Master's thesis
Master's Programme in Data Science

How the Barren Plateau Problem Influences Quantum Machine Learning

Leo Becker

May 21, 2025

Supervisor(s): MSc Ilmo Salmenperä
Professor Jukka Nurminen

Examiner(s): Professor Jukka Nurminen
Dr. Arianne Meijer-van de Griend

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Leo Becker

Työn nimi — Arbetets titel — Title

How the Barren Plateau Problem Influences Quantum Machine Learning

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	May 21, 2025	31

Tiivistelmä — Referat — Abstract

While quantum computers can give exponential advantages over classical computers, the difficulties in constructing quantum computers have led to small and erroneous devices which are unable to realize the theoretical advantage for useful problems. However, there are some results which show that current quantum computers can outperform classical devices on some toy problems. This has led to the search of an algorithm that could give exponential advantages on small and erroneous near-term devices.

Variational quantum algorithms (VQA) and quantum kernel methods (QKM) have been considered as potential candidates for near-term devices. Unfortunately, these methods often suffer from a barren plateau (BP) problem, where the values, which we evaluate using quantum computers, get exponentially smaller as qubits are added. This is an issue, because quantum mechanics limit us to only have access to samples, which we use to approximate the values of interest. When approximating a value with samples, we need to take more samples when we want to increase accuracy. As we need to keep the error, that comes from sampling, smaller than the value of interest, the BP problem leads to the need to take exponentially many samples. This diminishes the advantage that we gain from using quantum computers, because taking exponentially many shots leads to exponential scaling.

In this work, we go over research about the BP problem and how we can potentially get quantum advantage while avoiding it, which is not easy because the avoidance of the BP problem often leads to us being able to solve the problem in polynomial time without needing a quantum computer. We will show that the avoidance of the BP problem and polynomial classical computability greatly alters the way that VQAs and QKMs operate. Because of this we will also look at some results about VQAs and QKMs being viable for near-term devices. We find out that these results do not lead to current approaches of VQAs and QKMs being inherently suitable for near-term devices.

ACM Computing Classification System (CCS):
Computer systems organization → Architectures → Other architectures → Quantum Computing
Computing methologies → Machine learning → Machine learning algorithms

Avainsanat — Nyckelord — Keywords

Quantum Computing, Machine Learning, Barren Plateau

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

Quantum computers hold a promise of being able to solve specific problems in polynomial time, which take exponential time on classical computers [31]. Unfortunately, the development of quantum computers has proven to be difficult, and the promise of practical quantum advantage has yet to materialize. However, current devices have been shown to outperform a classical computers on theoretical problems [33]. This has led to a search for algorithms that could give practical quantum advantage on small and erroneous near-term devices.

One group of algorithms which has been proposed for these near-term devices is that of Variational Quantum Algorithms (VQA) [18, 5, 9, 30]. These algorithms consist of two parts: a loss function which is evaluated with the help of a quantum computer, and a classical optimizer which optimizes said loss function. As we are interested in the capabilities of quantum computers, we mainly focus on the loss function that is calculated with the help of a quantum computer. This loss function has a general form, which is the equal to kernel functions in Quantum Kernel Methods (QKMs) [35], which are also proposed for near-term devices[15]. Therefore, in this work we group quantum kernels into the group of VQAs.

The loss functions of VQAs depend on values that are present in the internal state of a quantum computer, but cannot be directly accessed due to quantum mechanics. Instead we are limited to probabilistic measurements that have probabilities corresponding to the internal values of interest. This means that we need to do multiple measurements to collect samples, which are then used to approximate the values that we are interested in. For this reason, the loss function can only be accessed through an approximation, which has an accuracy depending on the amount of samples.

