Assessing the Trainability of the Variational Quantum State Diagonalization Algorithm at Scale

by

Joan Étude Arrow

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization (Quantum Information)

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Quantum algorithm development is a famously difficult problem. The lack of intuition concerning the quantum realm makes constructing quantum algorithms which solve particular problems of interest difficult. In addition, modern hardware limitations place strong restrictions on the types of algorithms which can be implemented in noisy circuits.

These challenges have produced several solutions to the problem of quantum algorithm development in the modern Near-term Intermediate Scale Quantum (NISQ) Era. One of the most prominent of these is the use of classical machine learning to discover novel quantum algorithms by minimizing a cost function associated with the particular application of interest.

This quantum-classical hybrid approach, also called Variational Quantum Algorithms (VQAs) has emerged as a major interest for both academic and industrial research due to its flexible framework and existing applications in both optimization and quantum chemistry.

What is still unclear, is whether these algorithms will work at scale in the noisy training environment of the NISQ era. This is mainly due to the phenomenon of exponentially vanishing training gradients, commonly referred to as the Barren Plateaus problem, which prevents training of the classical machine learning model.

Recent results have shown that some types of cost functions used in training result in Barren Plateaus, while others do not. This cost function dependence of barren plateaus has implications for the entire field of VQAs which appear to be relatively unexplored thus far.

In this thesis I revisit a 2018 paper my collaborators and I published, which established a new Variational Quantum State Diagonalization (VQSD) algorithm, and demonstrate that this algorithm's cost function will encounter a Barren Plateau at scale. I then introduce a simple modification to this cost function which preserves the function of VQSD while also ensuring trainability at scale.

Acknowledgements

This thesis very nearly did not happen had it not been for the support of a number of exceptional people.

First, I would like to thank the researchers who supported my intellectual development in quantum computing. I would like to thank Julia Plavnik, who provided a young researcher her first opportunity to conduct research in the field of quantum computing and without which I would not have met my current adviser; Jon Yard, who has always had my back and who provided me a place here at Waterloo to explore and find my own way in this new field. I am thankful for both of my group-mates Sam and Connor, who were some of my first friends in a new country and who are always available to shoot the shit.

I want to give a special thanks to my VQSD coauthors from the Los Alamos Quantum Computing Summer School: Ryan LaRose, Arkin Tikku, Lukasz Cincio & Patrick Coles. Your primary contributions creating the bulk of VQSD have laid the intellectual foundations for this thesis. I hope that my analysis of your cost function serves to further our collective work. It has been a privilege to revisit VQSD and finally add something original of my own to our algorithm. Thank you for making room in your summer school for a freshly graduated baccalaureate who was simply trying to keep up.

I would finally like to thank everyone who provided me care and support in the wake of my bipolar diagnosis and struggles with gender identity over the past three years since coming to Waterloo.

Notably among these are the team at 1st Step CMHA, your early intervention made all the difference. I also want to thank queer IQC community members Al Sachs, Carie Earnest, and Simon Daley who provided couches to crash on when I was homeless, and shoulders to cry on when I was feeling hopeless. Your constant love, encouragement, and reminders that I was not alone and that my illness was not all I am can be truly said to have saved my life.

A special thank you is owed to my wonderful partner Gavynn McKay, who cared for me during my most recent battle with mania for three consecutive weeks as well as all my smaller depressive episodes. Gavynn's love has made it possible for me to almost fully return to pre-diagnosis levels of executive function and happiness.

Many of the above mentioned individuals also deserve a special thank you for encouraging me to love and accept my true self as an autistic non-binary transgender woman. It was only after resolving this crucial piece of the puzzle that I was able to begin working on this thesis using the mental space freed up by no longer denying who I've always been.

I came to see this part of myself at the same time I learned the basic building block of quantum computing transcends the binary.

This thesis is very much a product of that transgender energy.

Dedication

I dedicate this thesis to the next generation. May they far surpass us.

Table of Contents

List of Figures

Chapter 0

Mathematical Preliminaries

Here we provide the key definitions and terminology which are assumed as background in the main document.

Note that I do not claim ownership over any of the following well-known ideas. I simply present them here for the reader's convenience.

0.1 Complex Vector Spaces and Useful Identities

Throughout this work, we are dealing with quantum states which represent physical, natural systems at their smallest scale (photons, electrons, atoms, etc.). The mathematical foundation upon which all of this is described assumed familiarity with complex numbers

$$z = \alpha + \beta i = e^{-i\theta} \in \mathbb{C}, \qquad \text{where } \theta \in [0, 2\pi] \text{ and } i = \sqrt{-1}$$

The role of complex numbers in quantum is well beyond the scope of this thesis, but one feature that imaginary 'phases' provide is the ability of quantum states to undergo interference, which is a core feature of how quantum information differs from its classical counterpart.

A vectorspace over such a field of complex numbers is called a **Hilbert Space** and is denoted $\mathcal{H}_N$ where N is the dimension of the space. Vectors in this Hilbert space representing quantum states can transform into new states by multiplying the appropriate matrix/linear transformation M.

Some more miscellaneous definitions and identities from these areas which we do not assume are known are as follows:

1. Exponential of an operator A

$$e^{-iA} = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

2. Tensor product over addition

$$M \otimes (A + B) = M \otimes A + M \otimes B$$

0.2 Hello $\langle Q|uantum\rangle$

This is a basic introduction to quantum mechanics, the notation we use throughout the thesis, as well as the core concepts and principles which are essential to following sections.

0.2.1 $\langle Bra|Ket\rangle$ Notation

First we start with bra/ket notation. Fundamentally, this notation enables us to represent column vectors $|\rangle$, row vectors $\langle|$ and the interactions between them.

We can show how a more traditional column vector notation can be equivalently described as a quantum state

$$\vec{v} \equiv |v\rangle$$

row vectors can be similarly described using $\langle v|$. This notation becomes visually useful when we take inner products of two states $|\psi\rangle$ and $|\phi\rangle$ which we denote $\langle\psi|\phi\rangle \in \mathbb{C}$.

We can use this notation to denote the essential probabilities of various outcomes represented by $|\psi\rangle$ - we illustrate this with the following example of a quantum bit or qubit:

We define the qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle \in \mathcal{H}_2$ as the smallest possible piece of quantum information. We can represent this in vector form by identifying

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

then

$$|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}.$$

0.2.2 Baby Statistics

The state $|\psi\rangle$ encodes a probability distribution over the outcomes $|0\rangle$ and $|1\rangle$. We can think of $|\psi\rangle$ as representing the probability of two mutually exclusive (orthogonal) outcomes, such as a coin coming up heads $|0\rangle$ or tails $|1\rangle$. We represent the probability of heads with the following equation

$$Pr(0) = |\langle 0|\psi\rangle|^2 = |a|^2.$$

Similarly, the probability of measuring tails is $Pr(1) = |\langle 1|\psi\rangle|^2 = |b|^2$, here $|b|$ denotes the absolute norm of the complex number b, where the absolute norm is defined by

$$|z| = (\alpha + \beta i)(\alpha - \beta i) = \alpha^2 + \beta^2.$$

Given the probabilistic nature of quantum, our objective in quantum computing may either be to produce a particular measurement outcome (such as heads), or to encode the numerical output of our computation by storing it in the probability (e.g $|a|^2$) of a particular outcome, or in the average of the data we chose to observe in (or extract from) the quantum state $|\psi\rangle$.

To see how we may calculate average (or expectation) values of data extracted from the quantum state $|\psi\rangle$ - consider a concrete example:

Consider for a moment an experiment where we prepare an electron in a laboratory by putting the electron in a box where the electron is exposed to an energy field (Hamiltonian) which we represent as the matrix $\mathbf{H}$. For the purposes of this example, imagine that the energy of the electron is in a superposition of three possible measurement outcomes $|\psi\rangle = c_1|E_1\rangle + c_2|E_2\rangle + c_3|E_3\rangle$. This is simply a qubit with one extra level (a qutrit) where the measurement outcomes correspond to energy levels rather than heads or tails. The energy outcome $|E_i\rangle$ is an eigenvector of the Hamiltonian $\mathbf{H}$ and the energy E_i we measure is its eigenvalue.

If we measure the energy of the electron we will likely get a different answer E_i each time according to the relative size of the probabilistic weights $|c_1|^2, |c_2|^2, |c_3|^2$. If we want to learn more, we may prepare the same experiment many times and make multiple measurements to collect data which would allow us to approximate these weights. In this context, we may define the average energy in terms of the state by applying the Hamiltonian $\mathbf{H}$ to the state $|\psi\rangle$ and projecting the result back onto $|\psi\rangle$ via

$$\begin{aligned}\langle \mathbf{H} \rangle &\equiv \langle \psi | \mathbf{H} | \psi \rangle \\ &= (c_1^\dagger \langle E_1| + c_2^\dagger \langle E_2| + c_3^\dagger \langle E_3|)(c_1 \mathbf{H} |E_1\rangle + c_2 \mathbf{H} |E_2\rangle + c_3 \mathbf{H} |E_3\rangle) \\ &= (c_1^\dagger \langle E_1| + c_2^\dagger \langle E_2| + c_3^\dagger \langle E_3|)(c_1 E_1 |E_1\rangle + c_2 E_2 |E_2\rangle + c_3 E_3 |E_3\rangle) \\ &= E_1 |c_1|^2 + E_2 |c_2|^2 + E_3 |c_3|^2.\end{aligned}$$

Using this formula, we can then input our approximations of $|c_1|^2, |c_2|^2, |c_3|^2$ to calculate the average energy $\langle \mathbf{H} \rangle$.

0.2.3 Normalization

Now we are ready to state a core axiom of quantum. If the state $|\psi\rangle$ can be understood as containing the relative probabilities of different measurement outcomes - from a statistical perspective we know that the sum of the probabilities of each possible outcome must be one. We call this **normalization**. Revisiting our qubit $|\psi\rangle = a|0\rangle + b|1\rangle$ from before, normalization implies $Pr(0) + Pr(1) = |a|^2 + |b|^2 = 1$. In general, if $\{|0\rangle, |1\rangle, ..., |d-1\rangle\}$ is our orthonormal basis for $\mathcal{H}_d$ and we have the quantum state $|\psi\rangle = \sum_{i=0}^{d-1} c_i |i\rangle$ then normalization requires

$$\langle \psi | \psi \rangle = \sum_{i=0}^{d-1} |c_i|^2 = 1.$$

This normalization needs to hold even when our quantum state evolves in time

$$|\psi\rangle_i \xrightarrow{M} |\psi\rangle_f$$

and as a result we must constrain the types of valid matrices M which can evolve our state. We require that the state evolution matrix M be unitary, i.e.

$$MM^\dagger = M^\dagger M = I$$

Where I is the identity matrix. A corollary of this property is that, given quantum states $|\psi\rangle, |\phi\rangle$ applying the state transformation M to produce $M|\psi\rangle, M|\phi\rangle$ has no effect on their inner product

$$\langle\psi| M^\dagger M |\phi\rangle = \langle\psi|\phi\rangle .$$

We can see that this preserves normality when we consider the case where $|\phi\rangle = |\psi\rangle$ as no matter the evolution M applied, provided M is unitary it will always be the case that

$$\langle\psi| M^\dagger M |\psi\rangle = \langle\psi|\psi\rangle = 1.$$

0.3 Quantum Computers and Quantum Algorithms

Now that we have established the fundamental notation inherent to quantum, we are ready to introduce the concept of a quantum computer as well as the broader subject of this thesis: quantum algorithms.

