begin{aligned}
&\langle\psi|\hat{A}^\dagger\hat{A}|\psi\rangle \\
&= \sum_{\mathrm{i}=1}^{d} s_\mathrm{i}|\langle s_\mathrm{i}|\psi\rangle|^2 \\
&\leq \sum_{\mathrm{i}=1}^{d} s_\mathrm{i} \qquad \because s_\mathrm{i} \geq 0 \text{ and } 0 \leq |\langle \mathrm{s_i}|\psi\rangle|^2 \leq 1 \\
&= Tr(\hat{A}^\dagger A)
\end{aligned} \tag{B5}$$

We can thus make the following claim

$$\begin{aligned}
& Tr(\hat{A}^\dagger A) \\
& \geq \langle\psi|\hat{A}^\dagger\hat{A}|\psi\rangle \quad \because Eq.B5 \\
& \geq (\langle\psi|\hat{A}|\psi\rangle)^2 \quad \because Eq.B3 \\
\implies \langle\psi|\hat{A}|\psi\rangle &\leq \sqrt{Tr(\hat{A}^\dagger A)}
\end{aligned} \tag{B6}$$

□

Lemma 2. *$|\psi^*\rangle \in P$ and is a limit-point of the convergent sequence in P if $\langle\psi^*|(\hat{O} - \omega)^2|\psi^*\rangle = 0$*

Proof. Let us consider a state $|\psi'\rangle \in S$, where the set S is defined as

$$S = \{|x\rangle|\ \hat{O}|x\rangle = \omega|x\rangle\ \forall\ |x\rangle \in \mathbb{C}^d\} \tag{B7}$$

The following is then true

$$\langle\psi'|\hat{H}|\psi'\rangle = \langle\psi'|\hat{H}|\psi'\rangle + \lambda_k\langle\psi'|(\hat{O} - \omega)^2|\psi'\rangle \qquad \because |\psi'\rangle \in S \tag{B8}$$

$$= \mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi'\rangle) \tag{B9}$$

$$\leq \sqrt{Tr(\hat{H}^\dagger H)} \quad \because (\text{Lemma 1 using } \hat{\mathrm{A}} = \hat{\mathrm{H}}) \tag{B10}$$

Now since the $\hat{H}$ operator is assumed to have elements which are all finite (see definitions in B.2) (1), and since $\sqrt{Tr(\hat{H}^\dagger H)}$ is a polynomial on the matrix elements of $\hat{H}$, one can say $\mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi'\rangle)$ in Eq. B9 is always upper-bounded by a finite number (see Eq. B9 and Eq. B10). One should note that $\sqrt{Tr(\hat{H}^\dagger H)}$ being a trace property is invariant to the choice of basis for expressing the matrix elements of $\hat{H}$ and also independent of any state $|\psi'\rangle$ used for computing $\langle\psi'|\hat{H}|\psi'\rangle$. Also $\mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi'\rangle) \geq \mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi_\mathrm{k}\rangle)$ as $|\psi_k\rangle = \arg\min_\psi \mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle)$. This is true for any λ_k as $|\psi_k\rangle$ is the specific minimizer of the cost function in Eq. B1 for that λ_k and hence will produce a cost-function in Eq. B1 lower in value than with any state $|\psi'\rangle$. With this information we see

$$\lim_{k\to\infty} \mathrm{F}(\lambda_\mathrm{k}, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi_\mathrm{k}\rangle) \leq \sqrt{Tr(\hat{H}^\dagger H)}$$

$$\lim_{k\to\infty} \langle\psi_k|\hat{H}|\psi_k\rangle + \lim_{k\to\infty} \lambda_k \langle\psi_k|(\hat{O}-\omega)^2\rangle|\psi_k\rangle \leq \sqrt{Tr(\hat{H}^\dagger H)} \tag{B11}$$

$$\lim_{k\to\infty} \lambda_k \langle\psi_k|(\hat{O}-\omega)^2|\psi_k\rangle \leq \sqrt{Tr(\hat{H}^\dagger H)} - \lim_{k\to\infty} \langle\psi_k|\hat{H}|\psi_k\rangle \tag{B12}$$

$$\lim_{k\to\infty} \langle\psi_k|(\hat{O}-\omega)^2|\psi_k\rangle = \langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle = 0 \qquad (\lambda_k \to \infty) \tag{B13}$$

where in arriving at Eq. B13 from Eq. B12 we have used the fact that $\sqrt{Tr(\hat{H}^\dagger H)}$ is finite (as per definitions B.2 (1) and the fact that $\sqrt{Tr(\hat{H}^\dagger H)}$ is a polynomial on the matrix elements of $\hat{H}$) and $\langle\psi_k|\hat{H}\psi_k\rangle$ being a quadratic form is also upper-bounded using same Lemma 1 and hence is finite. Thus the RHS of Eq. B12 is a finite-upper bound on the LHS. Only way then the LHS of Eq. B12 can thus stay finite in the limit $\lambda \to \infty$ is when $\langle\psi_k|(\hat{O}-\omega)^2|\psi_k\rangle$ is pinned to zero. Since $|\psi^*\rangle = \lim_{k\to\infty} |\psi_k\rangle$ is the convergent limit point, the result immediately follows. □

Lemma 3. *$\langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle = 0$ if and only if $|\psi^*\rangle \in Null(\hat{O}-\omega)^2)$*

Proof. If-part

If $|\psi^*\rangle \in Null((\hat{O}-\omega)^2)$

$$(\hat{O}-\omega)^2|\psi^*\rangle = 0 \qquad (|\psi^*\rangle \in Null(\hat{O}-\omega)^2))$$
$$\langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle = 0$$

Only If-part

Let us define a set of eigenvectors of $(\hat{O}-\omega)^2$ as $\{|\eta_i\rangle\}_i$. Since the set is complete one can expand

$$\begin{aligned}|\psi^*\rangle &= \sum_i \langle\eta_i|\psi^*\rangle|\eta_i\rangle \\ &= \sum_{|\eta_i\rangle\in Null((\hat{O}-\omega)^2)} \langle\eta_i|\psi^*\rangle|\eta_i\rangle + \sum_{|\eta_i\rangle\notin Null((\hat{O}-\omega)^2)} \langle\eta_i|\psi^*\rangle|\eta_i\rangle \end{aligned} \tag{B14}$$

One can use Eq. B14 in $\langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle$ to arrive at

$$\begin{aligned}\langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle &= \sum_{|\eta_i\rangle\in Null((\hat{O}-\omega)^2)} 0|\langle\eta_i|\psi^*\rangle|^2 + \sum_{|\eta_i\rangle\notin Null((\hat{O}-\omega)^2)} \eta_i|\langle\eta_i|\psi^*\rangle|^2 \\ &= 0 \quad (\text{by condition}) \\ &\implies \langle\eta_i|\psi^*\rangle = 0 \ \forall\ |\eta_i\rangle \ \notin Null((\hat{O}-\omega)^2) \quad (\eta_i \geq 0, \ \textit{see } B.2(2)) \\ &\implies |\psi^*\rangle \in Null((\hat{O}-\omega)^2) \quad \because \text{ Using Eq.B14} \end{aligned} \tag{B15}$$

□

Lemma 4. *$Null(\hat{O}-\omega)) = Null(\hat{O}-\omega)^2)$*

Proof. This is actually trivial to show. For proof one can see [3] □

B.3 Theorem 2.1 in main text

Then using the Lemmas above the following is true.

Theorem B.1. *Let $\{\lambda_i\}_{i=1}^{\infty}$ be a sequence in the penalty parameter such that $\lambda_1 \leq \lambda_2 \leq \lambda_3.....\lambda_\infty \to \infty$ Also let $P = \{|\psi_i\rangle\}_{i=1}^{\infty}$ such that $\forall\ |\psi_i\rangle \in P$ the following is true.*

$$|\psi_i\rangle = \arg\min_{\psi} \mathrm{F}(\lambda_i, \hat{\mathrm{H}}, \hat{\mathrm{O}}, |\psi\rangle) \tag{B16}$$

In other words P is the set of minimizers for Eq. B1 for each penalty parameter $\lambda \in \{\lambda_i\}_{i=1}^{\infty}$ If $|\psi^*\rangle \in P$ is a limit-point of the convergent sequence $\{\psi_i\}_{i=1}^{\infty}$ in P i.e $|\psi^*\rangle = \lim_{i\to\infty}|\psi_i\rangle$ then $|\psi^*\rangle \in S$ (defined in lemma 2)

Proof. If $|\psi^*\rangle$ is a limit-point of the convergent sequence $\{\psi_{\mathrm{i}}\}^{\infty}$ in P then

$$\begin{aligned}
\langle\psi^*|(\hat{O}-\omega)^2|\psi^*\rangle = 0 \quad & \because \text{see Lemma 2} \\
\implies |\psi^*\rangle \in Null((\hat{O}-\omega)^2) \quad & \because \text{see Lemma 3} \\
\implies |\psi^*\rangle \in Null((\hat{O}-\omega)) \quad & \because \text{see Lemma 4} \\
\implies |\psi^*\rangle \in S \quad & \text{(B17)}
\end{aligned}$$

□

B.4 Deduction of a generic lower bound for successful sampling and characterization of k-parameter

After all the single qubit R_y rotations (parameterized by the bias vectors of the network $(\vec{a},\vec{b})$) and Controlled-Controlled Rotations ($C-C-R_y$) targeting the ancillas (parameterized by the interconnecting weights $\vec{W}$ between visible and hidden neurons), the state-vector $|\psi_{v,h,a}\rangle$ of the full set of $(m+n+m\times n)$ qubits is

$$|\psi_{v,h,a}\rangle = \sum_{(\vec{\sigma},\vec{h})} \sqrt{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})}|\vec{\sigma}\vec{h}\rangle_{vh} \otimes (\sqrt{(1-\eta(\vec{W},\vec{\sigma},\vec{h})}|\vec{0}\rangle_a + \sqrt{\eta(\vec{W},\vec{\sigma},\vec{h})}|\vec{1}\rangle_a) \quad \text{(B18)}$$

where the following definitions is used.