The need to sample the loss function was observed to be problematic, because many models have gradients, which get exponentially smaller in the amount of qubits [20]. This leads to programs with exponential runtime, because we need to take exponentially many samples in order to keep the statistical uncertainty smaller than the gradient. If the statistical uncertainty would be bigger than the value of the gradient, we would effectively use random values as gradients. It was show that this problem, named the Barren Plateau (BP) problem, is equivalent to exponential concentration [2], meaning

that the loss functions, or in the case of QKMs the kernel functions, give results which only differ by exponentially small values. To be able to distinguish between different input parameters, we are forced to take exponentially many samples which leads to exponential runtime.

As the goal is to run algorithms in polynomial time, we want to avoid the exponential scaling that comes from the BP problem. This has led to a lot of research into BP free models, which has accumulated to a collection of models that are analytically proven to be BP free. An inspection of these models has led to the realization that they are to a high degree computable in polynomial time on classical computers [6]. Because polynomial classical computability would go against our goal of running something that cannot be done in polynomial time on classical computers, we want to avoid it. For convenience, we define a VQA to be classically simulatable if it can be computed in polynomial time on classical computers.

In this work, we focus on figuring out how avoiding the BP problem and classical simulatability effects VQAs and QKMs. For this we look at current approaches and list out cases in which these allow for BP free models that are not classically simulatable. Firstly, different models need different classical simulation methods, which are often found with the help of analytical proofs that show that a given model is BP free. It could be that a model is observed to be BP free, but we are unable to find an analytical proof that allows classical simulation. In the other cases, in which we have analytical proofs which allows some classical simulatability, all the avenues are focused on models that have some specific parameter values that cannot be classically simulated. In order to use these non-simulatable points, we have to restrict the model parameter to these hard to simulate parameters, which leads to discrete parameters.

As the avoidance of the BP problem and classical simulatability has a profound effect on the form of viable VQAs, we also revisit old results about the suitability of VQAs to be run on near-term devices. We find out that, when taking into account the BP problem, these results are not viable for current approaches. Since we can not find any reason for current VQA approaches being suitable for near-term devices, we would like the research community to stop treating them as being suitable for such devices. With this the main contributions of this work are the following:

- We go over the BP problem and its connection to classical simulatability.
- We list out the cases in which current VQA approaches can lead to exponential quantum advantages.
- We argue that current VQA approaches are not suitable for near-term devices.

The structure of this work is as follows. Chapter 2 goes over quantum computing

theory and introduces the barren plateau problem. Chapter 3 continues by going over the classical simulatability of barren plateau free VQAs. In Chapter 4 we can use the knowledge about classical simulatability of VQAs to list out avenues which could lead to current VQA approaches that give exponential quantum advantages. After we have a good understanding of the current VQA landscape, we can continue chapter 4 with a discussion about the suitability of VQAs for near-term devices. Lastly chapter 5 concludes the work.

2. Background

It has been shown that quantum computers can solve specific problems in polynomial time, while classical computers need exponential time for the same problem [31]. This would lead to the capability to solve some problems that are unfeasible to solve on classical computers due to exponential scaling which would leading to infeasible resource needs, like needing more bits than there are atoms in the known universe. A famous example of such an algorithm is Shor's algorithm that can do integer factoring and find discrete logarithms in polynomial time [31].

In theory, even small quantum computers could be useful, as the classical resources needed to simulate quantum computer scale as 2^n, where n is the amount of qubits. This means that even simulating 60 qubits using 64 bit complex numbers, would lead to a memory need of over nine exabytes. For comparison this is around 80 times more memory than the storage capacity of LUMI, a supercomputer that was made available for customers at the end of 2022 [24].

Inspired by the scaling advantages, many companies have successfully built quantum computers. Unfortunately, these devices have low qubit counts and suffer from errors, leading to them being unable to run algorithms like Shor's algorithm at useful scales. On the other hand, it has been shown, that current quantum computers are able to compute problems, such as random circuit sampling [33], that are infeasible for current classical algorithms. The reason that these algorithms have not led to a quantum breakthrough, is that there are no use cases for them. This means that the quest for practical quantum advantage is still ongoing.