A quantum computer can be defined as a device containing a number of packets of quantum information. Often and for the purposes of this thesis, those packets of quantum information are the two-level subsystems we introduced in the previous section: qubits.

In general though, quantum computers can be built from subsystems with any number of levels. If all components of the quantum computer have d levels, then we call these components qu**d**its and the computer is referred to as a qudit quantum computer. The state of an arbitrary qudit was given briefly in the previous section as

$$|\psi\rangle = \sum_{i=0}^{d-1} c_i |i\rangle .$$

Whether we store information in qubits or qudits, this alone does not grant our device the status of quantum computer. Rather quantum computers are characterized by their ability to both store **and** manupulate quantum information via some given quantum algorithm.

0.3.1 Building Quantum Algorithms with Quantum Gates

In order to understand the way we will discuss quantum algorithms in future sections, here I briefly outline the 'atomic structure' of quantum algorithms in terms of a few much simpler quantum operations.

Each quantum computer may source it's qubits or qudits from a different technology. The details of these technologies is far beyond the scope of this thesis, but suffice to say that each different kind of quantum computer will interact with quantum information differently. These differences translate to the way each computer can compile, or 'articulate' quantum algorithms via its own 'native alphabet' $\mathcal{A}$.

These native gate alphabets are sets which contain more fundamental quantum operations called **quantum gates**. It is for this reason we will use the terms alphabet and gate set interchangeably.

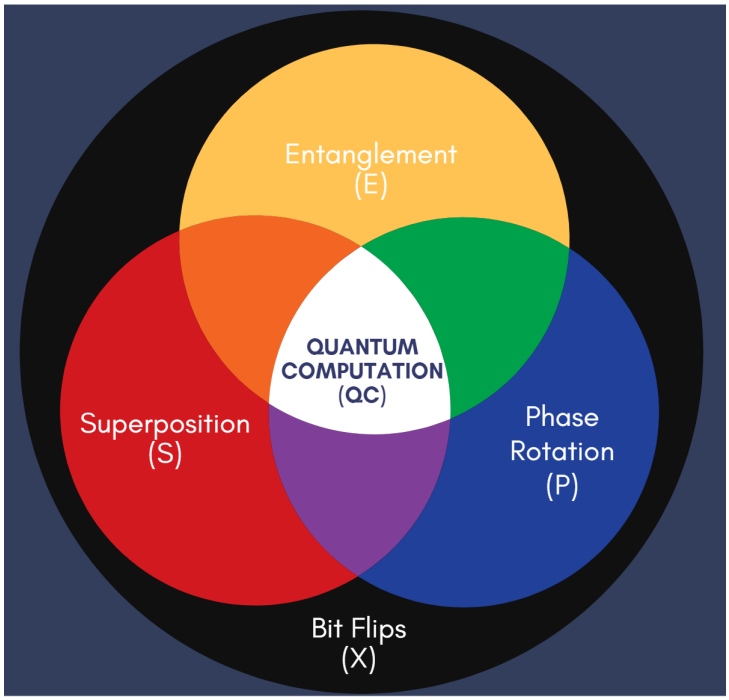


Figure 1: Any fully capable, or universal, gate alphabet must be able to perform all three fundamental quantum operations *Entangelent, Phase rotation, and Superposition.*

What are these fundamental gates? We will introduce an example of a particular

native gate alphabet by expanding on the fundamental opration underlying all classical computing.

Classical computers make use of logical circuits (NOT, OR, AND) to construct basic arithmetic circuits and the more complex programs and algorithms for which they are now known. These fundamental operations act on classical binary $\{0, 1\}$ and can be represented as extensions of a single circuit flip, or NOT operation controlled on zero other bits (NOT), one other bit (OR), and two other bits (AND). You can see this principle in practice and try your hand at building classical circuits here [1].

This NOT gate, which we will denote X, is the first gate which we will add to our alphabet $\mathcal{A}$. We define the action of X via it's effect on the basis vectors $|0\rangle, |1\rangle \in \mathcal{H}_2$

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

If we recall the vector form of our basis, then X takes on the matrix form

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Next we add the uniquely quantum gates which make quantum computation possible. While different gate sets vary in their details, all universal gate sets contain three fundamental quantum operations: Superposition (H), Entaglement (CX), and Phase Rotations (T) 1. An example of such operations for universal quantum computation is the Clifford + T set:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/8} \end{bmatrix}, CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Using these, we can construct another important family of quantum gates called the Paulis:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, T^4 = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = iXZ = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Using these basic components and quantum states composed of tensor products of qubits $|\phi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2 \otimes ... \otimes |\psi\rangle_n$. We can endeavor to construct quantum algorithms by acting on individual qubits with any of the above 2×2 or pairs of qubits using the 4×4 entangling gate CX.

Chapter 1

Introduction

This thesis investigates the efficacy of using machine learning models to discover new quantum algorithms. These quantum algorithms are called Variational Quantum Algorithms (VQAs). While these represent an exciting and relatively new approach to quantum algorithm development, this field remains highly heuristic in nature and general results which guarantee the efficacy of these methods at scale are only beginning to emerge [2]. In particular, VQAs are vulnerable to exponentially vanishing cost function gradients which prevent training at scale. This phenomenon, also called the Barren Plateaus Problem, poses a major challenge to every VQA.

This thesis starts off with an introduction to the general problem of quantum algorithm development and to the near-term infeasibility of fault-tolerance, thereby setting the stage for the NISQ Era and VQAs. Chapter 1 discusses the fundamentals of VQAs and introduces the problem of barren plateaus along with recent results which guarantee trainability for VQAs with local cost functions. Chapter 2 applies these results to my own variational quantum algorithm for density matrix diagonalization, called VQSD. Finally, Chapter 3 discusses next steps: How I intend to apply this same analysis to the entire field of VQAs.

1.1 Why is it so Hard to Write Quantum Algorithms?

Developing new quantum algorithms is hard. While the concept of quantum computation was first proposed by Richard Feynman in the 1980s [3], we didn't see the first example of a quantum algorithm which demonstrated an exponential advantage solving a problem of major interest until Peter Shor published his factoring algorithm in the mid nineties [4].

In the years since the publication of his landmark factoring algorithm, the general challenge of developing quantum algorithms for a given problem of interest has frustrated even Shor [5]. In this paper, which Shor titled *Why haven't more quantum algorithms been found?*, the master of factoring laments that so few algorithms have been discovered. Aside from worrying that there may not be very many use cases where quantum computers demonstrate a speedup, Shor's main explanation for the challenge is our lack of **quantum intuition**.

That is, intuition about how the unique properties of the quantum world can be leveraged to produce algorithms with advantage over their classical counterparts. Specifically, we need to develop intuition about the role that uniquely quantum phenomena, like interference, entanglement, and superposition play in speeding up computation.

The counter-intuitive nature of the quantum realm has made it difficult for the first generation of researchers since Shor to design new quantum algorithms. Many algorithms have been found since Shor's algorithm, but it still remains a very high bar to clear if a person wants to start writing quantum algorithms.

At this stage, it is important to contextualize that quantum computing is still very young as a discipline and that this problem need not be permanent. The roughly 100 years since the development of quantum mechanics in the first Quantum Revolution has been characterized by its own experts struggling to accept quantum's *"Spooky action at a distance"* in the case of Einstein [6], and the development of a research culture insisting that students *"Shut up and calculate"* rather than spend time contemplating central questions to which the experts of the field still have no good answers. Quantum works, even if nobody intuitively understands it.

In contrast to quantum mechanics, quantum computing provides a fundamentally new opportunity for new generations to advance beyond stale old debates and toward radical new understanding of the quantum world. This opportunity comes from the fact that quantum computing enables us to study the core features of quantum in a new context - that of information and computation rather than position and mechanics.

This optimism must be tempered by the reality that thus far, general quantum intuition for quantum algorithm development still largely eludes researchers. Quantum algorithm development becomes significantly more challenging when the noise of real-world implementations is considered.

Development of the first generation of quantum algorithms which followed Shor was conducted under the assumption that noise could be neglected provided robust error correction, with algorithms implemented on so-called 'fault tolerant' devices. More modern quantum algorithms will require a clear-headed assessment of the feasibility of this fault

tolerance and strategies for error mitigation in the absence of comprehensive error correction.

1.2 The in-feasibility of fault tolerance

Fault tolerant quantum computation is enabled by what are called 'error correction' techniques. Although is it beyond the scope of this thesis to dive into the field of error correction, we introduce the term in order to discuss the feasibility of building the long-awaited fault tolerant quantum computer. A quantum computer which achieves fault tolerance has done so by implementing error correction at a sufficient scale that we are able to perform quantum gate operations effectively noiselessly.

The idea behind error correction is to create a noise-free *logical* qubit[1] using many physical qubits subject to noise which are able to successfully correct errors to create the noise-free environment in which the logical qubit lives. This has the unfortunate side effect of exploding the number of physical qubits required to run the computer, with the specific ratio of noisy to logical qubits depending heavily on the quality of the device (error rate) and type of error correction used. In order for error correction to lead to fault tolerance, the error rates of the physical hardware needs to be below a set threshold determined by the error correcting code in question. The resource analysis which follows will focus on how long the algorithm takes to run and the number of qubits or the size of the device needed.

How feasible then is fault tolerance? Let us briefly examine the resources, both in device size and calculation time, to perform the two most famous quantum algorithms: Shor's factoring algorithm [4] and Grover's search [7].

First we consider Shor's quantum factoring algorithm.

How practical is factoring? In 2013 Devitt et al set out to answer this question [8]. It is important to point out that any answer to this question will be entirely dependent on our assumptions about device quality and what type of error correction we are employing - for the assumptions relevant to this study please see [8]. These researchers applied Shor's algorithm to its primary application: breaking RSA encryption - which is the basis of modern information security online. Breaking RSA requires factoring a $\sim$1024-bit number. In this study, it was determined that, after accounting for all realistic factors which arise when implementing Shor's algorithm fault-tolerantly, this calculation will require 2.3 *years*

[1] For all basic definitions see 0

and 1.9 *billion physical qubits* to complete. Although we note that this figure has recently been improved on, now requiring only 8 hours and 20 million noisy qubits [9] - this final metric still presents a serious challenge for modern engineers as the largest devices of today have ~ 100.

Next we turn to Grover's search.

In 2016, Amy et al. performed a cost analysis on the resources required to use Grover on attacking cryptographic schemes like SHA-256 and SHA3-256 [10]. It was found that the $\sqrt{n}$ query run time vastly underestimates the cost of implementing this algorithm. All fault tolerant quantum computations assume some error correction scheme, here researchers assumed a particular model called a surface code. It is here that the resource requirements begin to stack up.

Let's consider SHA-256. Researchers in [10] found that for this task Grover would require $2^{153.8} \approx 1.98 \cdot 10^{46}$ cycles to complete. These cycles are algorithm subroutines which clean up the noise to preserve fault tolerance and are repeated throughout the algorithm. In [11] a cycle time of $200ns/cycle$ was deemed reasonable for surface code computers. We may then calculate the run time of this algorithm as $2^{153.8} cycles \times 200 \cdot 10^{-9} s/cycle = 3.98 \cdot 10^{39} s = 1.26 \cdot 10^{32}$ years - keeping in mind the age of the universe is $13.8 \cdot 10^9$, we see this is beyond infeasible.