1. $|\psi_{v,h,a}\rangle$ is the combined state of the visible node qubits (abbreviated by subscript v), hidden node qubits (abbreviated by subscript h) and ancilla register (abbreviated by subscript a)

2. $(\vec{\sigma},\vec{h})$ denotes a sum over the 2^{m+n} bit strings where each $\{\sigma_{\mathrm{i}}\}$ or $\{h_{\mathrm{j}}\} \in \{1,-1\}$

3. $|\vec{\sigma}\vec{h}\rangle_{vh}$ is the 2^{m+n} dimensional state space of n visible node qubits and m hidden node qubits. Note the state $|0\rangle_v$ corresponds to $\sigma_{\mathrm{i}} = -1$ as mentioned in the main text. Similar statement holds for $h_{\mathrm{j}} = -1$ and $|0\rangle_h$ too.

4. The distribution $O(\vec{\sigma}, \vec{h}, \vec{a}, \vec{b})$ is

$$O(\vec{\sigma}, \vec{h}, \vec{a}, \vec{b}) = \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j)}}{\sum_{\vec{\sigma},\vec{h}} e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j)}} \tag{B19}$$

5. The distribution $\eta(\vec{W}, \vec{\sigma}, \vec{h})$ is

$$\eta(\vec{W}, \vec{\sigma}, \vec{h}) = \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij} \sigma_i h_j)}}{e^{\frac{1}{k}\sum_{i,j} |w_{ij}|}} \tag{B20}$$

6. $|\vec{0}\rangle_a$ and $|\vec{1}\rangle_a$ are the states of the $m \times n$ ancilla qubits (abbreviated as superscript a)

Thus from Eq. B18 we see that when all the qubits are measured, the probability of selecting a bit string $(\vec{\sigma}, \vec{h})$ **and** collapsing the ancilla qubits in state $|\vec{1}\rangle_a$ (only such states are important to this work as they are post-selected after measurement).

$$\begin{aligned} H((\vec{\sigma}, \vec{h}) \cap \vec{1}) &= O(\vec{\sigma}, \vec{h}, \vec{a}, \vec{b}) \eta(\vec{W}, \vec{\sigma}, \vec{h}) \\ &= \frac{e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j)}}{\sum_{\vec{\sigma},\vec{h}} e^{\frac{1}{k}(\sum_i a_i \sigma_i + \sum_j b_j h_j)}} \times \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij} \sigma_i h_j)}}{e^{\frac{1}{k}\sum_{i,j} |w_{ij}|}} \end{aligned} \tag{B21}$$

Now successful sampling would be an event wherein all ancilla would collapse to $|\vec{1}\rangle_a$. The probability of such events (denoted as $P_{success} = P(\vec{1}_a)$) irrespective of $(\vec{\sigma}, \vec{h})$ string selected can be obtained by marginalizing $H((\vec{\sigma}, \vec{h}) \cap \vec{1})$ **over all** bit-strings as follows:

$$
\begin{aligned}
P_{success} = P(\vec{1}_a) &= \sum_{\vec{\sigma},\vec{h}} H((\vec{\sigma},\vec{h}) \cap \vec{1}) \\
&= \sum_{\vec{\sigma},\vec{h}} \frac{e^{\frac{1}{k}(\sum_i a_i\sigma_i + \sum_j b_j h_j)}}{\sum_{\sigma,h} e^{\frac{1}{k}(\sum_i a_i\sigma_i + \sum_j b_j h_j)}} \times \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij}\sigma_i h_j)}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \qquad \text{(B22)} \\
&= \frac{\left\langle e^{\frac{1}{k}(\sum_{i,j} w_{ij}\sigma_i h_j)} \right\rangle_{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \\
&\geq \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij}\langle\sigma_i h_j\rangle_{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})})}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \qquad \because \text{(Jensen's inequality)} \qquad \text{(B23)}
\end{aligned}
$$

$$
\begin{aligned}
\langle\sigma_i h_j\rangle_{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})} &= \sum_{\vec{\sigma},\vec{h}} O(\vec{\sigma},\vec{h},\vec{a},\vec{b})\sigma_i h_j \\
&= \sum_{\vec{\sigma}} O_1(\vec{\sigma},\vec{a})\sigma_i \sum_{\vec{h}} h_j O_2(\vec{h},\vec{b}) \qquad \because (Eq.B19\ \ O(\vec{\sigma},\vec{h},\vec{a},\vec{b}) = O_1(\vec{\sigma},\vec{a})O_2(\vec{h},\vec{b}) \\
&= \sum_{\vec{\sigma}} \prod_m O_1(\sigma_m, a_m)\sigma_i \sum_{\vec{h}} h_j \prod_p O_2(h_p, b_p) \qquad (\because \text{no intralayer connections as RBM}) \\
&= \sum_{\sigma_i\in\{1,-1\}} O_1(\sigma_i, a_i)\sigma_i \sum_{h_j\in\{1,-1\}} h_j O_2(h_j, b_j) \\
&= \left(\frac{e^{a_i/k} - e^{-a_i/k}}{e^{a_i/k} + e^{-a_i/k}}\right)\left(\frac{e^{b_j/k} - e^{-b_j/k}}{e^{b_j/k} + e^{-b_j/k}}\right) \\
&= \tanh(a_i/k)\tanh(b_j/k) \qquad \text{(B24)}
\end{aligned}
$$

where we have used $\langle ... \rangle_{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})}$ to denote an average over the distribution $O(\vec{\sigma},\vec{h},\vec{a},\vec{b})$ defined in Eq. B19. Using Eq. B24 in Eq. B23 we thus get

$$
\begin{aligned}
P_{success} &= P(\vec{1}_a) \\
&\geq \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij}\langle\sigma_i h_j\rangle_{O(\vec{\sigma},\vec{h},\vec{a},\vec{b})})}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \\
\boxed{P_{success} \geq \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij}\tanh(a_i/k)\tanh(b_j/k)}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}}} &\qquad \because Eq.B24 \qquad \text{(B25)}
\end{aligned}
$$

The above lower bound in Eq. B25 is a generic lower bound independent of any occurrence of random variables $(\vec{\sigma},\vec{h})$ and only dependant on the parameters of the network $(\vec{a},\vec{b},\vec{W},k)$.

In the plot below, we simulate the performance of the lower bound deduced in Eq. B25 against the actual probability using **'RBM-qasm'** . In the plot, we designate the R.H.S of Eq. B25 as P_{lb} signifying lower bound and the actual probability of the event (L.H.S of Eq. B25) as $P_{success}$ as used before.

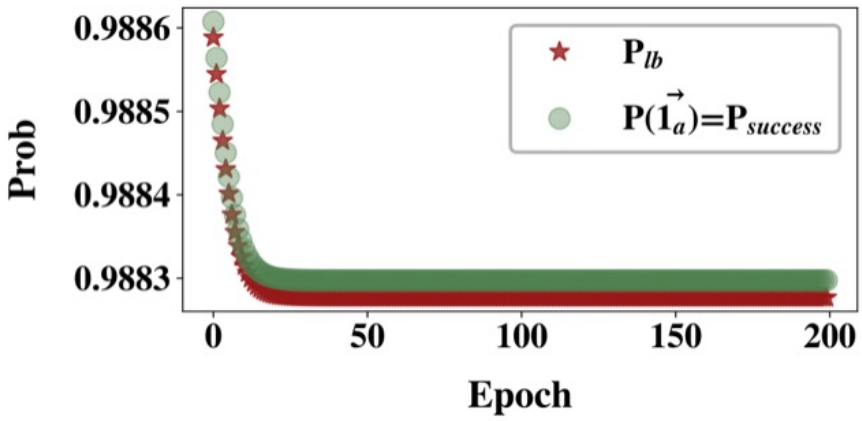

Figure B1. The lower bound for probability of successfully collapsing the ancilla register in state $|\vec{1_a}\rangle$ as deduced in Eq. B25 (R.H.S of Eq. B25) is plotted in red and the actual probability of such an event (Eq. B22 or L.H.S of Eq. B25) is plotted in green as a function of training iteration. We see that the green curve is always slightly above the red one in accordance with Eq. B25 deduced above. The simulation is from the **'RBM-qasm'** variant for conduction band (CB) of MoS_2 at $(k_x, k_y) = K$ symmetry point which is where the direct band-gap is lowest. The simulation is performed by warm starting with initial parameter set from a converged run at a nearby k-point to ensure faster and accurate convergence. With the said warm start the desired accuracy of $\leq 10^{-4}$ eV in energy error was reached within 200 iterations. The high value of the probability of the successful event (and the associated lower bound) as seen from the y-scale is problem-specific as it entirely depends on $(\vec{a}, \vec{b}, \vec{W}, k)$ (see Eq. B22 and Eq. B25) . For these systems, even with a moderate k parameter i.e. $k \leq 1.5$ for all iterations, the remaining set $(\vec{a}, \vec{b}, \vec{W})$ are such that a high value of $P_{success}$ as seen on the y-scale is attained. The parameter set $(\vec{a}, \vec{b}, \vec{W})$ depends on the updates from the cost-function and hence on the Hamiltonian of the system being treated. For this choice of k parameter, the specific values of the y-scale in the plot is thus characteristic of the systems being studied in this report and may be different for other systems. However the lower bound deduced in Eq. B25 is mathematically generic and should be valid for any arbitrary system and a given k parameter, even though the specific value it acquires during training may vary

From this lower bound, all pre-existing known bounds can be recovered as we shall see next.