This brings us to Variational Quantum Algorithms (VQA) that have been regarded as good candidates for reaching practical quantum advantage on small and erroneous quantum computers [18, 5, 9, 30]. The idea of these algorithms is to use a quantum computer to assist in evaluating a loss functions that have access to the exponential space of quantum computers, while using classical optimization methods to optimize these loss functions. This leads to a quantum-classical optimization loop as depicted in Figure 2.1, where the only difference to classical optimization loops is that a quantum computer is used in the evaluation of the loss function.

Examples of VQAs are: Variational Quantum Classifiers (VQC) for classification

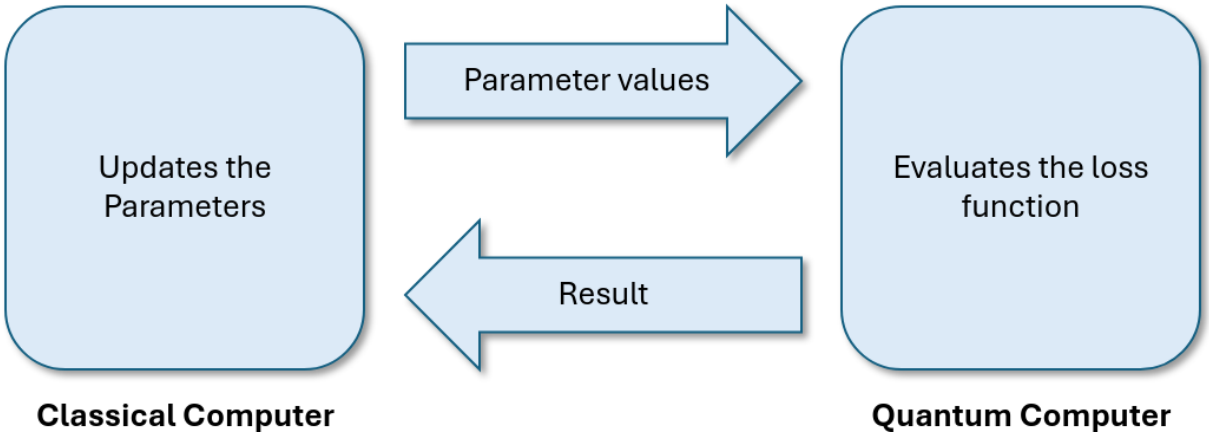

Figure 2.1: In a quantum-classical optimization loop, a quantum computer is used to evaluate a loss function and a classical computer is used to compute new parameters based on the evaluations of the loss function.

tasks [29]; Variational Quantum Eigensolvers (VQE) for approximating the smallest eigenvalue of a matrix, which can be used to find the ground state energy of molecules [25]; Quantum Approximate Optimization Algorithms (QAOA) for approximating solutions for combinatorial problems [10]; Quantum Convolutional Neural Networks (QCNN) for recognizing quantum states [8].

While the exact form of a VQA loss function is use case specific, we can discuss them all at once by using a general function which can be tailored to every use case by choosing suitable parameters. By choosing suitable parameters, we can use this general function to also describe all kernel functions of Quantum Kernel Methods (QKMs) [35], meaning that we can go over VQAs and QKMs at the same time. As most of the text treats VQAs and QKMs the same way, we will assume that QKMs are included in VQAs if these are not explicitly separated. For generality we will use the word model to describe the part* of the algorithms which is calculated with the help of a quantum computer. In order to introduce the general mode which we use throughout this work, we will first go over some theory about quantum computing.

2.1 Quantum Computing Theory

The inner state of a quantum computer with n qubits can be represented with a unit vector $|\psi\rangle$ that contains 2^n complex numbers. To operate on the internal state, we use unitary operators U, which can be represented with $2^n \times 2^n$ unitary matrices. When a unitary operator is used, an internal state $|\psi\rangle_0$ is evolves to another internal state $U|\psi\rangle_0 = |\psi\rangle_1$.