It is important to note that both of the above examples simply provide an estimate of resources with lots of room for improvement. Despite this, the estimated 1.9 billion qubits required for factoring, or the calculation time of $9 \cdot 10^{21}$ times the age of the universe to attack SHA-256 with Grover puts the feasibility of fault-tolerance in perspective - **fault-tolerance is just not feasible until we see many many orders of magnitude improvement on multiple fronts.**

We can't wait for fault tolerance. Instead, we look to prepare the next generation of quantum algorithms to suit those devices which can be reasonably built over the next few years which do not rely on error correction.

These are the constraints which give rise to the modern era of Noisy Intermediate Scale Quantum devices.

Welcome to the NISQ Era.

Chapter 2

Quantum Algorithms of the NISQ Era

So where does all this leave us? The famously hard problem of developing quantum algorithms becomes confined by the limitations of modern hardware. If we are to design *useful* quantum algorithms for the modern era, we are going to have to take a more realistic approach. If the number of qubits n is difficult to scale up, what is the smallest n that still gives us a quantum advantage over classical computers?

The answer to this question is very much a moving target, and will have changed in the time between when these words were written and when they were read. Quantum computers must compete alongside rapidly advancing supercomputers and the field of quantum computing is many decades younger than its classical counterpart. However, in 2018 Google made a claim to the first quantum computation which surpassed the capabilities of a modern supercomputer with a 52 qubit chip called Sycamore, which was the largest device of its time [12].

This value has already been greatly improved upon, but a realistic range of near-term values might be $50 - 150$ noisy qubits. This is derived from the values that John Preskill gave in 2018, right around the time when Sycamore was being built [13]. In his widely popular paper, Preskill introduced the term Noisy Intermediate-Scale Quantum (NISQ) to refer to our current phase of quantum technological development.

In this modern era, with our picture of $50 - 150$ noisy qubits in mind, we are ready to discuss a new paradigm for quantum algorithms which combines quantum circuits with classical optimization/machine learning.

2.1 Variational Quantum Algorithms

The fundamental recognition that modern quantum hardware is too noisy to handle long calculations gives rise to a very useful question. How might we exploit the availability of relatively error-free classical computers to assist quantum computation?

Enter Variational Quantum Algorithms (VQAs), also called Hybrid Quantum-Classical Algorithms. This class of NISQ algorithms employs a hybrid approach to solving an optimization problem. Broadly speaking, in these algorithms the quantum computer is fed a set of classical parameters specifying the quantum algorithm to perform. The output of this parametrized algorithm is then assessed via a predefined cost function such that the minimum of the cost function corresponds to the correct output. Until this minimum is found, the intermediate value of the cost function is fed into a classical optimizer, such as gradient descent, which updates the parameters closer to the optimal value and feeds those back into the quantum computer to try again. The quantum advantage of this method comes from using the quantum computer to evaluate cost functions which are infeasible/inefficient to compute on a classical computer. The appeal of VQAs is that, in principle, they minimize the complexity of the quantum computation by offloading as much of the computation to the classical computer as possible. A more detailed overview of VQAs can be found here [14].

We now turn our attention to two of the earliest and most famous VQAs which launched the field due to their near-term applications in quantum chemistry and optimization, respectively.

2.1.1 Variational Quantum Eigensolver (VQE)

When Richard Feynman laid out his idea of a quantum mechanical computer in 1986, an important question is *what could we use it for?* [3]. Feynman's reasoning was that a quantum computer should have a natural advantage in simulating quantum systems found in nature since nature itself is quantum.

In 2014, a NISQ algorithm was developed for this application using the hybrid paradigm we sketched out at the start of this chapter [15]. Where historically properties of molecules in quantum chemistry were determined by solving the Schrödinger equation, this approach becomes intractable for systems with more than 2-3 atoms due to exponential scaling of system size. Quantum Phase Estimation (QPE), a quantum algorithm developed in the pre-NISQ era of assumed fault-tolerance, can be used to calculate eigenvalues of eigenvec-

tors (including the ground energy eigenvalue) but as we have discussed, this approach is infeasible for the NISQ era.

This is the problem solved by the Variational Quantum Eigensolver (VQE). Where classical approaches all require exponential resources simply to represent the quantum states involved in computation, VQE is able to represent these using only linear resources by storing the quantum states naturally as qubits.

The goal of VQE is to calculate the ground state eigenenergy of a given Hamiltonian $\mathbf{H}$. As a quick reminder, the Hamiltonian of a system characterizes the energy landscape unique to that system. We represent the Hamiltonian as a sum of Paulis which we denote using the following indices $\mathcal{P} = \{I, X, Y, Z\} = \{\sigma_I, \sigma_X, \sigma_Y, \sigma_Z\} = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. We decompose the Hamiltonian $\mathbf{H}$ into a series of sums over increasing numbers of Paulis

$$\mathbf{H} = \sum_{i\alpha} h^i_\alpha \sigma^i_\alpha + \sum_{ij\alpha\beta} h^{ij}_{\alpha\beta} \sigma^i_\alpha \sigma^j_\beta + ...$$

where we sum over individual Paulis, pairs of Paulis, etc. Here the Roman indices $\{i, j, ...\}$ denote the qubit and the coefficients $h \in \mathbb{C}$. The elipses in the sum indicate additional terms involving sums with three, four, etc.

The cost function of VQE is obtained via an efficient means for calculating the expected value of the Hamiltonian. We have

$$\langle \mathbf{H} \rangle = \langle \lambda | \mathbf{H} | \lambda \rangle = \sum_{i\alpha} h^i_\alpha \langle \sigma^i_\alpha \rangle + \sum_{ij\alpha\beta} h^{ij}_{\alpha\beta} \langle \sigma^i_\alpha \sigma^j_\beta \rangle + ...$$

Each of these individual expectation values can then be directly calculated in parallel by measuring each qubit incurring a constant depth, a major reduction in complexity relative to QPE. Because expectation values require multiple samples to calculate precisely, this basic circuit will have to be repeated $O(|h_{max}|^2 Np)$ times in order to estimate individual expectation values with p precision and where N is the number of terms in the decomposition.

Assuming the Hamiltonian has only polynomially many terms in its Pauli representation, the above method is an efficient method for calculating expectation values. Although this does restrict the types of Hamiltonian which can be analyzed with VQE, authors of the original algorithm noted several Hamiltonians of this type which are of interest [15].

Of course, in order for gradient descent to work, a gradient has to be present. In 2018, a team of researchers at Google showed that for entire families of quantum circuits, the gradient vanishes exponentially with the number of qubits involved [16]. Research on circumventing this problem is ongoing with some important recent breakthroughs this year (2021). I will get to these a little later, but for the time being it suffices to be aware of this problem and to consider gradient-free optimization schemes.

This version of VQE provides a quantum advantage both in storage of quantum state and calculation of average energy given by $\langle \mathbf{H} \rangle$. By 2019, VQE had been subsequently improved to require fewer qubits [17] and fewer circuit samples [18].

Variational hybrid algorithms, of which VQE is an example, abound. These algorithms simplify the quantum algorithm development problem by requiring only the development of a quantum algorithm for a particular cost function. By plugging various cost functions into the classical optimization scheme many new applications, from optimization via the Quantum Approximate Optimization Algorithm (QAOA) [19] to matrix diagonalization via the Variational Quantum State Diagonalization algorithm (VQSD) [20], and beyond.

Let's look at the next most well-known of these applications: Optimization.

2.1.2 Quantum Approximate Optimization Algorithm

As we've already discussed, the primary difference between different VQAs is the cost function being optimized. In order to craft a cost function which allows for the solving of a given combinatorial optimization problem, we would like to incentivize the satisfying of constraints C_i. These constraints define the feasible values of typically a few of the n bits available in the computer and we define $C_i(x) = 1$ if n-bit string x satisfies constraint i and $C_i(x) = 0$ otherwise. We see then that

$$C(x) = \sum_i C_\alpha(x)$$

achieves maximum when x is a solution to all constraints in the combinatorial optimization problem. Now, if a bit string is found which satisfies these constraints we have a feasible string. Approximate optimization allows us to be content with a string $x*$ which gets us close to the maximum of C without actually reaching optimality[1].

[1] Although this is the traditional way to state the cost function of QAOA, if we want to be consistent with the minimization approach outlined previously by VQE and later with more general VQAs, we can do this by redefining the constraint $C_i(x) = 0$ if x satisfies the constraint and 1 otherwise.

This is the problem solved by QAOA. An interesting feature of QAOA is that the nature of the constraints C_i is highly application dependent, so although QAOA is an algorithm for optimization problems in general, it is more of a meta-algorithm which defines how one may tailor the VQA paradigm to a particular optimization problem of interest.

Since we would like to search over the space of bit strings to find the optimal x satisfying the most constraints. We initialize the input state of the QAOA algorithm in a uniform superposition over all states $|s\rangle = |+\rangle^{\otimes n}$.

Next we define two rotation operators:

The first of these rotation operators encodes the cost function into an exponential rotation by free parameter θ which we denote

$$U_C(\theta) = e^{-i\theta C} = \prod_{\alpha=1}^{m} e^{-i\theta C_\alpha}$$

The parameter θ is freely chosen and can be optimized from 0 to 2π in the classical post processing.

The next operator we define encodes the sum of individual bit flips over all qubits into an exponential rotation by a second free parameter ϕ. If we define X_j as the bit flip X acting on the jth qubit then we have

$$U_X(\phi) = e^{-i\phi \sum_{j=1}^{n} X_j} = \prod_{j=1}^{n} e^{-i\phi X_j}$$

Together these rotation operators U_C and U_X allow us to bias the uniform superposition according to the cost function to converge toward the optimal string [19].

As QAOA is only an approximate optimization algorithm, and given that any near term implementation is limited by noise, QAOA then defines the critical parameter $p \geq 1$. This parameter p fixes the depth of the algorithm by defining the rotation tuples $\Theta = (\theta_1, \theta_2, ...\theta_p)$ and $\Phi = (\phi_1, ...\phi_p)$. This is important because quantum information becomes corrupted by noise the longer the circuit runs. The parameter p effectively fixes the max runtime of the algorithm by specifying an algorithm with p gate layers - where each layer can be implemented in some fixed time dependent on the device runnign the circuit. We can then define the algorithm output state

$$|\Theta, \Phi\rangle = U_X(\phi_p)U_C(\theta_p)...U_X(\phi_1)U_C(\theta_1)|z\rangle$$

Referring to [19], we also rapidly define the expectation value and maximum expectation value over all $2p$ angles (Θ, Ω).

$$E_p(\Theta, \Omega) = \langle \Theta, \Omega | C | \Theta, \Omega \rangle$$
$$M_p = max_{\Theta,\Omega} E_p(\Theta, \Omega)$$

In [19] it is further shown that

$$M_p \geq M_{p-1} \tag{2.1}$$

and

$$\lim_{p\to\infty} M_p = \max_x C(x) \tag{2.2}$$

These results establish the strategy behind QAOA. Depending on the user's knowledge of the depth restrictions on their device, choose the largest reasonable p. This value can always be increased later to test hardware capacity and the result (1.1) shows this can only improve the optimized expectation value[2].