Limiting cases

- $\tanh(a_i/k) \to \pm 1, \tanh(b_j/k) \to \pm 1$

In this case we get from Eq. B25

$$
\begin{aligned}
P_{success} &= P(\vec{1}_a) \\
&\geq \frac{e^{\frac{1}{k}(\sum_{i,j} w_{ij} \tanh(a_i/k) \tanh(b_j/k)}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \\
&\geq \frac{e^{-\frac{1}{k}|(\sum_{i,j} w_{ij} \tanh(a_i/k) \tanh(b_j/k)|}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \qquad \because e^{\alpha x} \geq e^{-\alpha|x|} \ \ \forall x \in \mathbb{R} \ , \alpha \geq 0) \\
&\geq \frac{e^{-\frac{1}{k}(\sum_{i,j}|w_{ij} \tanh(a_i/k) \tanh(b_j/k)|}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \qquad \because \text{Triangle Inequality} \\
&\geq \frac{e^{-\frac{1}{k}(\sum_{i,j}|w_{ij}||\tanh(a_i/k)||\tanh(b_j/k)|}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \\
&\geq \frac{e^{-\frac{1}{k}(\sum_{i,j}|w_{ij}|)}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \qquad \because |\tanh(a_i/k)| = |\tanh(b_j/k)| = 1 \\
&\geq \frac{1}{e^{\frac{2}{k}(\sum_{ij}|w_{ij}|)}}
\end{aligned}
\tag{B26}
$$

Thus choosing $k = \max(\sum_{ij}\frac{|w_{ij}|}{2}, 1)$, the lower bound attained in Eq. B26 for the probability for successful sampling becomes a constant value of e^{-4}. This bound was deduced in [4] in a completely different manner. Here we derived a master bound from which this is recovered.

- $\underline{\tanh(a_i/k) \to (a_i/k), \tanh(b_j/k) \to (b_j/k)}$

  Using similar kind of reasoning as in the previous case one can show

$$
\begin{aligned}
P_{success} &= P(\vec{1}_a) \\
&\geq \frac{e^{-\frac{1}{k}(\sum_{i,j}|w_{ij}||\tanh(a_i/k)||\tanh(b_j/k)|}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}} \\
&\geq \frac{e^{-\frac{1}{k}(\sum_{i,j}|w_{ij}||(a_i/k)||(b_j/k)|}}{e^{\frac{1}{k}\sum_{i,j}|w_{ij}|}}
\end{aligned}
$$

Here one cannot do much to proceed unless some assumptions are made .We assume that $|a_{\rm i}| \leq a_0 \ \forall$ i and $|b_{\rm j}| \leq b_0 \ \forall$ j. Then we get

$$\begin{aligned} P_{success} &= P(\vec{1}_a) \\ &\geq \frac{e^{-\frac{1}{k}(\sum_{\rm i,j}|w_{\rm ij}||(a_{\rm i}/k)||(b_{\rm j}/k)|}}{e^{\frac{1}{k}\sum_{\rm i,j}|w_{\rm ij}|}} \\ &\geq \frac{e^{-\frac{a_0 b_0}{k^3}(\sum_{\rm i,j}|w_{\rm ij}|}}{e^{\frac{1}{k}\sum_{\rm i,j}|w_{\rm ij}|}} \\ &\geq \frac{1}{e^{(\frac{1}{k}+\frac{a_0 b_0}{k^3})(\sum_{\rm i,j}|w_{\rm ij}|)}} \end{aligned} \tag{B27}$$

One can then numerically choose k to make the bound in this limit in Eq. B27 a constant value greater than a user-defined real number.

B.5 Hamiltonian for MoS_2 and WS_2

For the Hamiltonian matrix used in this report for MoS_2 and WS_2, we use the 3-band third-nearest neighbor tight-binding description as adopted in Ref.[5]. The Hamiltonian matrix elements are provided below for brevity and completeness.

The Hamiltonian matrix ($\hat{H}$) being a 3×3 description is written as

$$\hat{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \tag{B28}$$

Since the Hamiltonian is hermitian ($\hat{H} = \hat{H}^\dagger$), the only unique elements are the upper triangular block. Each such element is described below. For each of the elements we use the symbol $1.0\text{i} = \sqrt{-1}$ to denote the imaginary components. Also a_0 is the lattice constant which is 3.190 Å for MoS_2 and 3.191 Å for WS_2 [5]

$$H_{11} = \epsilon_1 + 2t_0(2\cos(\frac{k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2}) + \cos(k_x a_0)) + 2r_0(2\cos(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2}) + \cos(\sqrt{3}k_y a_0))$$
$$+ 2u_0(2\cos(k_x a_0)\cos(\sqrt{3}k_y a_0) + \cos(2k_x a_0))$$

$$H_{12} = -2\sqrt{3}t_2\sin(\frac{k_x a_0}{2})\sin(\frac{\sqrt{3}k_y a_0}{2}) + 2(r_1 + r_2)\sin(\frac{3k_x a_0}{2})\sin(\frac{\sqrt{3}k_y a_0}{2})$$
$$- 2\sqrt{3}u_2\sin(k_x a_0)\sin(\sqrt{3}k_y a_0) + 1.0\mathrm{i} * 2t_1\sin(\frac{k_x a_0}{2})(2\cos(\frac{k_x a_0}{2}) + \cos(\frac{\sqrt{3}k_y a_0}{2}))$$
$$+ 2(r_1 - r_2)(\sin(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2})) + 2u_1\sin(k_x a_0)(2\cos(k_x a_0) + \cos(\sqrt{3}k_y a_0))$$

$$H_{13} = 2t_2(\cos(k_x a_0) - \cos(\frac{\sqrt{3}k_y a_0}{2})\cos(\frac{k_x a_0}{2})) - \frac{2}{\sqrt{3}}(r_1 + r_2)(\cos(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2}) - \cos(\sqrt{3}k_y a_0))$$
$$+ 2u_2(\cos(2k_x a_0) - \cos(k_x a_0)\cos(\sqrt{3}k_y a_0)) + 1.0\mathrm{i} * 2\sqrt{3}t_1\cos(\frac{k_x a_0}{2})\sin(\frac{\sqrt{3}k_y a_0}{2})$$
$$+ \frac{2}{\sqrt{3}}(r_1 - r_2)\sin(\frac{\sqrt{3}k_y a_0}{2})(\cos(\frac{3k_x a_0}{2}) + 2\cos(\frac{\sqrt{3}k_y a_0}{2})) + 2\sqrt{3}u_1\cos(k_x a_0)\sin(\sqrt{3}k_y a_0)$$

$$H_{22} = \epsilon_2 + (t_{11} + 3t_{22})\cos(\frac{k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2})) + 2t_{11}\cos(k_x a_0) + 4r_{11}\cos(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2})$$
$$+ 2(r_{11} + \sqrt{3}r_{12})\cos(\sqrt{3}k_y a_0) + (u_{11} + 3u_{22})\cos(k_x a_0)\cos(\sqrt{3}k_y a_0) + 2u_{11}\cos(2k_x a_0)$$

$$H_{23} = \sqrt{3}(t_{22} - t_{11})\sin(k_x a_0)\sin(\sqrt{3}k_y a_0) + 4r_{12}\sin(\frac{3k_x a_0}{2})\sin(\frac{\sqrt{3}k_y a_0}{2})$$
$$+ \sqrt{3}(u_{22} - u_{11})\sin(k_x a_0)\sin(\sqrt{3}k_y a_0) + 1.0\mathrm{i} * 4t_{12}\sin(\frac{k_x a_0}{2})(\cos(\frac{k_x a_0}{2}) - \cos(\frac{\sqrt{3}k_y a_0}{2}))$$
$$+ 4u_{12}\sin(k_x a_0)(\cos(k_x a_0) - \cos(\sqrt{3}k_y a_0))$$

$$H_{33} = \epsilon_2 + (3t_{11} + t_{22})\cos(\frac{k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2}) + 2t_{22}\cos(k_x a_0) + 2r_{11}(2\cos(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2})$$
$$+ \cos(\sqrt{3}k_y a_0)) + \frac{2}{\sqrt{3}}r_{12}(4\cos(\frac{3k_x a_0}{2})\cos(\frac{\sqrt{3}k_y a_0}{2}) - \cos(\sqrt{3}k_y a_0))$$
$$+ (3u_{11} + u_{22})\cos(k_x a_0)\cos(\sqrt{3}k_y a_0) + 2u_{22}\cos(2k_x a_0) \quad \text{(B29)}$$

The energy parameter set[5] for both the systems MoS_2 and WS_2 is tabulated below