In order to benefit from the computation, we need to read out the result. Unfortu-

*This part can be a loss function or kernel function.

nately, we can only do probabilistic measurements in an orthonormal basis of the complex vector space $\mathbb{C}^n$. The measurement results correspond to the orthonormal basis elements $|x_i\rangle$, which have measurement probabilities given by $|\langle\psi|x_i\rangle|^2$, where $|\psi\rangle$ is the internal state before the measurement operation. In this work, we will use the orthonormal basis $\{e_i\}$, where the index i starts from 0, and we use the notation $e_i \equiv |i\rangle$.

A measurement with the result $|i\rangle$ changes the internal state to $|i\rangle$, meaning that measurements alter the internal state, and are therefore destructive. As the measurement operator is probabilistic, we often want to measure an internal states multiple times. This leads to the need to rerun the whole program multiple times, because we need to recreate the state for every measurement. In the quantum computing literature these reruns are often called shots.

While all measurements follow the same rules, we can use them in different ways. In the case that the result of a computation is encoded into a basis state $|i\rangle$, which has a high probability to be measured, when compared to basis states which are not correct results, we can get the result with only a couple of shots. Another option is that we are interested in the values $|\langle\psi|i\rangle|^2$, which we can approximate by taking multiple shots to approximate the probability that the state $|i\rangle$ is measured. As we can only take a limited amount of shots N, this approximation has in general and error of size $1/\sqrt{N}$, which can in some cases improved up to $1/N$ [11]. This statistical uncertainty that comes from having a limited amount of shots is called shot noise. As the word noise can mean different different things depending on the context, we want to make it clear that shot noise has nothing to do with the errors that occur because of imperfections in quantum computers.

We can use the ability to approximate $|\langle\psi|i\rangle|^2$ to approximate the general model of VQAs that is of the form $\langle\psi| O |\psi\rangle$, where O is an hermitian matrix which is often called an observable. In order to build up to this, we will start with the ability to approximate values of the form $\langle\psi| D |\psi\rangle$, where D is a Kronecker product of

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

matrices, meaning that it is a diagonal matrix with diagonal elements from the set $\{1, -1\}$. This can be done by averaging over 1 and -1 according to the diagonal matrix elements

$D_{ii} \in \{1, -1\}$:

$$\begin{aligned}\langle\psi| D |\psi\rangle &= \langle\psi| \left(D_{0,0} |0\rangle \langle 0| + D_{1,1} |1\rangle \langle 1| + ... + D_{2^n-1,2^n-1} |2^n - 1\rangle \langle 2^n - 1|\right) |\psi\rangle \\ &= D_{00} \langle\psi|0\rangle \langle 0|\psi\rangle + D_{11} \langle\psi|1\rangle \langle 1|\psi\rangle + ... + D_{2^n-1,2^n-1} \langle\psi|2^n - 1\rangle \langle 2^n - 1|\psi\rangle \\ &= \sum_{i=0}^{2^n-1} D_{ii} \langle\psi|i\rangle \langle i|\psi\rangle \\ &= \sum_{i=0}^{2^n-1} D_{ii} |\langle\psi|i\rangle|^2.\end{aligned} \tag{2.2}$$

By using the unitary operators

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \tag{2.3}$$

we can transform

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.4}$$

operations into Z by using the rules $X = HZH$ and $Y = SHZHS^\dagger$. Because H and S are unitary operators, which we can apply on quantum computers, we can measure

$$\langle\psi| M_{XY} |\psi\rangle = \langle\psi| U^\dagger Z U |\psi\rangle, \tag{2.5}$$

where M_{XY} is X or Y and U correspondingly H or $HS^\dagger$. By using the property $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, we realize that we can apply separate H and $S^\dagger$ operators to measure all Kronecker products of I, Z, X, and Y. This means that we can use quantum computers to measure

$$\langle\psi| P |\psi\rangle = \langle\psi| U^\dagger D U |\psi\rangle, \tag{2.6}$$

where P is a Pauli string, meaning a Kronecker product of I, Z, X, and Y, and U is an Kronecker product consisting of matrices I, H, and $HS^\dagger$.