Once we fix some maximal p we then use the quantum computer to produce $|\Theta, \Omega\rangle$ which is then measured in the computational basis to produce a bit string x. We can then check how many constraints x satisfies to calculate $C(x)$. Because x is produced randomly, we would need to repeat this step several times to determine whether the state $|\Theta, \Omega\rangle$ has successfully biased the uniform superposition $|s\rangle$ to favor bit strings which approximately maximize C, which we confirm by measuring an increasing expectation value E_p.

[2]At this point we should pause for a moment to point out that this argument (we will produce ever better optimizations as we add layers and re-train) assumes (conjectures) that training additional layers will always be possible. Although QAOA may be the most famous example of this, the so-called 'layer-wise trainability conjecture', which appears throughout the VQA literature - and it also happens to be wrong [21]. Researchers in [21] showed, in 2021, that counterexample algorithms can be found where this conjecture fails and we can not be train beyond a particular depth p. We will come back to the issue of trainability in a future section.

Here is where classical optimization plays a role as we vary the parameters (Θ, Ω) to produce ever higher expectation values E_p. This part of the algorithm is nontrivial, and the trove of various classical optimization methods for choosing and searching for appropriate parameters to maximize the cost function is extensive and beyond the scope of this thesis.

Further we run into issues of trainability in the famous Barren Plateaus problem [16], which will be discussed in subsequent sections. Next, we will examine the basic concepts and tools that go into the construction of general variational hybrid algorithms and in further sections we delve into the trainability of VQAs in general.

2.1.3 General Structure of Variational Quantum Algorithms

Since the development of VQE and QAOA in 2014, Variational Quantum Algorithms (VQA)s have emerged as a leading paradigm to quantum algorithm development for near term devices [14].

One of the advantages of this paradigm is that it can be flexibly applied to a wide range of problems. In order to find a VQA which solves given application of interest, we first define the relevant parameter-dependent cost function $C(\boldsymbol{\alpha})$[3]. This cost function often suggests itself naturally relative to the problem to be solved. Despite this, one of the chief challenges of designing a VQA is designing a quantum algorithm which evaluates $C(\boldsymbol{\alpha})$ with some advantage over classical methods.

In the previous example of VQE, where we would like to find the ground energy state, the energy $\mathcal{H}$, or functions dependent on $\mathcal{H}$ present themselves as natural cost functions to be minimized. Here we chose to minimize the average $\langle \mathcal{H} \rangle$ of the energy because we have already established a protocol for evaluating expected values of Paulis and combinations of Paulis used in the construction of $\mathcal{H}$.

Extending this concept, we can consider a more general observable than energy, which we denote $\mathcal{O}$. The expectation value of this observable then gives us a more general cost function $C(\boldsymbol{\alpha})$ which we define in terms of the average or expected value of the observable $\mathcal{O}$ via

$$C(\boldsymbol{\alpha}) \equiv \langle \mathcal{O} \rangle = Tr(\mathcal{O}\rho_{out}) = Tr[\mathcal{O}U(\boldsymbol{\alpha})\rho_{in}U^{\dagger}(\boldsymbol{\alpha})] \tag{2.3}$$

Here $U(\boldsymbol{\alpha})$ is the parametrized unitary[4] implemented by the quantum computer given

[3]Here the cost function is averaged over all possible bit strings and is taken as a function of the k-parametric input $\boldsymbol{\alpha}$ rather than as a function of some particular bit string.

[4]We use the term unitary interchangeably with quantum algorithm.

k-parameter input $\boldsymbol{\alpha}$ which prepares the output state ρ_{out} from the input state ρ_{in} and where $Tr(A)$ is the trace of operator A. Given an ensemble of such observables $\{\mathcal{O}_i\}$, [14] extends this definition into the more general cost function

$$C(\boldsymbol{\alpha}) = \sum_i f_i(\langle \mathcal{O}_i \rangle) \tag{2.4}$$

Where $\{f_i\}$ are functions which are application-specific and generally provide more flexibility into the above definition. It is important to note that, for a given problem of interest, there may be multiple choices of $\{f_i\}$ which yield valid cost functions.

While the above (1.4) is a more general form, many of the cost functions we are interested in stick to the simpler case of

$$C(\boldsymbol{\alpha}) = Tr[\mathcal{O}U(\boldsymbol{\alpha})\rho U^\dagger(\boldsymbol{\alpha})] \tag{2.5}$$

In order for either of the above to be valid cost function, the cost function must be **T**rans-**FEM**:

First, we require that the classical machine learning or optimization algorithm be **Trainable**. This critical requirement turns out to be highly nontrivial and will need to be more thoroughly examined in the next section. Second, the minimum of the cost function must be **Faithful** by always corresponding to the desired solution. Third, the quantum circuit which implements the cost function must be **Efficient** to implement[5]. Fourth, a **Meaningful** cost function associates better approximate solutions to smaller values [14].

There are now more than 40 VQAs with FEM cost functions in the literature. While these characteristics are typically easy enough to verify on paper, the trainability of these algorithms remains only heuristically verified at best for the majority of existing cost functions. In the next section we outline the trainability problem and possible ways around this roadblock. This will allow us to produce a true Trans-FEM cost function for state diagonalization in the next chapter.

2.2 Trapped on a Barren Plateau: The Trainability Problem

Of the four properties that each cost function C must have, ensuring that the optimization landscape generated by C is actually trainablity which proves to be the most difficult

[5]Its depth can scale at most polynomially with number of qubits.

property to guarantee - at least in the sense that it is still a very open problem in the field. To compound this issue, a paper from 2018 [16] has shown that some choices of C are impossible to train when scaled up - this is the issue of barren plateaus.

More concretely, the issue of barren plateaus is as follows: For a given VQA specified by the cost function $C(\boldsymbol{\alpha}) = \langle \mathcal{O} \rangle$, we need to classically train our k-parameter vector $\boldsymbol{\alpha}$ to find the minimum of the $k+1$ landscape induced by C. One way we might accomplish this is by calculating the gradient $\partial_\beta C$ with respect to some directional component $\beta \in \boldsymbol{\alpha}$ and employing gradient descent.

In [16] it was shown that, if we initialize $\boldsymbol{\alpha}$ randomly at the start of the algorithm, then the average gradient

$$\langle \partial_\beta C(\boldsymbol{\alpha}) \rangle = 0$$

will not be biased in any particular direction. On its own this is not a problem, after all, it would be nice to start off without a bias so ideally we find the best direction for gradient descent. Given this average center around zero, we have the variance of the gradient given by

$$Var(\partial_\beta C) = \langle (\partial_\beta C)^2 \rangle.$$

This variance becomes the key parameter of interest to determine the trainability of our algorithm when we consider that Chebyshev's inequality bounds the probability that the gradient will deviate from its mean value of 0 via

$$Pr(\partial_\beta C \geq a) \leq \frac{Var(\partial_\beta C)}{a^2}. \tag{2.6}$$

This inequality allows us to calculate the variance for a particular type of VQA and determine whether it will not be trainable - this is what is done in [16].

Their result states that, if $U(\boldsymbol{\alpha})$ is at least $O(n^{1/d})$ deep, where n is the number of qubits and $d \geq 2$ is the dimension of the qudits, and we have no problem-specific structure built into the ansatz which structures $U(\boldsymbol{\alpha})$, then the variance will shrink exponentially as we increase the system size. Via Chebyshev, as the variance shrinks so too does the probability that the gradient will take on a significant nonzero value. Thus we see that in this context, we will have gradients which vanish exponentially with system size - thus

foiling our gradient descent. This phenomenon is called a barren plateau and its presence ensures that C will be untrainable. Since the discovery of this phenomenon in 2018, the number of VQAs which have barren plateaus has only increased, and in this section we will survey the existing 3 years of literature on this phenomenon to develop a comprehensive understanding of the implications of barren plateaus for the future of the field.

Since this initial result in 2018, a great deal of interest has been devoted to understanding how systemic the issue of barren plateaus was for the field. Although the initial result focused on gradient descent facilitated by first order derivatives, it was further shown in 2020 that these exponentially vanishing gradients also apply to higher order derivatives as well [22]. This effectively killed gradient-based methods for training, and in 2020, researchers investigated whether gradient-free methods were similarly affected by barren plateaus or whether these could be used to train in the presence of a barren plateaus. If information about gradients of C is not available, then another data source which is used by many gradient-free optimization schemes is cost function values $C(\boldsymbol{\alpha}_i)$ for multiple inputs $\boldsymbol{\alpha}_i$. Analytical results showed that differences between consecutive points in the cost function landscape $C(\boldsymbol{\alpha}_A) - C(\boldsymbol{\alpha}_B)$, also vanish exponentially with system size [23]. Thus distinct points in our data set $\{C(\boldsymbol{\alpha}_i)\}_{i \in Index}$ require exponential resources to distinguish and thus lose their value in navigating C. This has broad implications for the entire field of gradient-free algorithms, though [23] provides concrete numerical verification for Nelder-Mead, Powell, and COBYLA algorithms.

While we have these examples of types of VQAs we can design which exhibit barren plateaus, we also find examples of *problem classes* which have barren plateaus regardless of the structure of your VQA. The only case we discuss here involves learning the unitary dynamics behind *scrambling* processes [24].

To lead our discussion of scrambling processes, we explore the example of scrambling of information which is dropped into a black hole. Concretely, if our information were contained in the state $|x\rangle$, then the black hole will scramble this information and redistribute it via the emission of Hawking radiation. If we collect all of this radiation, and if we share a maximally entangled state (e.g. one part of $|\psi\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$) with the black hole, then we can recover the scrambled information $U|x\rangle$, where U is referred to as the scrambling unitary. If U is known, or *if we can learn* U, then we can *unscramble* $|x\rangle$ and recover the information which would have been otherwise lost to the black hole. The above example is known as the Hayden-Preskill thought experiment [25]. While this example may seem, otherworldly, and therefore irrelevant to practical algorithm development, consider that black holes are merely the most efficient example of what is actually a very wide category of approximate scramblers (e.g. handwriting a note and putting it through a blender). In addition, by studying VQAs in this context, we can find results which have big implications

for our field right here on Earth.

This year, researchers in [24] investigated whether quantum machine learning (QML), which in this context is indistinguishable with the field of VQAs, can be employed to learn the scrambling unitary U. In the original result of [16], it was found that barren plateaus emerge, in some sense as a consequence of the inherent randomness of the starting unitary we are training. In [24], a complementary result is found where if the target scrambler U is *sufficiently random*, then barren plateaus are guaranteed. What is important to illustrate, is that these barren plateaus are agnostic to the structure, or ansatz, of your algorithm. As such this problem case represents the potential start of an as yet poorly understood whole family of problems for which no amount of clever design will allow us to solve using VQAs.

In addition to the existence of problems which are untrainable, as we are dealing with NISQ algorithms, it should be noted that the above results apply even to noise-free scenarios. As no VQA is likely to operate in a noise-free environment anytime soon, it is important for us to understand the impact of noise on circuit training. In early 2021 it was shown that noise can actually induce barren plateaus provided the depth of the circuit grows at least linearly with system size [26]. The noise model behind this result breaks $U(\boldsymbol{\alpha})$ into a series of layers, similar to the approach of QAOA, and implements single qubit Pauli noise operation $\mathcal{N}(\sigma) = c_\sigma \sigma$ between each layer where $\sigma \in \{I, X, Y, Z\}$. Here we restrict the constant $-1 \leq c_X, c_Y, c_Z \leq 1$ and define the noise rate $r = max c_x, c_Y, c_Z$. From this it can be shown that in a noisy environment representable by the above noise model, only circuits which have sublinear scaling remain possible candidates for successful training.