Parameter List for three-band model from GGA calculations		
Parameter(eV)	MoS_2	WS_2
ϵ_1	0.683	0.717
ϵ_2	1.707	1.916
t_0	-0.146	-0.152
t_1	-0.114	-0.097
t_2	0.506	0.590
t_{11}	0.085	0.047
t_{12}	0.162	0.178
t_{22}	0.073	0.016
r_0	0.060	0.069
r_1	-0.236	-0.261
r_2	0.067	0.107
r_{11}	0.016	-0.003
r_{12}	0.087	0.109
u_0	-0.038	-0.054
u_1	0.046	0.045
u_2	0.001	0.002
u_{11}	0.266	0.325
u_{12}	-0.176	-0.206
u_{22}	-0.150	-0.163

B.6 Importance of Measurement Error Mitigation

In this section we simulate the performance of the algorithm using the **'RBM-IBMQ'** variant with and without the use of Measurement Error Mitigation(MEM) [6]. In Fig. B2(a) we plot the final energy error after the training process for four 4 arbitrarily chosen (k_x, k_y) points on the conduction band (CB) of MoS_2. These points are not coincident with the symmetry points of the system as symmetry points converges better regardless. We also plot in Fig. B2(b)-(e) the change in the energy error during the training iterations/epoch for each of the four points marked as (1),(2),(3),(4). These points are marked on the x-axis in Fig. B2(a). From each of the plots in Fig. B2(b)-(e) we see that the the red curve (with MEM) converges smoothly whereas the green dots (without MEM) displays noisy oscillations leading to poor self-convergence. To account for this statistical uncertainty, the last 30 points from each of the curves Fig. B2(b)-(e) is time averaged and the results constitutes the points in Fig. B2(a). The error bars on each points for the results without MEM are the sample standard deviation of these 30 points and the points themselves are sample mean. For the results with MEM since the convergence is smooth the corresponding error bars over the last 30 iterations are an order of magnitude lesser than the sample mean and hence not displayed. All of the simulations are performed in IBM-Sydney in a single run and followed till 150 iterations. Each of the 4 points are warm-started with parameter set from nearby k-points in the **'RBM-qasm'** variant to hasten convergence . Also in each case, the operator $\hat{O} = |v_0\rangle\langle v_0|$ is the corresponding ground state from **'RBM-qasm'** variant. This together with the fact that the same initial parameter set is used for both the runs with and without MEM eliminates errors due to faulty initialization and erroneous construction of the operator $\hat{O}$ (due to ground state infidelity) and focuses only on the errors introduced in the algorithm in the presence and absence of MEM . We see that for some of the points in the main manuscript the energy errors in the CB for MoS_2 in the **'RBM-IBMQ'** variant were higher due to infidelity in the corresponding ground state which adversely affects the operator $\hat{O}$. Thus the usage of MEM, appropriate warm starting and the fact that the probability of successfully sampling of the quantum circuit for these systems is naturally very favorable (see section B.7) explains the superior quality of the results on the actual IBM hardware for

all systems studied in this report. Among other factors, quality of results on the hardware can also be affected by sparsity of the H matrix for a given architecture of the network with (n, m) even though that point has not been investigated much in this report. Also it is generally assumed that hybrid variational algorithms like ours are resilient to certain noises[7] which may be playing a role as well.

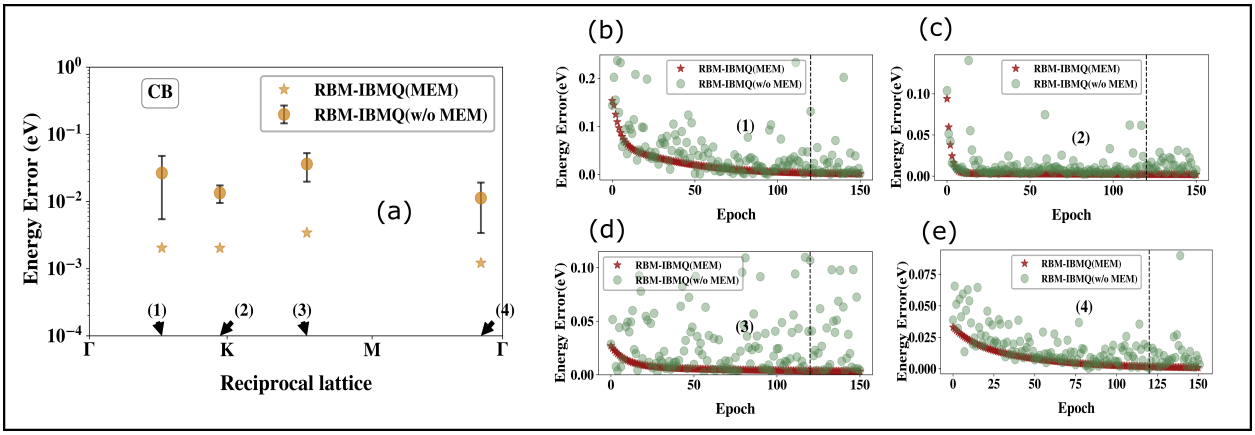

Figure B2. The energy error (eV) for 4 arbitrarily chosen (k_x, k_y) points within the Brillouin zone in the $\Gamma - K - M - \Gamma$ path of the conduction band (CB) of MoS_2 after training the network using the **'RBM-IBMQ'** variant with and without Measurement Error Mitigation (MEM). Each of the 4 points is marked on the x-axis as (1),(2),(3),(4). We also plot the energy error as a function of the training epoch/iteration with and without MEM in (b) for point index (1) in (c) for point index (2) (d) for point index (3) (e) for point index (4). We see from (a) that the results with MEM are of higher accuracy for all 4 points than without MEM. However by far the greatest impact which MEM has on the results is on improving self-convergence. This is best seen from (b)-(d). The error bars in (a) on the points without MEM are to highlight the statistical uncertainty due to time averaging from this poor self-convergence. Each such bar designates the sample standard deviation of the last 30 points (marked in (b)-(d) with a vertical dashed line) in the training process whereas the orange circles in (a) are the corresponding sample mean. Each calculation with and without MEM is done using a single run on IBM-Sydney and followed till 150 iterations. All simulations are performed by warm starting with initial parameter set from a converged run at a nearby k-point in **'RBM-qasm'** variant. This is done so that same initial parameter set is used for simulations with and without MEM which eliminates biases due to random initial parameterization and affords a strictly fair comparison. The 4 points chosen are not the symmetry points as we have seen that symmetry points usually converges better regardless

B.7 Measurement statistics

For all the systems studied in the manuscript, while training the network using the quantum circuit we use 10^6 measurement shots for the **'RBM-qasm'** variant. For the

'RBM-IBMQ' variant (IBM-Sydney and IBM-Toronto) we use 8192 measurement shots which happens to be the maximum allowed value. We show in this section why the said number of measurement counts are adequate for all the systems we study in this report.

We simulate the statistics of the measurement using **'RBM-qasm'** in Fig. B3 wherein the probability of successfully collapsing the ancilla register in $|\vec{1_a}\rangle$ is plotted both in the ideal case from Eq. B22 (see Fig. B3(a)) and from the frequency distribution of the actual quantum measurement (see Fig. B3(b)). The total number of measurement count used is 10^6. We see that for these systems, the favorable event is nearly exclusive indicating that in only less than 2% of the measurements the ancilla register collapsed into the unwanted state $|\vec{0_a}\rangle$. The latter events are discarded when constructing the RBM distribution from the measurement statistics (see Eq. B18). As a result nearly 9.8×10^5 samples are available to faithfully construct the target distribution. Each such sample constitutes a $(\vec{\sigma}, \vec{h})$ pair. The frequency distribution of such samples is the simulated RBM distribution (P_{meas}) as obtained from the quantum circuit. The fact that this distribution agrees with the actual one (P_{RBM}) is quantified using the KL divergence between the two which is defined as

$$KL\ div = \langle -\log(\frac{\mathrm{P_{meas}}}{\mathrm{P_{RBM}}})\rangle_{\mathrm{P_{RBM}}} \tag{B30}$$

KL divergence can only be zero if the two distributions P_{meas} and P_{RBM} exactly agree. This is seen to be the case in Fig. B3(c) for all iterations during the training process. This indicates that with the 10^6 total measurement shots , the quantum circuit faithfully produces the RBM distribution for the systems studied in the report.