Pauli strings form a basis for complex $2^n \times 2^n$ matrices meaning that we can decompose any $2^n \times 2^n$ matrix into a sum of Pauli strings with coefficients [17]. Furthermore, in the case of a hermitian observable (matrix), these coefficients are real numbers [17]. This leads to the following decomposition:

$$\begin{aligned}\langle\psi| O |\psi\rangle &= \langle\psi| \left(\sum_i a_i P_i\right) |\psi\rangle \\ &= \sum_i a_i \langle\psi| P_i |\psi\rangle,\end{aligned} \tag{2.7}$$

where O is a hermitian observable, and a_i:s are real coefficients. As the value $\langle\psi| O |\psi\rangle$ depends on $\langle\psi| P |\psi\rangle$ which depends on $\langle\psi| D |\psi\rangle$ which depends on $|\langle\psi|i\rangle|^2$, and $|\langle\psi|i\rangle|^2$

is an approximation, the final value is also an approximation. It is also of note, that we need to evaluate every $\langle\psi| P_i |\psi\rangle$ separately, meaning that we need to limit the amount of P_i:s to be polynomial in order to avoid exponential scaling.

VQAs and QKMs use quantum computers to evaluate models of the form $\langle\psi| U^\dagger(\theta)OU(\theta) |\psi\rangle$, where $|\psi\rangle$ is often called the initial state, and $U(\theta)$ is a parametrized operation called ansatz, that is run on a quantum computer. This forms the model that we are calculating with the help of a quantum computer:

$$\mathcal{L}_\theta(\rho, O) = \langle\psi| U^\dagger(\theta)OU(\theta) |\psi\rangle = \mathrm{tr}[U(\theta)\rho U^\dagger(\theta)O] = \mathrm{tr}[U^\dagger(\theta)OU(\theta)\rho], \tag{2.8}$$

where $\rho = |\psi\rangle \langle\psi|$ is the initial state in its density matrix representation. The second equality comes from the property that the trace of an outer product is equal to the inner product, and the last one from selecting the vectors for the inner product to outer product transformation differently.

For specific use cases the model (2.8) uses different initial states ρ, observables O, and ansätze $U(\theta)$. The initial state is often the $|0\rangle \langle 0|$ state, because it is the starting state for computations on many devices. But we are not limited to the starting state of the device, because the initial state of the model can be prepared by using a unitary operator to evolve the starting state of the device into the starting state of the algorithm. As the initial state ρ can come from an operator and $U(\theta)$ is also an operator, it is ambiguous which part of the model belongs to ρ and which to $U(\theta)$. While there is no common rule, in this work we assume that the initial state has no trainable parameters, and that in the case of kernel functions the data point pairs (x_i, x_j) are contained in the ansatz via the parameters θ.

As we often train at least some of the parameters θ in $U(\theta)$, we need access to the gradients of the model (2.8). These gradients can be approximated using samples around the current θ or by evaluating the partial derivatives with the parameter shift rule:

$$\frac{\partial\theta\mathcal{L}_\theta(\rho, O)}{\partial} = \frac{1}{2}\left(\mathcal{L}_{\theta;\theta_i+\pi/2}(\rho, O) - \mathcal{L}_{\theta;\theta_i-\pi/2}(\rho, O)\right), \tag{2.9}$$

where $\theta; \theta_i + a$ means that we add a to θ_i and leave all other elements of θ unchanged [23]. While the parameter shift rule gives exact gradients in theory, we still end up with approximations that suffer from shot noise. This is because of the dependency on the loss function, which we can only approximate via samples that give us approximations of $|\langle\psi|i\rangle|^2$.

2.2 The Barren Plateau Problem

Unfortunately, it was soon found out that many VQA models suffer from a problem, where the variance of the gradient gets exponentially smaller as qubits are added [20]. This prob-

lem, named the barren plateau (BP) problem, has later been shown to be equivalent to exponential concentration and exponential narrowness of minima [2]. Exponential concentration means that the variance of the loss function is exponentially small, which leads to functions being effectively constant, if we do not have exponential accuracy. Exponential narrowness of the minima means that if there is an area where the function varies in a meaningful way, the area in the parameter space where it varies, gets exponentially smaller when qubits are added.