Given each of these examples where barren plateaus eliminate any possibility of training our VQA, and the fact that this field will only continue to grow, it behooves us to find ways to avoid barren plateaus.

2.3 Escaping the Barren Plateau: Cost function dependent barren plateaus

With the problem of barren plateaus fresh in our minds, we now turn to a new result from 2021 that shows how barren plateaus are dependent on the global or local nature of the

cost function [2]. Recalling the general structure of the cost function given by

$$\begin{aligned} C(\boldsymbol{\alpha}) &= \sum_i f_i(Tr[\mathcal{O}_i U(\boldsymbol{\alpha})\rho_{in}U^\dagger(\boldsymbol{\alpha})]) \\ &= \sum_i f_i(\langle\mathcal{O}_i\rangle), \end{aligned}$$

then the key result from [2] was to show that the observables $\{\mathcal{O}_i\}$ we choose have a direct impact on whether each $C_i(\boldsymbol{\alpha})$ will be trainable. In further notation we use ρ to represent ρ_{in} unless otherwise specified. Considering again the simplified form of

$$C(\boldsymbol{\alpha}) = Tr[\mathcal{O}_{G/L}U(\boldsymbol{\alpha})\rho U^\dagger(\theta)],$$

where the observable

$$\begin{aligned} \mathcal{O}_{G/L} &= c_0 I + \hat{O}_{G/L} \\ \text{where } \hat{O}_{G/L} &= \sum_{i=1}^{N} c_i \hat{O}_i \end{aligned}$$

is called global or local depending on the structure of $\hat{O}_i$ which we define in the coming subsections, where I is the identity and each $c_i \in \mathbb{R}$.

If $\mathcal{O}$ is global, then it has been shown in [2] that the cost function derived from this observable will encounter a barren plateau and will thus be unable to train at scale. Conversely, if this observable is local, then it has been similarly shown that no barren plateau will occur and training will be possible.

2.3.1 Global

Here we define the global observable $\mathcal{O}_G = c_0 I + \hat{O}_G$ where the operator $\hat{O}_G = \sum_{i=1}^{N} c_i \hat{O}_i$ is a linear combination of N nontrivial operators $\hat{O}_i = \hat{O}_{1,i} \otimes \hat{O}_{2,i} \otimes ... \otimes \hat{O}_{n,i}$. There are two forms of $\hat{O}_G$ which are known to produce barren plateaus via [2]:

1. If $N = 1$ and each $\hat{O}_k$ is a nontrivial projector of rank r_k acting on the kth subsystem ($Ex.$ $\hat{O}_G = |00...0\rangle\langle 00...0| = \bigotimes_{k=1}^{n} \hat{O}_k$ where $\hat{O}_k = |0\rangle\langle 0|_k$)
2. If $N > 1$ and each $\hat{O}_{ki}$ acts on m qubits such that $Tr(\hat{O}_{ki}) = 0$ and $Tr(\hat{O}_{ki}^2) \leq 2^m$ ($Ex.$ $\hat{O}_G = \sum_{i=1}^{N} c_i(\hat{O}_{1,i} \otimes \hat{O}_{2,i} \otimes ... \otimes \hat{O}_{n,i})$ where $\hat{O}_{ik} = \bigotimes_{j=1}^{m} \sigma_j$ is a nontrivial tensor product of m paulis $\sigma_j \in \{I, X, Y, Z\}$)

2.3.2 Local

Next we define the local observable $\mathcal{O}_L = c_0 I + \hat{O}_L$ where the operator $\hat{O}_L = \sum_{i=1}^{N} c_i \hat{O}_i$ such that $\hat{O}_i$ acts on at most m qubits such that $\hat{O}_i = \hat{O'}_i \otimes \hat{O'}_{i'}$, where $\hat{O'}_i$ is a tensor product of Paulis acting on $m/2$ qubits.

By understanding the structure of these global and local observables, we enable ourselves to recognize these types of observables we they appear in existing VQAs as well as any new VQAs we develop.

Note: The above definitions do not in fact constitute a dichotomy as the terms global & local may suggest. These descriptions should be considered a partial understanding of the factors which give rise to barren plateaus. If an observable happens to fall in one of the above two categories then we will know whether they will encounter a barren plateau - otherwise these results have little to say.

Chapter 3

Trainability of Variational Quantum State Diagonalization

Now that we have set the stage, we turn to the novel calculation of this thesis: showing that the Variational Quantum State Diagonalization (VQSD) algorithm which Ryan LaRose, Arkin Tikku, Lukasz Cincio, Patrick Coles, and myself published in 2019 will *not be trainable at scale* and how we can fix it.

We will start by explaining the details of VQSD and understand the cost function used in our original paper [27][1]. Then we will apply results from [2] to prove that VQSD will not scale with this cost function.

3.1 VQSD: How it Works

The VQSD algorithm was designed to perform matrix diagonalization in the variational context. The real world context in which this would take place is by representing the matrix we wish to diagonalize as the density matrix ρ of some quantum system we can prepare many times in the laboratory. Once we have diagonalized this quantum state, we are able to extract the eigenvectors and eigenvalues of the matrix.

This quantum matrix representation has some natural advantages over classical diagonalization methods in the limit of large system size. Specifically, for sufficiently large

[1]Prior to adopting my preferred name of Joan Arrow, I went by my legal name of Étude O'Neel-Judy, you can find my early publications under this name.

numbers of qubits N, the resulting density matrix will have $2^N \times 2^N$ entries, an exponential number which would not be efficient to even write down, let alone diagonalize. We see then that a quantum approach to matrix/state diagonalization has the potential for application in exponential regimes inaccessible to classical methods. The VQSD algorithm is shown in figure 3.1 from [20].

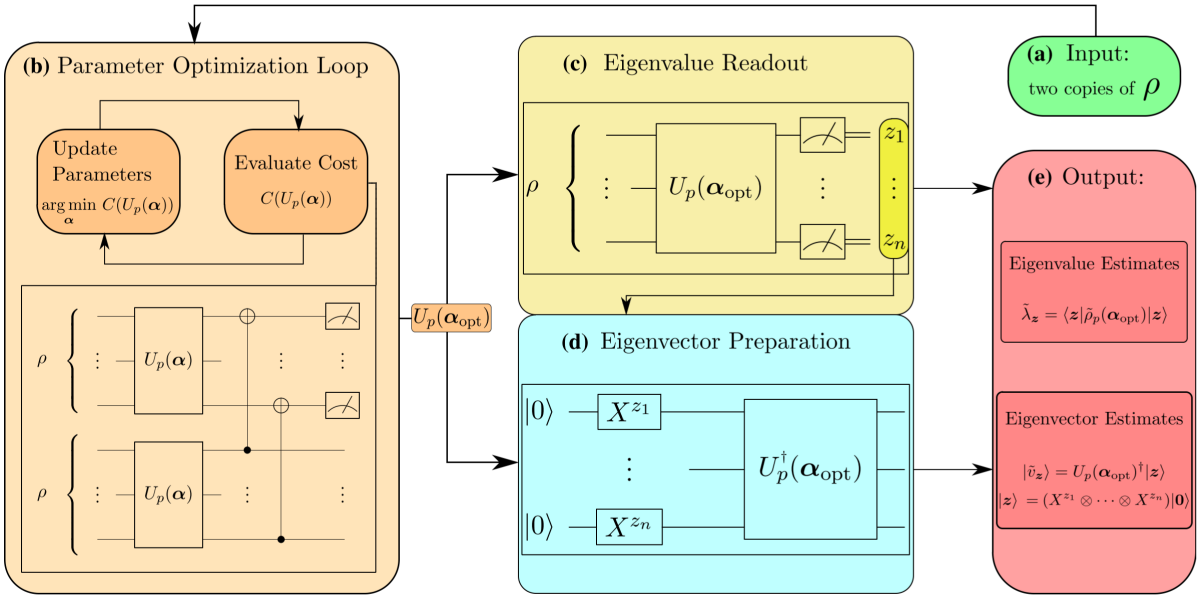
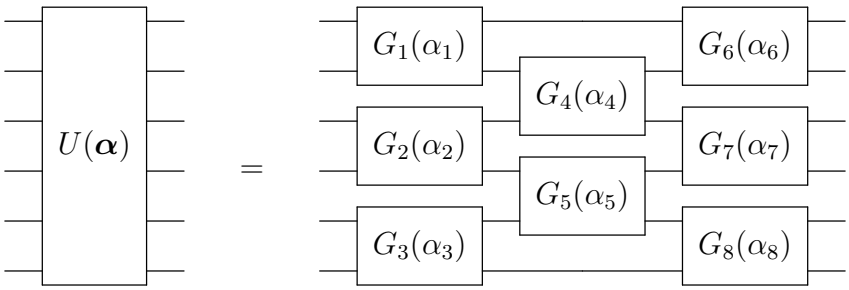


Figure 3.1: Diagram of VQSD

The algorithm takes as input many copies of the quantum state ρ, an initial guess for the parameter vector $\boldsymbol{\alpha}$, and a depth p chosen as shallow as possible to mitigate noise. These inputs are then used to determine the approximate diagonalizing $U_p(\boldsymbol{\alpha})$, and applies this unitary to two copies of ρ. This produces two copies of the approximately diagonal state $\tilde{\rho} = U_p(\boldsymbol{\alpha})\rho U_p^\dagger(\boldsymbol{\alpha})$. Here, the structure of $U_p(\boldsymbol{\alpha})$ is given by the alternating layered ansatz shown bellow for $p = 3$ on 6 qubits without connectivity between the top and bottom qubits[2]. The two-body operation $G_i(\boldsymbol{\alpha}_i)$ is left without further specifying in order to allow this ansatz to be agnostic to the native gate set of a particular device.

[2]Note that if connectivity allows, we also include a two-body operation between top and bottom qubits



If we are able to successfully minimize the cost function and obtain the optimal diagonalizing unitary $U_p(\boldsymbol{\alpha}_{opt})$, then we can proceed to use $U_p(\boldsymbol{\alpha})$ for eigenvalue readout and eigenvector preparation as shown in 3.1 (c) and (d). This provides the capability of eigenvalue spectroscopy and eigenvector preparation for use in subsequent condensed matter experiments and is the main application of our algorithm.

3.1.1 Cost Function

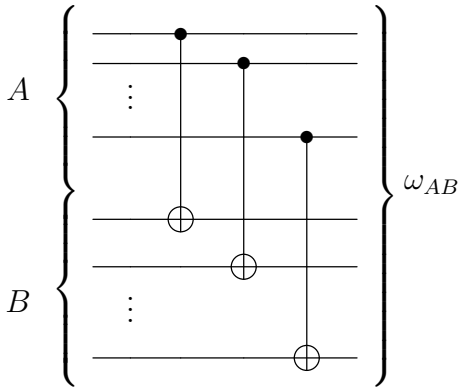
The cost function which was used in our experimental implementations, and which measures 'how approximate' the approximately diagonalized quantum state $\tilde{\rho}$ we have produced is is given by

$$C(\tilde{\rho}) = Tr(\rho^2) - Tr((D(\tilde{\rho}))^2). \tag{3.1}$$

Here D is the *dephasing channel* which artificially diagonalizes the input state by simply deleting off diagonal terms. Taken together, when we take the trace of the dephased state and subtract this value from the original purity $Tr(\rho^2)$, we produce 0 if and only if $Tr((D(\tilde{\rho}))^2) = Tr(\rho^2)$ which only can happen if dephasing has no impact on $\tilde{\rho}$, i.e. $\tilde{\rho}$ is diagonal. If we are guaranteed ideal conditions, we can assume $Tr(\rho^2) = 1$, otherwise we may simply calculate the purity of the input state at the start of the algorithm (which can be done efficiently [28]) and reuse that value for each subsequent training step.