We see from Fig. B3 (d)-(f) that the distribution can be constructed with appreciable accuracy with 10^4 measurement shots too. This situation simulates the measurements on **'RBM-IBMQ'** (without the noise) wherein only 8192 shots are maximally allowed. This is a consequence of the fact that the probability of successful sampling (see Fig. B3(a) and (d) is high and hence practically all measurements yields the favorable outcome. Even though from Fig. B3(e) and Fig. B3(f) one can see that the deviations from the expected value has increased than in B3(b) or B3(c) due to lesser number of shots yet such deviations are still too small to be of practicable consequences. For example, the final error in energy at the end

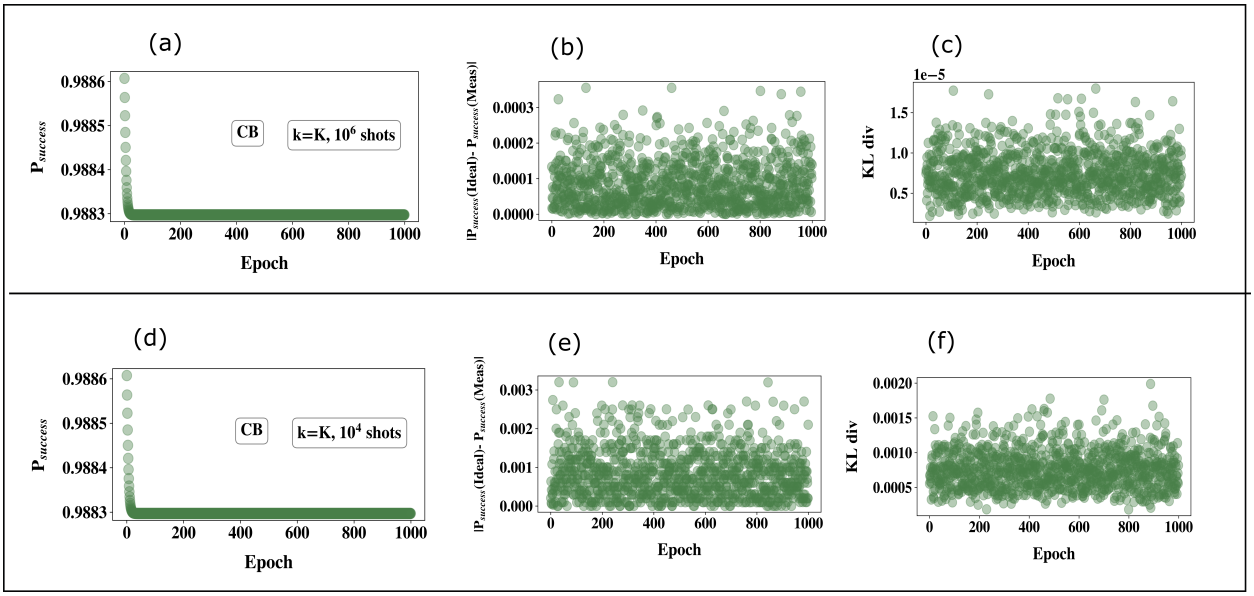

Figure B3. The ideal probability of successfully collapsing the ancilla register in state $|\vec{1_a}\rangle$ as computed from Eq. B22 (b) The difference between a) and the fraction of the total number of times the ancilla register collapsed in state $|\vec{1_a}\rangle$ as obtained from the direct measurement statistics in the quantum circuit. This quantity is procured by counting the number of times such an event happened while measuring all the qubits and dividing the count with the total number of measurement shots used (10^6 in this case). We see this value deviates only marginally from the ideal value in a) indicating that the desired event is extremely favorable. (c) The KL divergence of the distribution constructed from post-selecting all bit-strings $(\vec{\sigma}, \vec{h})$ for the visible and hidden-node qubits after the ancilla register collapsed in state $|\vec{1_a}\rangle$ and the exact RBM distribution. We see that the KL divergence is extremely close to zero indicating that the circuit can correctly learn the RBM distribution with the designated number of shots. This is because in most of them the favorable outcome of ancilla register collapsing to $|\vec{1_a}\rangle$ happens naturally (see (a) and (b)) for the systems being studied in this report even with moderate k-parameter ($k \leq 1.5$) (see text for discussion). (d)-(e)-(f) are similar plots as (a)-(b)-(c) but with 10^4 total measurement shots. All the results are simulated in the **'RBM-qasm'** variant for conduction band (CB) of MoS_2 at $(k_x, k_y) = K$ symmetry point which is where the direct band-gap is lowest. The simulation is performed by warm starting with initial parameter set from a converged run at a nearby k-point.

of the training in both 10^6 and 10^4 shots is $\leq 10^{-4}$ eV. Thus we conclude for reasonably large number of measurements (10^6 or 10^4) one can construct the distribution faithfully for the systems studied in this report with moderate k parameter. Fig B3(d)-(f) simulates only the ideal case for the hardware data. Of course for the **'RBM-IBMQ'** variant, the distribution during the training is further corrupted due to gate infidelities and qubit-decoherence error which is manifested in the higher errors in the final result for **'RBM-IBMQ'** variant as compared to the **'RBM-qasm'** variant in all the systems studied in the main manuscript.

It must be emphasized that the probability of sampling and collapsing the ancilla register in the desired state $|\vec{1_a}\rangle$ is entirely a function of the parameter set $(\vec{a}, \vec{b}, \vec{W}, k)$ encountered during the training process (see Eq. B22, Eq. B25). For these systems, the k-parameter during the course of the training is always found to be moderate $k \leq 1.5$. Even with such a low k, the remaining set $(\vec{a}, \vec{b}, \vec{W})$ is such that a high value of $P_{success}$ is naturally obtained. The parameter set $(\vec{a}, \vec{b}, \vec{W})$ is updated using the cost function which makes the values it acquires dependant on the Hamiltonian $\hat{H}$ and symmetry operator $\hat{O}$ of the specific system being treated. Thus the specific values of the successful sampling probability $P_{success}$ in Fig. B3(a) and Fig. B3(d) is somewhat characteristic of the system. This makes formulating a general expression for the total number of measurements required in our algorithm to successfully construct the target distribution (with a chosen error) difficult as even for a given architecture of the network (visible layer n and hidden layer m) and a certain k parameter, the parameter set $(\vec{a}, \vec{b}, \vec{W})$ may change for different systems which may in turn alter the probability of successful sampling thereby necessitating different number of total required measurements. But in systems wherein the the parameter set $(\vec{a}, \vec{b}, \vec{W})$ do not lead to favorable value of $P_{success}$ with low k , k-parameter in our model will have to be tuned by the user adaptively to a higher value to make the lower bound for the probability of successful sampling (see Eq. B25) greater than a chosen preset thereby guaranteeing that at each iteration a good subset of these measurements are always fruitful with which the distribution can be constructed.

B.8 Variation of the results with changing hidden node density

In this section we simulate the effect of changing the hidden node density $\alpha = \frac{m}{n}$ on the results for the systems treated in this report. We take MoS_2 as a prototypical example and train the network using the **'RBM-qasm'** and **'RBM-cl'** variant for the $(k_x, k_y) = K$ point for both the valence band (VB) and conduction band (CB). The number of visible node neurons n as discussed for this system is 2. We vary the number of hidden neurons $m = [2, 3, 4]^T$ which corresponds to a changing density $\alpha = [1, 1.5, 2]^T$ respectively. Varying the hidden node density changes the circuit depth/gate-requirements and the number of

parameters used for training the network. We see for these systems, the results do not change much as all energy errors are below 10^{-3} eV and hence way below the threshold of chemical accuracy. However for correlated systems one may need to make the RBM ansatz more expressive by enhancing the hidden node density[8]. We shall explore this point again for a molecular example in section B.12

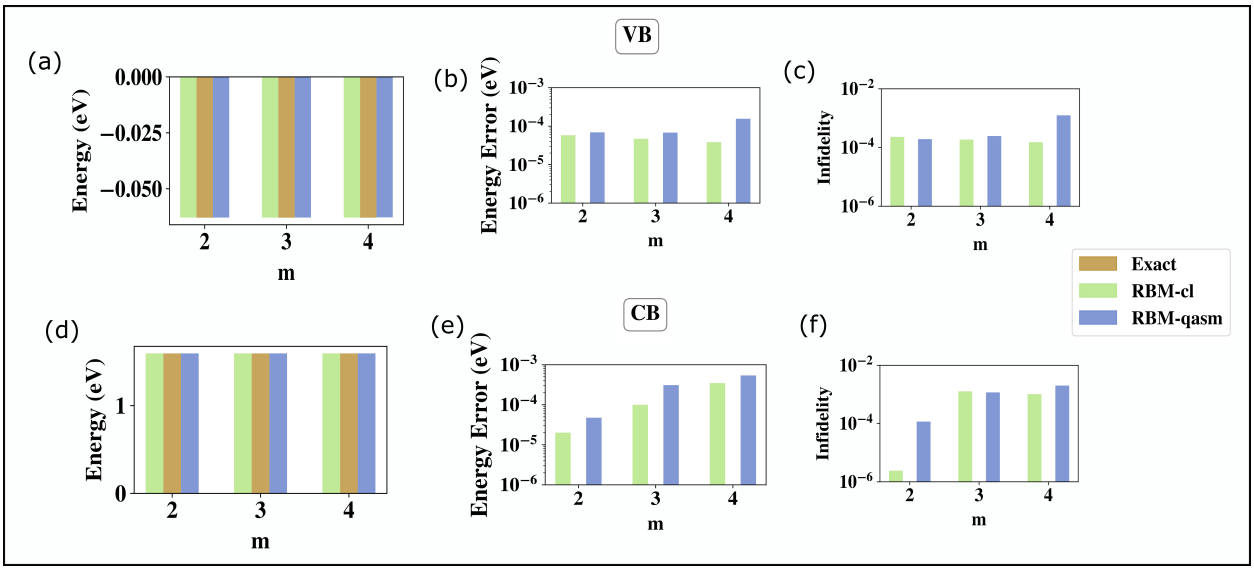

Figure B4. (a) The energy of the valence band (VB) at $(k_x, k_y) = K$ point for MoS_2 for the exact case and the two flavors of RBM namely **'RBM-qasm'** and **'RBM-cl'** are plotted by changing the number of neurons m in the hidden node. (b) The corresponding energy errors from the calculations in a) (c) The corresponding state infidelities from the calculations in a) (d) Similar result as in a) but for the conduction band (CB) at $(k_x, k_y) = K$ point for MoS_2. (e) The corresponding energy errors from the calculations in d) (f) The corresponding state infidelities from the calculations in d)