VQA models are defined in such a way that all off the parameters are cyclic [5], this means that, because we need to always go the same amount up and down to end up at the same point after a cycle, the average of all gradients is 0. If we assume that the variance of the gradient is exponentially small, the average difference between data points is exponentially small. This means that the gradient is, on average, exponentially close to 0. This is a problem, because if the gradient is smaller than the shot noise, we are walking around randomly, because the contribution of the shot noise is bigger than that of the gradient. As we need to keep the shot noise smaller than the gradient, the BP problem forces us to take exponentially many shots, which leads to exponential scaling when doing model training. In the case of kernel function evaluation and model inference, the BP problem also leads to exponential cost, because exponential concentration forces us to take exponentially many shots to distinguish between different answers.

As the BP problem cancels out the advantage which is gained from using quantum computers, there has been a lot of research into avoiding it. This has led to many combinations of ρ, O, and $U(\theta)$ that have been proven to be BP free [6]. These proofs are mainly based on finding components of the loss function that do not suffer from the BP problem [6]. With components we mean independent values that contribute to the sum of the loss function. We can see an example of a decomposition in Equation (2.7), where the function $\langle\psi| O |\psi\rangle$ is split into components of the form $a_i \langle\psi| P_i |\psi\rangle$.

As the parameters θ change the behavior of the function, the behavior of the components also changes based on θ. This can lead to situations where the components that lead to the absence of the BP problem change based on the parameters of the model. It has been summarized that in most cases the proofs find polynomially many BP free components which live in proper of effective polynomial subspaces [6]. Some BP free models bring an exception to this by having specific parameter values, for which exponentially many components come together to create a meaningful contribution even though the individual components are exponentially small, and therefore do not lead to meaningful contributions individually [6].

When we say that a component or model lives in a proper polynomial subspace, we mean that they fulfill both of the following conditions:

- Only a polynomial amount of values in the initial state ρ and observable O contribute

to the result.

- Given these values, the computation only depends on a polynomially sized subset of the exponential space available for quantum computers.

Effective polynomial subspaces have a more relaxed definition in that we allow that the values outside of the proper polynomial subspace are non-zero with the rule that they are exponentially small and together only amount to an exponentially small contribution. As the values outside of the proper polynomial subspace amount to only an exponentially small contribution, we cannot measure their contribution with polynomially many shots, and they can therefore be ignored. It has to be noted that, while the components that live in polynomial subspaces can lead to the absence of the BP problem, the existence of such components is not an sufficient condition for the absence of the BP problem [6]. A trivial example of this is a model without parameters.

As proper polynomial subspaces only use polynomially many values from an exponential space, the amount of values that stays outside of this subspace is exponentially large. This means that when we have an effective polynomial subspace, we can have exponentially many exponentially small values outside of the proper polynomial subspace. This can lead to some specific parameter values, for which these exponentially many exponentially small values come together to create an meaningful contributions that can be measured in polynomial time. This means that for these specific parameter values the model depends on an exponential space, even in the case of polynomially many shots, which means that the model does not live in an effective (or proper) polynomial subspace. For these models, we say that the model lives with high probability given θ in an effective polynomial subspace. Current analytically proven BP free models, which do not live in proper or effective polynomial subspaces, are these kind of models, and most notably contain quantum convolutional neural networks and cases where specific small angle initialization strategies are used [6].

In addition to the proven BP free models, there are cases where the absence of the BP problem is suspected based based on numerical calculations [27, 9]. In these cases, there is no analytical proof that the models live in proper or effective polynomial subspaces, but it is still sensible to believe, that if they are BP free, they follow these same rules as the proven BP free models [6]. To get intuition to why this is we can look at the the decomposition in Equation (2.7) where we see that the quantum computer is used to measure values of the form $\langle\psi| P_i |\psi\rangle$ which are inner products between two exponentially big unit vectors $|\psi\rangle$ and $P_i |\psi\rangle$. As the norm of unit vectors is 1 and the value is distributed along exponentially many fields, the absolute values of individual values in the vector are mostly exponentially small. Then when we take an inner product between two such vectors, we sum over values which are on average products of two exponentially

small values. This means that we mostly sum together exponentially small values which need to work together to give meaningful contributions.