The key insight at the heart of this cost function is that D is not a unitary gate. So as long as D acts nontrivially on the state, i.e. so long as off-diagonal terms remain after the attempt at diagonalization, D will reduce the purity and leave room from improvement.

The quantum circuit which we can use to calculate $Tr((D(\tilde{\rho}))^2)$ is simply a single layer of entangling controlled-not (CX) gates, given bellow for arbitrary input density matrices A, B, so named to identify their corresponding registers for later use:



Tracing out the A system and measuring the remaining subsystem ω_B in the computational basis then yields the desired value $Tr(D(A)D(B)) = \sum_z A_{z,z}B_{z,z}$ encoded into the probability of measuring the all-zero state $|00...0\rangle$.

This final statement is non-obvious, so to understand it here we reproduce the proof from [20]:

We start on the left-hand side of the above circuit with the input state $A \otimes B$. The proof works in this general case, but keep in mind that for state diagonalization, we also have $A = B = \tilde{\rho}$.

After applying our layer of CNOTs, we produce the state ω_{AB} given by

$$\omega_{AB} = \sum_{z,z'} |z\rangle \langle z| A |z'\rangle \langle z'| \otimes X^z B X^{z'}.$$

Where $X^z = X^{z_1} \otimes X^{z_2} \otimes ...X^{z_n}$ is a series of n bit flip (or NOT) gates and we are summing over all possible n-bit strings z, z'. By *tracing out* the A system, i.e. by eliminating the degrees of freedom given by that system, we produce:

$$\begin{aligned}\omega_B &= \sum_z \langle z| A |z\rangle X^z B X^z \\ &= \sum_z A_{z,z} X^z B X^z.\end{aligned}$$

We then calculate the probability of the all zero outcome:

$$\begin{aligned}\langle 00...0| \omega_B |00...0\rangle &= \langle 00...0| \sum_z \tau_{z,z} X^z B X^z |00...0\rangle \\ &= \sum_z A_{z,z} \langle 00...0| X^z B X^z |00...0\rangle \\ &= \sum_z A_{z,z} \langle z| B |z\rangle \\ &= \sum_z A_{z,z} B_{z,z} \\ &= Tr(D(A)D(B)).\end{aligned}$$

A critical advantage of this cost function, is that we may perform this quantum calculation in constant depth 1, regardless of system size.

3.2 Investigating Trainability

Now that we have familiarized ourselves with VQSD, we turn to the question of whether this algorithm's cost function will allow training at scale or whether it will encounter a barren plateau.

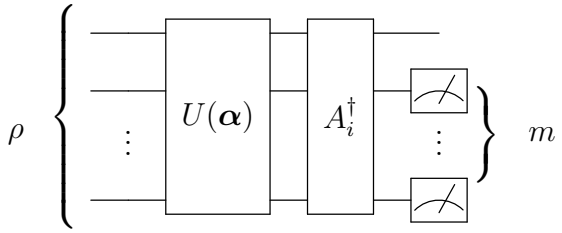
Since barren pleateaus have been found to be dependent on cost function locality, and since each VQA cost function $C(\boldsymbol{\alpha}) = \langle \mathcal{O} \rangle$ can be expressed in terms of the expectation value of some observable $\mathcal{O}$, we need only determine whether the observable associated with VQSD is sufficiently local or not.

3.2.1 Identifying $\langle \mathcal{O}_i \rangle$

Here we detail the basic process used by VQAs to calculate the value of the cost function for a given training parameter $\boldsymbol{\alpha}$ and input state ρ. By understanding this basic VQA

algorithm in explicit detail, we will see how to identify the observable of interest for a particular VQA given its cost function. Note that this basic process may be have to be repeated for every $\mathcal{O}_i$ if the cost function depends on several observables.

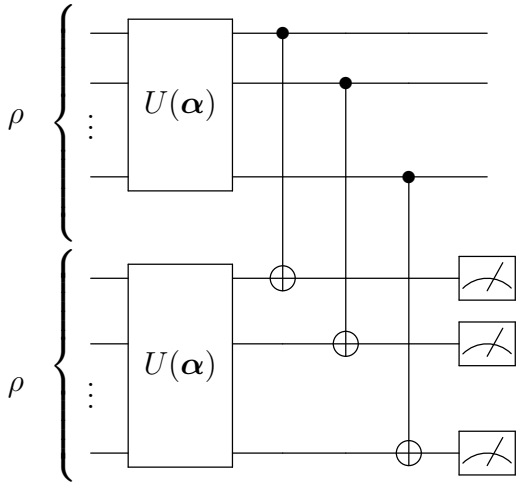
1. Input state $\rho = |\psi\rangle\langle\psi|$ into the device and set parameter α.
2. Apply unitary $U(\boldsymbol{\alpha})$ to produce $\rho' = U(\boldsymbol{\alpha})|\psi\rangle\langle\psi|U^\dagger(\alpha) = |\psi'\rangle\langle\psi'|$.
3. Perform a measurement of an m qubit subsystem in the
$$\mathcal{O}_i = A_i(I^{\otimes n-m} \otimes Z^{\otimes m})A_i^\dagger \tag{3.2}$$
basis as shown in the following circuit diagram for some $m \leq n$. Here Z is the Pauli operator and $A_i^\dagger$ is the matrix which rotates the $Z^{\otimes m}$ eigenbasis to the eigenbasis of the given $\mathcal{O}_i$.
4. Repeat steps 1-3 many times to approximate $|p_j|^2$, the probability of observing bit string j in order to calculate the expectation value $\langle\mathcal{O}_i\rangle = \sum_j z_j|p_j|^2$ where z_j is the jth eigenvalue of $Z^{\otimes n}$.



This $A_i^\dagger$ matrix and the subsequent measurement **is** the physical realization of the cost function, and given this circuit for calculating the cost function of a particular VQA, we can then determine the locality of the observable.

3.2.2 Theorem 1: VQSD will not scale

Next we turn to the locality of VQSD. Applying our knowledge of this algorithm to the structure of the previous section, we have the following circuit for one full run of the algorithm:



Here we identify the observable of interest by recalling that the cost function circuit $A_i^\dagger = CX^{\otimes n}$ encodes the value of the cost function (corresponding to how well we diagonalized the input state) into the probability of the all zero state in the bottom register. This output observable is captured by the operator

$$\mathcal{O}_{output} = I^{\otimes n} \otimes (|0\rangle \langle 0|)^{\otimes n}.$$

By inspection, this observable will produce a cost function which is maximized when the probability that the bottom register is in the $|00...0\rangle$ state is similarly maximized.

If we would like to recast this as a minimization problem, then we can implement

$$\mathcal{O'}_{output} = I^{\otimes 2n} - \mathcal{O}_{output}.$$

Which, for some arbitrary pure output state $|\psi\rangle \in \mathcal{H}_{2^n}$ has the cost function

$$\begin{aligned} C(\boldsymbol{\alpha}, \mathcal{O'}_{output}) &= Tr(\mathcal{O'}_{output} |\psi\rangle \langle\psi|) \\ &= \langle\psi| \mathcal{O'}_{output} |\psi\rangle \\ &= \langle\psi| I^{\otimes 2n} - \mathcal{O}_{output} |\psi\rangle \\ &= \langle\psi| I^{\otimes 2n} |\psi\rangle - \langle\psi| \mathcal{O}_{output} |\psi\rangle \\ &= 1 - C(\boldsymbol{\alpha}, \mathcal{O}_{output}) \end{aligned}$$

Now we have seen the way the output observable $\mathcal{O}_{output}$ facilitates diagonalization, let us analyze whether this input state produces a local or global observable $\mathcal{O}_{VQSD}$ once we conjugate by the cost function.

$$\mathcal{O}_{VQSD} = CX^{\otimes n}(I^{\otimes n} \otimes (|0\rangle\langle 0|)^{\otimes n})CX^{\otimes n}. \tag{3.3}$$

Note, intuitively we might expect this observable to be global since a key component of the observable, namely $|00...0\rangle\langle 00...0| = \bigotimes_{i=1}^{n}(|0\rangle\langle 0|)_i = \bigotimes_{i=1}^{n} P_i$ *is a tensor product of rank 1 projectors and as such fits the definition of a global operator*

Now we can turn to proving our first original result:

Theorem 1. *$\mathcal{O}_{VQSD}$ is in fact a global observable and as such will produce a barren plateau and prevent training at scale.*

Proof. First we simplify $\mathcal{O}_{VQSD}$ to show that it has global structure similar to that shown above.

$$\begin{aligned}
\mathcal{O}_{VQSD} &= \sum_{z,z'} |z\rangle\langle z| I^{\otimes n} |z'\rangle\langle z'| \otimes X^z |00...0\rangle\langle 00...0| X^{z'} \\
&= \sum_{z} |z\rangle\langle z| \otimes |z\rangle\langle z| \\
&= \bigotimes_{j=1}^{n} \sum_{z_j \in \{0,1\}} |z_j z_j\rangle\langle z_j z_j| \\
&= \bigotimes_{j=1}^{n} (|00\rangle\langle 00| + |11\rangle\langle 11|)_j
\end{aligned}$$

where z, z' are n-bit strings and z_j is a bit in the jth position of z.

We can define the rank 2 projector $P = |00\rangle\langle 00| + |11\rangle\langle 11|$, to produce

$$\mathcal{O}_{VQSD} = \bigotimes_{j=1}^{n} P_j. \tag{3.4}$$

From this form we can recognize the VQSD observable as a global observable made from an n-fold tensor product of rank 2 projectors. From [2], we can then conclude that the cost function of VQSD will encounter a barren plateau when employed at scale and thus will not be trainable. □

With this first key result in the bag, we can now push further to identify a new, local observable which will enable VQSD to be salvaged.

3.2.3 Theorem 2: Fixing VQSD

Next we turn our attention to fixing VQSD with a sufficiently local observable that nevertheless preserves the action of the diagonalizing cost function which enables this application to work.

Since we still need an output observable which is maximized when the probability that bottom register is $|00...0\rangle$ is also maximized, we naively try the hyper-local observable from the previous chapter modified for our two-register context with VQSD

$$\mathcal{O}_{output} = I^{\otimes n} \otimes (\frac{1}{n}\sum_{i=1}^{n}(|0\rangle\langle 0|)_i \otimes I_{i'}) \tag{3.5}$$

where we define $(|0\rangle\langle 0|)_i \otimes I_{i'} = I \otimes ... \otimes |0\rangle\langle 0| \otimes ... \otimes I$ projects the ith qubit onto zero and does nothing to each other qubit. We divide by the number of qubits to make sure we produce cost function values between 1 and 0.