B.9 Spin-Orbit Coupling (SOC) data for WS_2

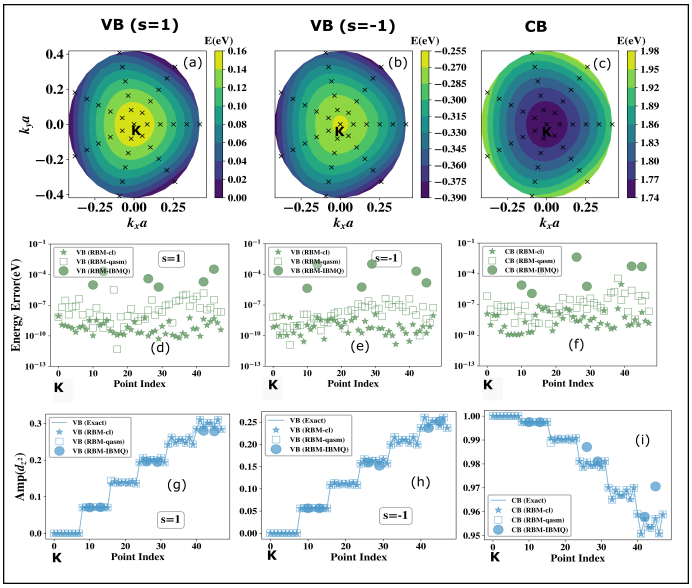

Figure B5. (a) The exact energy contours in valence band (s=1) within the three-band approximation for the Hamiltonian (see Eq. 8 in main manuscript) as a function of (k_x, k_y) near the K-point in WS_2 (b) Same as in a) but for s=-1 (c) Same as in a) for the conduction band. The crosses in (a), (b) and (c) denotes the (k_x, k_y) pair wherein calculations for all three flavors of RBM have been executed. (d) Energy errors in eV from three flavors of RBM calculations for points denoted as cross in a) for the valence band (s=1) case computed using $\lambda = 0$ in Eq. B1 in WS_2. The x-axis is a flattened point index with (k_x, k_y) pairs marked as crosses in (a) mapped to integers such that the origin is at the K-point. From the K-point, the flattened point index scale moves spirally outwards grouping all (k_x, k_y) pairs satisfying $|k| = \sqrt{k_x^2 + k_y^2}$ as consecutive integers and then proceeding to the next $|k|$ (e) Same as in d) but with points denoted in b) as crosses for other valence band with s=-1.(f) Same as in d) but for points denoted in c) as crosses for the conduction band computed with $(\lambda = 5, \omega = 0, \hat{O} = |\nu_0\rangle\langle\nu_0|)$ in Eq. B1 (g) The amplitude for the occupancy of d_{z^2} orbital on the metal for states computed at (k_x, k_y) pairs near the K-point from all three flavors of RBM as well as the exact states in valence band (s=1) for WS_2. The amplitude of states with the same $|k| = \sqrt{k_x^2 + k_y^2}$ appear bunched together as 'steps' due to flattened point-index scale used. Near the K-point the amplitude is the same for all such pairs within a given step due to isotropy of the energy surface. However away from the K-point deviations appear due to trigonal warping owing to the D_{3h} symmetry of the unit cells in TMDCs. The states from all flavors of RBM can resolve the influence of warping accurately with the performance worsened for the noisy variant.(h) Same as in g) for valence band (s=-1) (i) Same as in g) for conduction band. For all these calculations the warping parameters are kept the same as that for MoS_2 even through the band energies are obtained within the three-band approximation calculated using RBM for WS_2 in the main manuscript

B.10 Eigenvectors of L_z and L^2 operator for MoS_2 and WS_2

The L_z operator at $K-$ point in the three-band basis of $(d_{z^2}, d_{xy}, d_{x^2-y^2})$ orbitals of the metal centre is given as [5]

$$\hat{L_z} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 2\mathrm{i} \\ 0 & -2\mathrm{i} & 0 \end{bmatrix} \tag{B31}$$

where $\mathrm{i} = \sqrt{-1}$. The eigenvectors and eigenvalues of the L_z and L^2 matrix is given as

Eigenvectors and Eigenvalues of L_z/L^2 operator			
Eigenvalue (L_z)	Eigenvalue (L^2)	Eigenvector	Band Index
-2	4	$\frac{1}{\sqrt{2}}[0, -1, -\mathrm{i}]^T$	CB+1
0	0	$[1, 0, 0]^T$	CB
2	4	$\frac{1}{\sqrt{2}}[0, 1, -\mathrm{i}]^T$	VB

Note that the basis is $d_{z^2} = [1, 0, 0]^T$, $d_{xy} = [0, 1, 0]^T$, $d_{x^2-y^2} = [0, 0, 1]^T$ and the notation VB= valence band (ground state), CB = conduction band (1st excited state) and CB + 1 = Higher energy band (2nd excited state above conduction band). As mentioned in [5], in the chosen basis, the matrix elements of L_x operator and L_y operator are all zeros and hence L^2 operator enjoys exclusive contribution from L_z operator given above. As a result, the eigenvectors of L^2 operator are the same as given in the table above but the eigenvalue pair (-2, 2) of L_z maps to the same eigenspace of L^2 with eigenvalue =4. In other words L^2 has a doubly-degenrate eigenspace of eigenvalue =4 made from eigenvectors VB and CB+1 (see

Table above) whereas a non-degenerate eigenspace of eigenvalue =0 with the eigenvector CB (see Table above).

B.11 Transferability of the Learning to Other systems

In this section we simulate the possibility of using our algorithm which has trained the network for one system and see if it is possible to 'transfer' the learning to get converged results in a different but closely related system. For unsupervised classical deep learning algorithms once the parameters of the neural network is tuned so that the probability distribution of the visible node mimics the unknown distribution of the training data closely, any new sample drawn from the visible node will be representative of a sample generated from the target distribution. No further training is necessary indicating that such models are highly transferable. However for quantum data as has been treated in this report, the meaning of transferability needs to be clarified. In our algorithm, the objective is to train the network to mimic the amplitude and phase distribution of an eigenstate of a given Hamiltonian H. Even if we use a fully trained network from some system yet some further amount of training would be necessary for a similar system as the Hamiltonian matrix $\hat{H}$ and the symmetry operator $\hat{O}$ are changing and hence the eigenstates of the new $\hat{H}$ are also slightly different than in the previous system. In that sense, the learning of our algorithm for quantum data is only partially transferable. That being said, a fully trained for a similar system does help and can enhance the rate of convergence and reduce the number of iterations for further re-training. We simulate this possibility using **'RBM-qasm'** and **'RBM-cl'** variant for WSe_2 using the trained network for MoS_2. Due to the symmetry partitioning of the metal orbitals as guaranteed in [5], to treat WSe_2 too one would need $n = 2$ qubits. We use $m = 2$ as for other systems as well. The gate requirements is exactly the same as for MoS_2 and WS_2. We see that in both the variants starting with a trained network for MoS_2 converged results can be obtained using just 1000-2000 iterations. All the results are displayed in Fig. B6.