3. Classical Simulatability

As the benefit from using quantum computers comes from the ability to use exponentially many values, the realization that proven BP free models mostly live in polynomial subspaces brings into question if a quantum computer is needed for these tasks. Because the goal is to use quantum computers on problems which take exponential time on classical computers, we focus this chapter on finding cases in which we avoid both the BP problem and polynomial classical simulation. As the models are originally created to be run on a quantum computer, we call a model classically simulatable, if it can be calculated in polynomial time without the use of a quantum computer.

It was found out that for common analytically proven BP free method, the research community has been able to find a polynomially scaling classical algorithms, which characterizes the action of the ansatz when the model lives in a polynomial subspace [6]. For this to lead to classical simulatability of a model, we also need to know the polynomially many values from the initial state and observable which interact with the ansatz. Because there are cases in which we need a quantum computer to figure out these values the initial state and observable can lead to models not being classically simulatable [6]. In addition, there are also models which are BP free, but do not live in polynomial subspaces for each model parameter θ. This means that we have models, which have specific parameter values that we cannot classically simulate. This narrows the exploration of classical simulatability of current VQA approaches to the initial state, observable, and non-simulatable parameter values, because only these can allow BP free models which are not classically simulatable.

While the components that live in polynomial subspace are model specific and vary greatly in how they are represented, in general the components come from splitting the initial state or observable [6]. To make the idea of classical simulatability more concrete, we will use an example decomposition along observable and initial state. Because the observable is hermitian, we can decompose it into $\sum_i a_i H_i^O$, where $\{H_i^O\}$ is a basis for

hermitian matrices and $\{a_i\} \in \mathbb{R}$. When applying this to the model (2.8) we get

$$\begin{aligned}\mathcal{L}_\theta(\rho, O) &= \operatorname{tr}[U^\dagger(\theta) O U(\theta) \rho] \\ &= \operatorname{tr}[U^\dagger(\theta)(\sum_i a_i H_i^O) U(\theta) \rho] \\ &= \sum_i a_i \operatorname{tr}[U^\dagger(\theta) H_i^O U(\theta) \rho].\end{aligned} \tag{3.1}$$

By using the property that the inner product equals to the trace of the outer product do the trick with inner outer product, we can swap the places of H_i^O:s and ρ:

$$\begin{aligned}\mathcal{L}_\theta(\rho, O) &= \sum_i a_i \operatorname{tr}[U^\dagger(\theta) H_i^O U(\theta) \rho] \\ &= \sum_i a_i \langle\psi| U^\dagger(\theta) H_i^O U(\theta) |\psi\rangle \\ &= \sum_i a_i \operatorname{tr}[U(\theta) \rho U^\dagger(\theta) H_i^O]\end{aligned} \tag{3.2}$$

Now we can decompose the model further by using the property that the when the initial state ρ is in its density matrix form, it is hermitian*. This leaves us with

$$\begin{aligned}\mathcal{L}_\theta(\rho, O) &= \sum_i a_i \operatorname{tr}[U(\theta) \rho U^\dagger(\theta) H_i^O] \\ &= \sum_i a_i \operatorname{tr}[U(\theta)(\sum_j b_j H_j^\rho) U^\dagger(\theta) H_i^O] \\ &= \sum_i \sum_j a_i b_j \operatorname{tr}[U(\theta) H_j^\rho U^\dagger(\theta) H_i^O],\end{aligned} \tag{3.3}$$

where the initial state is decomposed into $\sum_j b_j H_j^\rho$ where $\{H_j^\rho\}$ is a basis for hermitian matrices and $\{b_i\} \in \mathbb{R}$.