Given an arbitrary output state $|\psi\rangle$, this observable produces the following cost function

$$\begin{aligned} C(\boldsymbol{\alpha}, \mathcal{O}_{output}) &= Tr(\mathcal{O}_{output}|\psi\rangle\langle\psi|) \\ &= \langle\psi|\mathcal{O}_{output}|\psi\rangle \\ &= \langle\psi| I^{\otimes n} \otimes \frac{1}{n}\sum_{i=1}^{n}(|0\rangle\langle 0|)_i \otimes I_{i'}|\psi\rangle \\ &= \frac{1}{n}\sum_{i=1}^{N} Prob(0)_{Bi} \end{aligned}$$

where $Prob(0)_{Bi}$ is the probability that the ith qubit in the bottom register is measured in the zero state. Thus we see that this cost function is maximized when each individual probability $Prob(0)_{Bi}$ is maximized simultaneously, thereby maximizing the probability of the all zero state.

Next we prove our second result:

Theorem 2. *The observable*

$$\mathcal{O'}_{VQSD} = CX^{\otimes n}(I^{\otimes n} \otimes (\frac{1}{n}\sum_{i=1}^{n}(|0\rangle\langle 0|)_i \otimes I_{i'}))CX^{\otimes n} \tag{3.6}$$

does in fact produce a local observable which induces a new cost function for VQSD that is immune to barren plateaus.

Proof. The structure of this calculation will closely mirror that shown in the previous section. Here we intend to show that $\mathcal{O'}_{VQSD}$ has a local structure similar to that shown above.

$$\begin{aligned}
\mathcal{O'}_{VQSD} &= \sum_{z,z'} |z\rangle\langle z| I^{\otimes n} |z'\rangle\langle z'| \otimes X^z(\frac{1}{n}\sum_{i=1}^{n}(|0\rangle\langle 0|)_i \otimes I_{i'}))X^{z'} \\
&= \sum_{z} |z\rangle\langle z| \otimes (\frac{1}{n}\sum_{i=1}^{n}(|z_i\rangle\langle z_i|) \otimes I_{i'})) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{z_1,z_2,...,z_n\in\{0,1\}} |z\rangle\langle z|_A \otimes (|z_i\rangle\langle z_i|) \otimes I_{i'})_B \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{z_1,z_2,...,z_n\in\{0,1\}} (|z_1\rangle\langle z_1|_A \otimes ... \otimes |z_n\rangle\langle z_n|_A) \otimes (I_B \otimes ... \otimes |z_i\rangle\langle z_i|_B \otimes ... \otimes I_B) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{z_1,z_2,...,z_n\in\{0,1\}} (|z_1\rangle\langle z_1|_A \otimes I_B) \otimes ... \otimes |z_iz_i\rangle\langle z_iz_i| \otimes ... \otimes (|z_n\rangle\langle z_n|_A \otimes I_B)
\end{aligned}$$

Here we take a pause. We have reorganized our terms to place the ith qubit of register A beside the ith qubit of register B - we do this because this is the adjacency relationship enacted by the series of CX gates in the cost function. We have then placed next to each other the qubits which have been acted on by the cost function, which will help us simplify.

An identity which we need to resolve this calculation is given by

$$M \otimes (N_1 + N_2) = M \otimes N_1 + M \otimes N_2. \tag{3.7}$$

That is, we may add N_i terms componentwise across a tensor product of matrices provided they share a constant M term. Further, we recall that the identity matrix can be constructed via

$$\sum_{i\in\{0,1\}} |z_i\rangle\langle z_i| = |0\rangle\langle 0| + |1\rangle\langle 1| = I. \tag{3.8}$$

Now we are ready to complete the proof by summing each $j \neq i$ term to the identity

$$\begin{aligned}
\mathcal{O'}_{VQSD} &= \frac{1}{n}\sum_{i=1}^{n} \sum_{z_1,z_2,\ldots,z_n\in\{0,1\}} (|z_1\rangle\langle z_1|_A \otimes I_B) \otimes \ldots \otimes |z_i z_i\rangle\langle z_i z_i| \otimes \ldots \otimes (|z_n\rangle\langle z_n|_A \otimes I_B) \\
&= \frac{1}{n}\sum_{i=1}^{n} \sum_{z_1,z_2,\ldots,z_{n-1}\in\{0,1\}} (|z_1\rangle\langle z_1|_A \otimes I_B) \otimes \ldots \otimes |z_i z_i\rangle\langle z_i z_i| \otimes \ldots \otimes (I_A \otimes I_B) \\
&= \vdots \\
&= \frac{1}{n}\sum_{i=1}^{n} \sum_{z_i\in\{0,1\}} (I_A \otimes I_B) \otimes \ldots \otimes (|z_i z_i\rangle\langle z_i z_i|) \otimes \ldots \otimes (I_A \otimes I_B) \\
&= \frac{1}{n}\sum_{i=1}^{n} (I_A \otimes I_B) \otimes \ldots \otimes (|00\rangle\langle 00| + |11\rangle\langle 11|)_i \otimes \ldots \otimes (I_A \otimes I_B) \\
&= \frac{1}{n}\sum_{i=1}^{n} (|00\rangle\langle 00| + |11\rangle\langle 11|)_i \otimes I_{i'} \\
&= \frac{1}{n}\sum_{i=1}^{n} P_i \otimes I_{i'}.
\end{aligned}$$

Note how this observable is remarkably similar in structure to our starting hyper-local observable. Due to the linear nature of the trace, the cost function will be an averaged sum over $\langle P_i \otimes I_{i'}\rangle$ given by

$$C(\theta, \mathcal{O'}_{VQSD}) = \frac{1}{n}\sum_{i=1}^{n} \langle P_i \otimes I_{i'}\rangle. \tag{3.9}$$

We can now analyze whether this new observable is still local enough to guarantee trainability. Given the above structure of $\mathcal{O'}_{VQSD}$, we fit the definition of a local observable provided P_i can be equivalently replaced by a tensor product of paulis.

Exploring this concept further, if we write the projector P_i in terms of Paulis

$$P_i = (|00\rangle\langle 00| + |11\rangle\langle 11|)_i = \frac{1}{2}(Z_1 \otimes Z_2 + I \otimes I).$$

Taking the expectation value of $\langle P_i \otimes I_{i'}\rangle$ we have $\langle P_i \otimes I_{i'}\rangle = \langle \frac{1}{2}(Z_1 \otimes Z_2 + I_1 \otimes I_2) \otimes I_{i'}\rangle = \frac{1}{2}(\langle Z_1 \otimes Z_2 \otimes I_{i'}\rangle + \langle I \otimes I \otimes I_{i'}\rangle) = \frac{1}{2}(\langle Z_1 \otimes Z_2 \otimes I_{i'}\rangle + 1)$

Where we obtain the last equality since $\langle I \otimes I \otimes I_{i'}\rangle = Tr(I^{\otimes 2n}|\psi\rangle\langle\psi|) = \langle\psi|\psi\rangle = 1$. Thus we see that our cost function

$$\begin{aligned} C(\theta, \mathcal{O'}_{VQSD}) &= \frac{1}{n}\sum_{i=1}^{n}\langle P_i \otimes I_{i'}\rangle \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(\langle Z_1 \otimes Z_2 \otimes I_{i'}\rangle + 1) \\ &= \frac{1}{2n}(n + \sum_{i=1}^{n}(\langle Z_1 \otimes Z_2 \otimes I_{i'}\rangle)) \\ &= \frac{1}{2} + \frac{1}{2n}\sum_{i=1}^{n}(\langle Z_1 \otimes Z_2 \otimes I_{i'}\rangle) \\ &= \frac{1}{2} + C'(\theta, \mathcal{O'}_{VQSD}) \end{aligned}$$

where the new cost function C' matches perfectly the definition of a local cost function. Thus we see that the hyper-local output observable $\mathcal{O'}_{output}$ does in fact induce a trainable algorithm for VQSD. □

As a final remark, note how the cost function circuit $A^\dagger = CX^{\otimes n}$ for VQSD did not radically change the structure of the global/local output observables $(\mathcal{O}_{output}, \mathcal{O'}_{output})$. Recall that this whole process outlined above boiled down to

$$\mathcal{O}_{output} = \bigotimes_{i=1}^{n}|0\rangle\langle 0|_i \longrightarrow \mathcal{O}_{VQSD} = \bigotimes_{i=1}^{n}(|00\rangle\langle 00| + |11\rangle\langle 11|)_i$$

$$\mathcal{O'}_{output} = \frac{1}{n}\sum_{j=i}^{n}|0\rangle\langle 0|_j \otimes I_{j'} \longrightarrow \mathcal{O'}_{VQSD} = \frac{1}{n}\sum_{j=i}^{n}(|00\rangle\langle 00| + |11\rangle\langle 11|)_j \otimes I_{j'}.$$

This may not be terribly surprising since in this case the cost function circuit was very simple. However, it does beg the question as we step back and consider other possible cost function circuits, whether general properties of those circuits can be identified which 'delocalize' the local output observable. To better understand this more general behavior, we outline the next steps for this research.

Chapter 4

Next Steps & Future Research

The calculations of the previous section enabled us to interrogate the cost function of VQSD and determine whether this cost function will live up to its promise in real implementations.

The follow-up of this work is a wholesale audit of the entire field of VQAs, first to determine whether the cost functions upon which they are built are Trans-FEM, and if not, to propose more localized observables which solve the same task while also avoiding barren plateaus. This analysis is important because it will shine a light on the efficacy of this field by showing how many VQAs can be said to actually avoid barren plateaus and perform in the scale regime for which they are designed. Without a clear understanding of this issue, the field of VQAs will remain speculative in its real promise.

Here we enumerate the majority of the VQAs in the literature, organized with newer algorithms on top, and the problem they endeavor to solve:

1. Circuit compiler for quantum error correction [29]
2. Nonlinear system solver [30]
3. Linear system solver [31]
4. General quantum simulator [32]
5. Hamiltonian diagonalization for dynamical quantum simulation [33]
6. Fast-forwarding for quantum simulation beyond coherence time [34]
7. Quantum Fisher Information estimator [35]

8. Adaptive circuit learning for quantum metrology [36]
9. State quantum metrology [37]
10. Singular value decomposer [27]
11. Fidelity estimator [38]
12. Training deep quantum neural networks [39]
13. The Born Supremacy [40]
14. Circuit-centric quantum classifiers [41]
15. Unsampling on a photonic processor [42]
16. Theory of variational quantum simulation [43]
17. Accelerated VQE [18]
18. Factoring [44]
19. Quantum simulation of imaginary time evolution [45]
20. Quantum circuit compilation [46]
21. Quantum subspace simulator [47]
22. Spin-squeezing [48]
23. Consistent histories for quantum foundations [49]
24. Quantum convolutional neural networks [50]
25. Generative adversarial quantum machine learning [51]
26. Training neural networks using low-depth circuits [52]
27. Supervised learning with quantum enhanced feature spaces [53]
28. QML in feature hilbert spaces [54]
29. Addressing hard classical problems with adiabatically assisted Variational Quantum Eigensolvers [55]

30. Variational Quantum Gate Optimization [56]
31. Classification with QNN [57]
32. Differentiable learning of quantum circuit Born machines [58]
33. Quantum circuit learning [59]
34. Efficient Variational Quantum Simulator Incorporating Active Error Minimization [60]
35. Quantum autoencoders for efficient compression of quantum data [61]
36. QVECTOR: an algorithm for device-tailored quantum error correction [62]
37. Performance of QAOA on Typical Instances of Constraint Satisfaction Problems with Bounded Degree [63]
38. A Quantum Approximate Optimization Algorithm (QAOA) [19]
39. A variational eigenvalue solver on a photonic quantum processor (VQE) [15]

Beyond analysis of each of the above, answering the more general question of which circuit cost functions $A^\dagger$ are able to 'delocalize' our hyper-local output observable $\mathcal{O'}_{output}$ will help us to better and more quickly establish whether $A^\dagger \mathcal{O'}_{output} A$ is local or not without having to explicitly calculate the result every time.