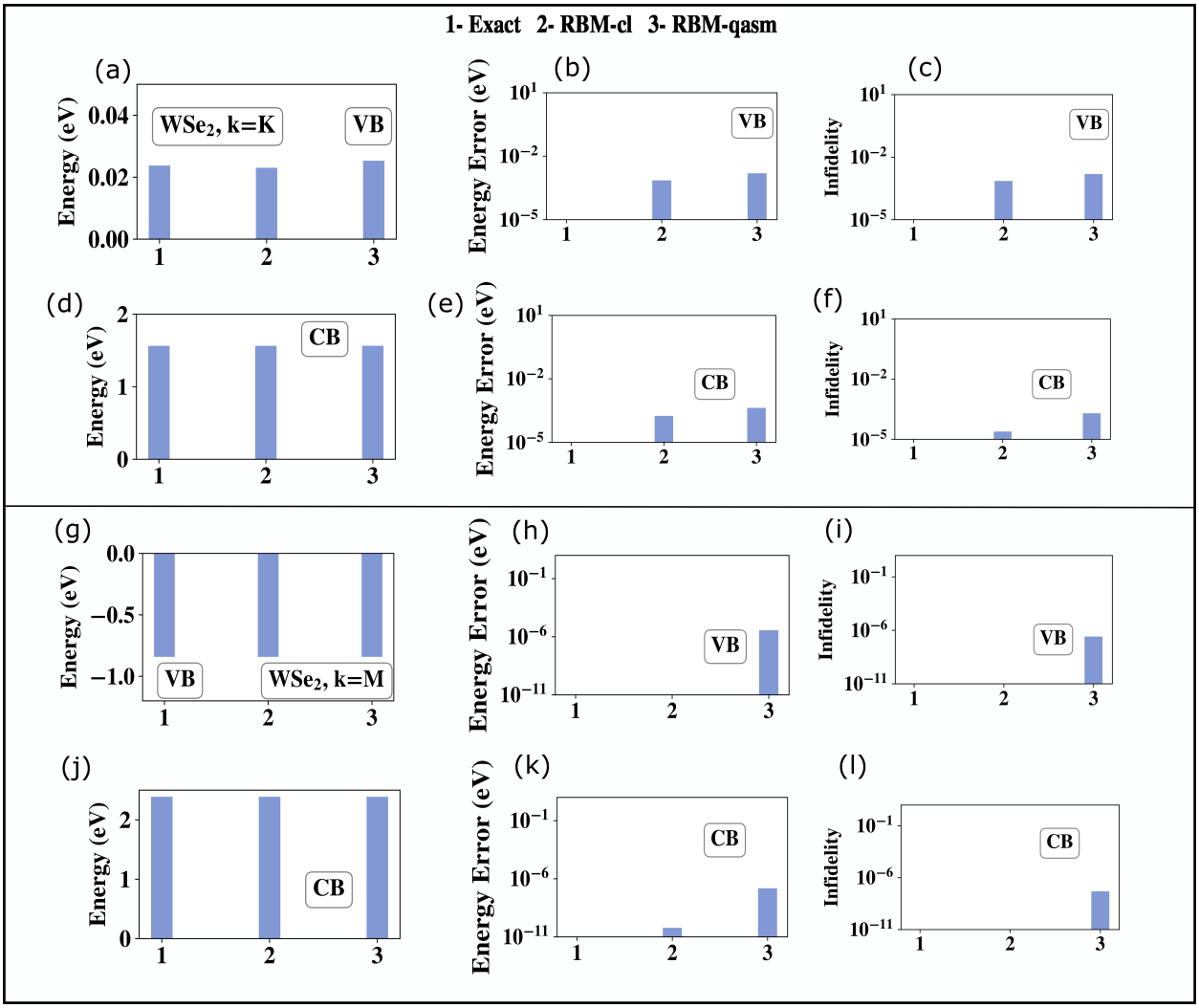

Figure B6. (a) The energy of the valence band (VB) at $(k_x, k_y) = K$ point for WSe_2 in the three-band model [5] for the exact case and the two flavors of RBM namely **'RBM-qasm'** and **'RBM-cl'**. The calculations are done by starting from a network trained with the converged results for the VB of MoS_2 at the K point. (b) The corresponding energy errors from the calculations in a) (c) The corresponding state infidelities from the calculations in a) (d) Similar result as in a) but for the conduction band (CB) at $(k_x, k_y) = K$ point for WSe_2. The calculations are done by starting from a network trained with the converged results for the CB of MoS_2 at the K point (e) The corresponding energy errors from the calculations in d) (f) The corresponding state infidelities from the calculations in d). (g)-(l) are results for WSe_2 similar to (a)-(f) but at a different symmetry point i.e. $(k_x, k_y) = M$ point. The calculations in this case are done by starting from a network trained with the converged results for the VB/CB of MoS_2 at the M point

B.12 LiH- A molecular example

In this section we benchmark the performance of our algorithm using a molecular example LiH in STO-6G basis set. The said basis generates 6 molecular orbitals (MOs) for LiH out of which the lower lying two MOs (HOMO and HOMO-1) are completely filled each with 2 electrons corresponding to the two spin orbitals. We freeze the core (HOMO-1) and use the HOMO and the LUMO in the active space[9]. The LUMO+1, LUMO+2, LUMO+3 are a part of the virtual orbital set and are always unoccupied. The Hamiltonian matrix so constructed is a 16×16 matrix . We therefore need $n = 4$ neurons in the visible node and 4 qubits in the quantum circuit for these visible neurons. For the hidden neurons we have used $m = 4$ as well as $m = 6$ to show how the results change by varying hidden node density. The total number of qubits in the quantum circuit for the case $m = 4$ is then $4+4+16 = 24$ where 16 ancillary qubits (mn) are used for mediating the interacting terms in the RBM distribution. For $m = 6$ we consequently need 34 qubits. The total number of gates for the case of $m = 4$ is thus 8 single qubit R_y gates and 16 $C-C-R_y$ gates. For the case of $m = 6$ the corresponding numbers are 10 single qubit R_y gates and 24 $C - C - R_y$ gates.

We simulate the ground and excited state potential energy surface as a function of stretching the Li and H bond length. The results are displayed in Fig. B7 and in Fig. B8 along with the corresponding errors from the exact diagonalization results. We use the **'RBM-qasm'** and **'RBM-cl'** variant for all benchmarking. While most runs on the **'RBM-cl'** variant are randomly initialized near the equilibrium bond length warm starting has been used extensively, especially near the dissociation limit where multi-reference correlation is important. For **'RBM-qasm'** variant the runs are sometimes warm started with the initial parameter set of a nearby bond length in the **'RBM-cl'** case. We see that in all cases away from the dissociation limit, errors are fairly low in the range of $10^{-5} - 10^{-3}$ eV whereas it is $\leqslant 0.0022$ a.u. near the terminal bond lengths r_{LiH} studied in the plots ($0.7\,\text{Å} \leq r_{\text{LiH}} \leq 2.8\,\text{Å}$)

In [9] (see Fig. 4(b) and Fig. 4(d), the ground state potential energy surface for LiH is discussed with and without the effect of warm-starting (which the reference called 'Transfer Learning'). Fig. 4(d) with warm-starting shows all points to have energy errors much lesser than in Fig. 4(b)) which is also found to be the case for the calculations in this report. The key differences between the results in Fig. B7 and in [9] needs to be emphasized at this point. [9] simulated the results at the **'RBM-cl'** variant only as the circuit was not

directly implementable on a quantum simulator (like Qiskit's *qasm_simulator* as used in this report) or a NISQ device. Also [9] used STO-3G as the basis set and number of hidden nodes $m = 8$ unlike here. Besides the phase node included one neuron in [9]. We see that for the ground state all errors in energy were ≤ 0.001 a.u. in [9] which is likely due to the higher number of hidden nodes $m = 8$ used which makes the network more expressive. We also changed the hidden node density and simulated the ground and the excited state for LiH in Fig. B8 with $m = 6$ hidden neurons apart from $m = 4$. One must note that only **'RBM-cl'** variant is used in this case akin to [9]. This is because to simulate the system at all bond lengths using $(n = 4, m = 6)$ neurons one would need 34 qubits in the quantum circuit which is beyond the current standards of *qasm_simulator* in Qiskit Aer backend. We see the errors in the ground state calculation are usually lower in this case akin to [9] but for excited states the trend is less clear. Possibly to lower the energy errors further for excited states one needs higher hidden node density than what is considered here. Although not used for Fig. B7 or Fig. B8, optimizers like ADAM, RMSProp etc which are known to recover a neural network from locally trapped minima may also help.

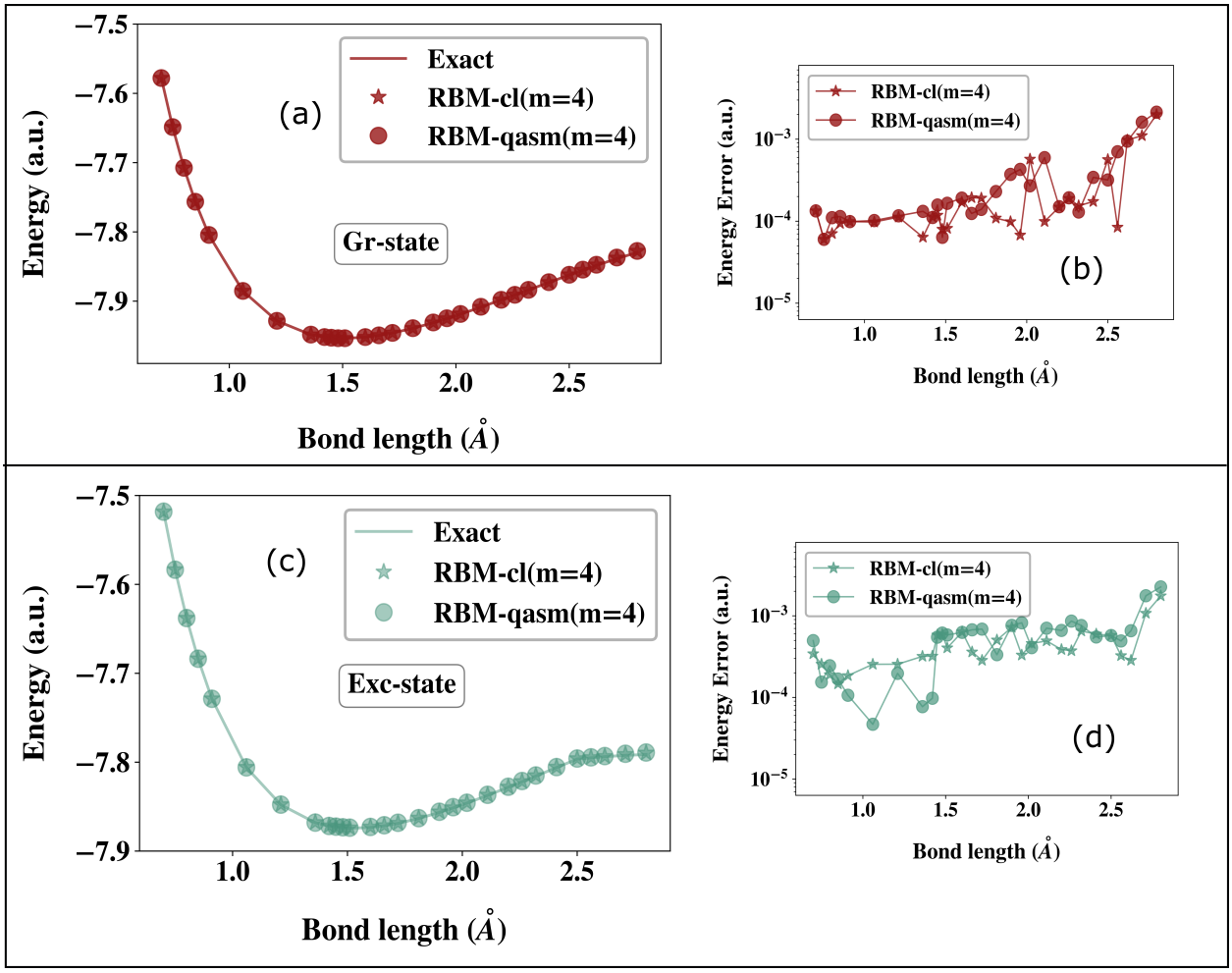

Figure B7. The dissociation curve for the ground state of LiH in **'RBM-cl'** and **'RBM-qasm'** variant overlayed against the exact value. (b) The error in energies from a) from the exact value. c)The dissociation curve for the excited state of LiH in **'RBM-cl'** and **'RBM-qasm'** variant overlayed against the exact value. (d) The error in energies from c) from the exact value. In all of the results in this panel we use $n = 4$ and $m = 4$

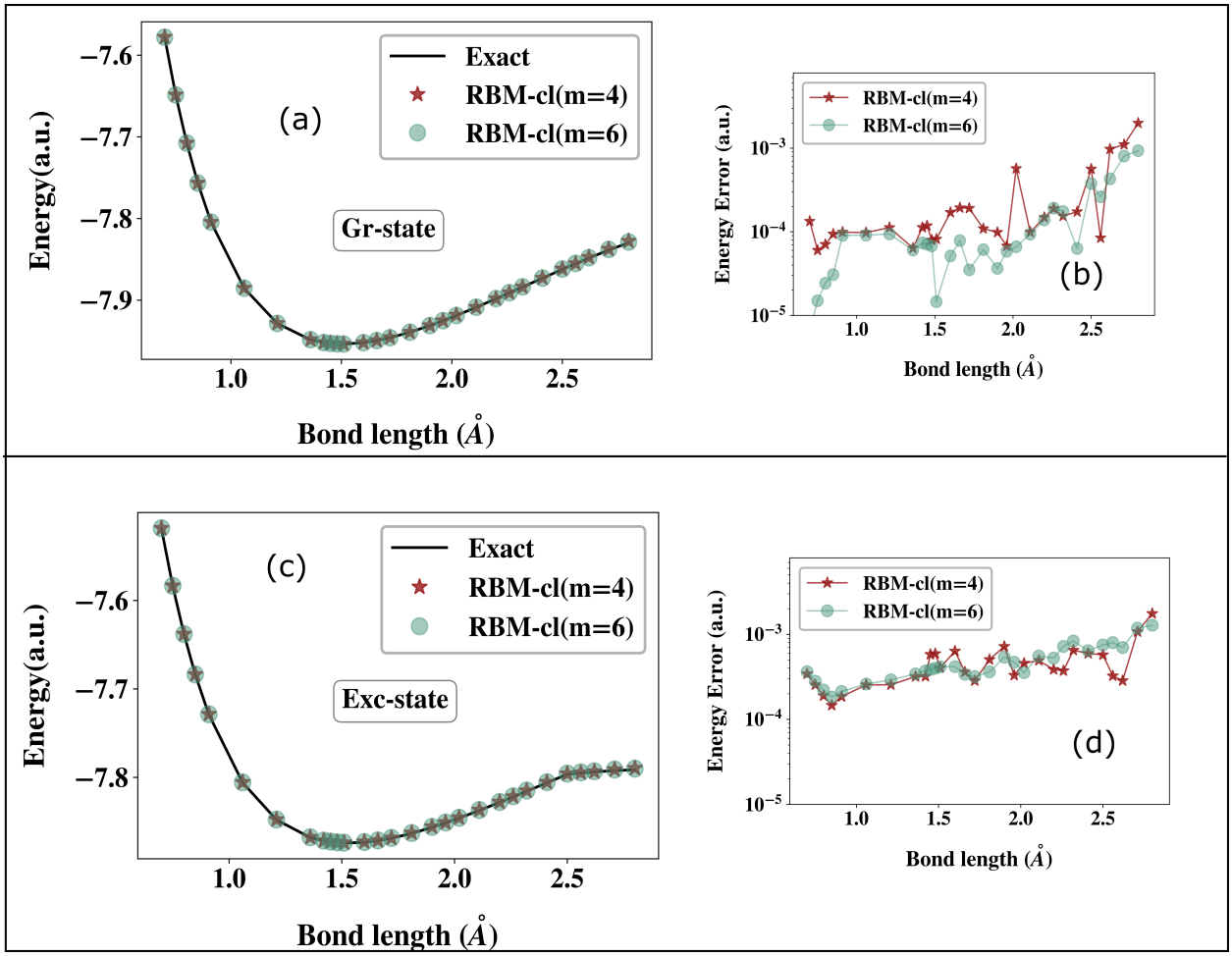

Figure B8. The dissociation curve for the ground state of LiH in **'RBM-cl'** variant overlayed against the exact value for two different m. (b) The error in energies from a) from the exact value for both m. c)The dissociation curve for the excited state of LiH in **'RBM-cl'** variant overlayed against the exact value for two different m. (d) The error in energies from c) from the exact value for both m. In all of the results in this panel we use $n = 4$ and $m = 4$ and compare it with $n = 4$ and $m = 6$

C FINITE-SIZE SCALING ON A DIGITAL QUANTUM SIMULATOR [APPENDIX]

C.1 Bulirsch-Stoer Algorithm

For $h_N = 1/N$ where $N = 0, 1, 2, \ldots$, the Bulirsch-Stoer algorithm can be used to find the limit of a function $T(h_N)$ as $N \to \infty$[10], [11]. For demonstration, consider that we only have $T(h_N)$ for $N = 0, 1, 2, 3$, then the following rows are computed successively,

$n=0$	$T_0^{(0)}$	$T_0^{(1)}$	$T_0^{(2)}$	$T_0^{(3)}$
$n=1$	$T_1^{(0)}$	$T_1^{(1)}$	$T_1^{(2)}$	
$n=2$	$T_2^{(0)}$	$T_2^{(1)}$		
$n=3$	$T_3^{(0)}$			

using the following rules

$$T_{-1}^{(N)} = 0 \tag{C1}$$

$$T_0^{(N)} = T(h_N) \tag{C2}$$

$$T_{m\geq 1}^{(N)} = T_{m-1}^{(N+1)} + (T_{m-1}^{(N+1)} - T_{m-1}^{(N)}) \left[\left(\frac{h_N}{h_{N+m}} \right)^{\omega} \left(1 - \frac{T_{m-1}^{(N+1)} - T_{m-1}^{(N)}}{T_{m-1}^{(N+1)} - T_{m-2}^{(N+1)}} \right) - 1 \right]^{-1} \tag{C3}$$

where ω is a free parameter determined by minimizing $\varepsilon_m^{(\mathrm{i})} = \left| T_m^{(\mathrm{i}+1)} - T_m^{(\mathrm{i})} \right|$. The final answer is $T_3^{(0)}$.

References

[1] R. Kundu, "Tight-binding parameters for graphene," *Modern Physics Letters B*, vol. 25, no. 03, pp. 163–173, 2011.

[2] G. Dresselhaus, M. S. Dresselhaus, and R. Saito, *Physical properties of carbon nanotubes*. World scientific, 1998.

[3] S. Axler, *Linear Algebra Done Right*, 3rd ed. Springer, Cham, 2015.

[4] R. Xia and S. Kais, "Quantum machine learning for electronic structure calculations," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[5] G.-B. Liu, W.-Y. Shan, Y. Yao, W. Yao, and D. Xiao, "Three-band tight-binding model for monolayers of group-vib transition metal dichalcogenides," *Physical Review B*, vol. 88, no. 8, p. 085 433, 2013.

[6] G. S. Barron and C. J. Wood, "Measurement error mitigation for variational quantum algorithms," *arXiv preprint arXiv:2010.08520*, 2020.

[7] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023 023, 2016.

[8] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, "Restricted Boltzmann machines in quantum physics," *Nature Physics*, vol. 15, no. 9, pp. 887–892, 2019, ISSN: 17452481. DOI: 10.1038/s41567-019-0545-1. [Online]. Available: http://dx.doi.org/10.1038/s41567-019-0545-1.

[9] R. Xia and S. Kais, "Quantum machine learning for electronic structure calculations," *Nature Communications*, vol. 9, no. 1, p. 4195, 2018, ISSN: 2041-1723. DOI: 10.1038/s41467-018-06598-z. [Online]. Available: https://doi.org/10.1038/s41467-018-06598-z.

[10] R. Bulirsch and J. Stoer, "Numerical treatment of ordinary differential equations by extrapolation methods," *Numer. Math.*, vol. 8, pp. 1–13, 1966. [Online]. Available: https://doi.org/10.1007/BF02165234.

[11] M. Henkel and G. Schutz, "Finite-lattice extrapolation algorithms," vol. 21, no. 11, pp. 2617–2633, Jun. 1988. DOI: 10.1088/0305-4470/21/11/019. [Online]. Available: https://doi.org/10.1088/0305-4470/21/11/019.