References

[1] https://nandgame.com/, "NandGame - Build a computer from scratch."

[2] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature Communications*, vol. 12, p. 1791, Mar. 2021.

[3] R. P. Feynman, "Quantum mechanical computers," *Foundations of Physics*, vol. 16, pp. 507–531, June 1986.

[4] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Nov. 1994.

[5] P. W. Shor, "Why haven't more quantum algorithms been found?," *Journal of the ACM*, vol. 50, pp. 87–90, Jan. 2003.

[6] P. Heimann, "Reviews : The Born-Einstein Letters. Correspondence between Albert Einstein and Max and Hedwig Born from 1916 to 1955 with commentaries by Max Born. Translated by Irene Born. London, Macmillan, I970. 320 pp. £3.85," *European Studies Review*, vol. 3, pp. 198–199, Apr. 1973. Publisher: SAGE Publications.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, STOC '96, (New York, NY, USA), pp. 212–219, Association for Computing Machinery, July 1996.

[8] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto, "Requirements for fault-tolerant factoring on an atom-optics quantum computer," *Nature Communications*, vol. 4, p. 2524, Oct. 2013. Number: 1 Publisher: Nature Publishing Group.

[9] C. Gidney and M. Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, Apr. 2021. arXiv: 1905.09749.

[10] M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck, "Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3," *arXiv:1603.09383 [quant-ph]*, Nov. 2016. arXiv: 1603.09383.

[11] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, p. 032324, Sept. 2012. Publisher: American Physical Society.

[12] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, Oct. 2019. Number: 7779 Publisher: Nature Publishing Group.

[13] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, pp. 625–644, Sept. 2021.

[15] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, p. 4213, July 2014. Number: 1 Publisher: Nature Publishing Group.

[16] 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, p. 4812, Nov. 2018. Number: 1 Publisher: Nature Publishing Group.

[17] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, "Variational quantum eigensolver with fewer qubits," *Physical Review Research*, vol. 1, p. 023025, Sept. 2019. Publisher: American Physical Society.

[18] D. Wang, O. Higgott, and S. Brierley, "Accelerated Variational Quantum Eigensolver," *Physical Review Letters*, vol. 122, p. 140504, Apr. 2019. Publisher: American Physical Society.

[19] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," *arXiv:1411.4028 [quant-ph]*, Nov. 2014. arXiv: 1411.4028.

[20] R. LaRose, A. Tikku, O'Neel-Judy, L. Cincio, and P. J. Coles, "Variational quantum state diagonalization," *npj Quantum Information*, vol. 5, pp. 1–10, June 2019. Number: 1 Publisher: Nature Publishing Group.

[21] E. Campos, A. Nasrallah, and J. Biamonte, "Abrupt Transitions in Variational Quantum Circuit Training," *Physical Review A*, vol. 103, p. 032607, Mar. 2021. arXiv: 2010.09720.

[22] M. Cerezo and P. J. Coles, "Higher Order Derivatives of Quantum Neural Networks with Barren Plateaus," Aug. 2020.

[23] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, "Effect of barren plateaus on gradient-free optimization," *arXiv:2011.12245 [quant-ph, stat]*, Nov. 2020. arXiv: 2011.12245.

[24] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, "Barren plateaus preclude learning scramblers," *Physical Review Letters*, vol. 126, p. 190501, May 2021. arXiv: 2009.14808.

[25] P. Hayden and J. Preskill, "Black holes as mirrors: quantum information in random subsystems," *Journal of High Energy Physics*, vol. 2007, pp. 120–120, Sept. 2007. Publisher: Springer Science and Business Media LLC.

[26] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, "Noise-Induced Barren Plateaus in Variational Quantum Algorithms," *arXiv:2007.14384 [quant-ph]*, Feb. 2021. arXiv: 2007.14384.

[27] C. Bravo-Prieto, D. García-Martín, and J. I. Latorre, "Quantum singular value decomposer," *Physical Review A*, vol. 101, p. 062310, June 2020. Publisher: American Physical Society.

[28] D. Gottesman and I. Chuang, "Quantum Digital Signatures," *arXiv:quant-ph/0105032*, Nov. 2001. arXiv: quant-ph/0105032.

[29] X. Xu, S. C. Benjamin, and X. Yuan, "Variational Circuit Compiler for Quantum Error Correction," *Physical Review Applied*, vol. 15, p. 034068, Mar. 2021. Publisher: American Physical Society.

[30] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, "Variational quantum algorithms for nonlinear problems," *Physical Review A*, vol. 101, p. 010301, Jan. 2020. Publisher: American Physical Society.

[31] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational Quantum Linear Solver," *arXiv:1909.05820 [quant-ph]*, June 2020. arXiv: 1909.05820.

[32] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, "Variational Quantum Simulation of General Processes," *Physical Review Letters*, vol. 125, p. 010501, June 2020. Publisher: American Physical Society.

[33] B. Commeau, M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, "Variational Hamiltonian Diagonalization for Dynamical Quantum Simulation," *arXiv:2009.02559 [quant-ph]*, Sept. 2020. arXiv: 2009.02559.

[34] C. Cîrstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, "Variational fast forwarding for quantum simulation beyond the coherence time," *npj Quantum Information*, vol. 6, pp. 1–10, Sept. 2020.

[35] J. L. Beckey, M. Cerezo, A. Sone, and P. J. Coles, "Variational Quantum Algorithm for Estimating the Quantum Fisher Information," *arXiv:2010.10488 [physics, physics:quant-ph]*, Oct. 2020. arXiv: 2010.10488.

[36] Z. Ma, P. Gokhale, T.-X. Zheng, S. Zhou, X. Yu, L. Jiang, P. Maurer, and F. T. Chong, "Adaptive Circuit Learning for Quantum Metrology," *arXiv:2010.08702 [quant-ph]*, Oct. 2020. arXiv: 2010.08702.

[37] B. Koczor, S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin, "Variational-state quantum metrology," *New Journal of Physics*, vol. 22, p. 083038, Aug. 2020. Publisher: IOP Publishing.

[38] M. Cerezo, A. Poremba, L. Cincio, and P. J. Coles, "Variational Quantum Fidelity Estimation," *Quantum*, vol. 4, p. 248, Mar. 2020. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[39] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, "Training deep quantum neural networks," *Nature Communications*, vol. 11, p. 808, Feb. 2020.

[40] B. Coyle, D. Mills, V. Danos, and E. Kashefi, "The Born supremacy: quantum advantage and training of an Ising Born machine," *npj Quantum Information*, vol. 6, pp. 1–11, July 2020.

[41] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, p. 032308, Mar. 2020. Publisher: American Physical Society.

[42] J. Carolan, M. Mohseni, J. P. Olson, M. Prabhu, C. Chen, D. Bunandar, M. Y. Niu, N. C. Harris, F. N. C. Wong, M. Hochberg, S. Lloyd, and D. Englund, "Variational quantum unsampling on a quantum photonic processor," *Nature Physics*, vol. 16, pp. 322–327, Mar. 2020.

[43] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, "Theory of variational quantum simulation," *Quantum*, vol. 3, p. 191, Oct. 2019. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[44] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, "Variational Quantum Factoring," in *Quantum Technology and Optimization Problems* (S. Feld and C. Linnhoff-Popien, eds.), Lecture Notes in Computer Science, (Cham), pp. 74–85, Springer International Publishing, 2019.

[45] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, "Variational ansatz-based quantum simulation of imaginary time evolution," *npj Quantum Information*, vol. 5, pp. 1–6, Sept. 2019.

[46] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, "Quantum-assisted quantum compiling," *Quantum*, vol. 3, p. 140, May 2019. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[47] K. Heya, K. M. Nakanishi, K. Mitarai, and K. Fujii, "Subspace Variational Quantum Simulator," *arXiv:1904.08566 [quant-ph]*, Apr. 2019. arXiv: 1904.08566.

[48] R. Kaubruegger, P. Silvi, C. Kokail, R. van Bijnen, A. M. Rey, J. Ye, A. M. Kaufman, and P. Zoller, "Variational Spin-Squeezing Algorithms on Programmable Quantum Sensors," *Physical Review Letters*, vol. 123, p. 260505, Dec. 2019. Publisher: American Physical Society.

[49] A. Arrasmith, L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles, "Variational consistent histories as a hybrid algorithm for quantum foundations," *Nature Communications*, vol. 10, p. 3438, July 2019.

[50] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, pp. 1273–1278, Dec. 2019.

[51] J. Romero and A. Aspuru-Guzik, "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions," *arXiv:1901.00848 [quant-ph]*, Jan. 2019. arXiv: 1901.00848.

[52] G. Verdon, M. Broughton, and J. Biamonte, "A quantum algorithm to train neural networks using low-depth circuits," *arXiv:1712.05304 [cond-mat, physics:quant-ph]*, Aug. 2019. arXiv: 1712.05304.

[53] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, pp. 209–212, Mar. 2019.

[54] M. Schuld and N. Killoran, "Quantum Machine Learning in Feature Hilbert Spaces," *Physical Review Letters*, vol. 122, p. 040504, Feb. 2019. Publisher: American Physical Society.

[55] A. Garcia-Saez and J. I. Latorre, "Addressing hard classical problems with Adiabatically Assisted Variational Quantum Eigensolvers," *arXiv:1806.02287 [cond-mat, physics:quant-ph]*, June 2018. arXiv: 1806.02287.

[56] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii, "Variational Quantum Gate Optimization," *arXiv:1810.12745 [quant-ph]*, Oct. 2018. arXiv: 1810.12745.

[57] E. Farhi and H. Neven, "Classification with Quantum Neural Networks on Near Term Processors," *arXiv:1802.06002 [quant-ph]*, Aug. 2018. arXiv: 1802.06002.

[58] J.-G. Liu and L. Wang, "Differentiable learning of quantum circuit Born machines," *Physical Review A*, vol. 98, p. 062324, Dec. 2018. Publisher: American Physical Society.

[59] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, p. 032309, Sept. 2018. Publisher: American Physical Society.

[60] Y. Li and S. C. Benjamin, "Efficient Variational Quantum Simulator Incorporating Active Error Minimization," *Physical Review X*, vol. 7, p. 021050, June 2017. Publisher: American Physical Society.

[61] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology*, vol. 2, p. 045001, Aug. 2017. Publisher: IOP Publishing.

[62] P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, "QVECTOR: an algorithm for device-tailored quantum error correction," *arXiv:1711.02249 [quant-ph]*, Nov. 2017. arXiv: 1711.02249.

[63] C. Y.-Y. Lin and Y. Zhu, "Performance of QAOA on Typical Instances of Constraint Satisfaction Problems with Bounded Degree," *arXiv:1601.01744 [quant-ph]*, Jan. 2016. arXiv: 1601.01744.