If we assume that the decomposed model is proven to be BP free and lives in a polynomial subspace, we can evaluate its value using only polynomially many components. Furthermore, in all inspected cases it has been possible to figure out the interaction of the ansatz $U(\theta)$ using classical methods [6]. In our example decomposition this means that we can evaluate the traces $\operatorname{tr}[U(\theta) H_j^\rho U^\dagger(\theta) H_i^O]$ without using quantum computers. With this we only need to figure out the values a_i and b_j for the polynomially many components which have meaningful contributions to the result. While we can sometimes find these values classically, sometimes we need a quantum computer to figure out these values [6]. This means that we can have models for which a quantum computer is only needed for figuring out some values which describe the observable and initial state.

The rest of this chapter is structured as follows: Firstly we will discuss the effects that non-simulatable points have on classical simulations, as these cannot be classically simulated. Then we will go more in depth about the initial state and observable, because these can lead to the need to use a quantum computer which in turn means that there exists a possibility of quantum advantage.

*$\rho = |\psi\rangle\langle\psi| = (|\psi\rangle\langle\psi|)^\dagger$

3.1 Effects of non-simulatable points

For the models that live in effective polynomial subspaces only with high probabilities given θ, it is crucial to know if the parameter values that do not live in polynomial subspaces are needed. This is because if these points are not needed, we can solely rely on classical simulation, but if they are need then the model requires a quantum computer.

As mentioned earlier, the points which are not characterizable using polynomial subspaces depend on exponentially many exponentially small values that work in unison to create measurable contributions [6]. For this to happen, the exponential space has to be used very deliberately, and in the found cases, at least some parameters need to hit specific parameter values [6, 1]. This means that we have to use discrete parameters, which is similar to for example Shor's algorithm, which also uses a quantum computer to run a parametrized operation with discrete parameters.

In the cases in which we train VQAs with continuous parameters, we can easily start in a spot that is simulatable. After this it is of interest to know, if optimization steps can lead to the specific non-simulatable parameter values. For this we argue that, as the optimization steps consist of floating point numbers that go through multiple calculations, it is in practice so unlikely to hit the exact values needed, that these points can be ignored. There are also practical results, which show that even in the presence of non-simulatable points quantum convolutional neural networks can be successfully trained using classical simulation methods [3].

With quantum kernels, that use continuous variables, we get into a similar situation, as it is unlikely to have data points that hit exactly the non-simulatable points. Where the situation gets more interesting is in the use of discrete variables, as we could map these to the exact parameter values that are non-simulatable. This leaves us with a model that does not suffer from the BP problem and is not classically simulatable. One example of such a model can be created by creating a kernel that is based on the discrete logarithm problem, which is hard to calculate classically but can be solved in polynomial time using Shor's algorithm [19]. While this result is meaningful in theory, the kernel is created to separate data with very specific rules, which mean that no machine learning is needed and we could just simply run Shor's algorithm on individual data points to figure out the discrete logarithms and group the data based on these.

Similarly to kernel methods, we can also use discrete parameters in VQAs which undergo a quantum-classical optimization loop. There has been research into an ADAPT approach which tries adding operations to a quantum program and keeps the operations that give the best results [13]. For this training method there is a theoretically suitable BP free model which was introduced in Ref [6]. This model, which is BP free and not simulatable classically based on the difficulty of the discrete logarithm problem, has dis-

crete parameters which can correspond to specific individual operations doing nothing. This means that we could start from a point where the parameters correspond to identity operations and use the ADAPT algorithm to pick operations which replace the identity operations with operations that contain some other hard to simulate parameters. This means that we have a trainable BP free model that can not be simulated classically. While this is an avenue where quantum computers can beat classical computers, it remains to be seen if a practical use case with discrete parameters presents itself.

3.2 The Initial State

For the discussion about the initial state ρ, we will split the VQA algorithms into two groups based on how they use the initial state:

- Optimization problems and QKMs where the initial state has no meaning by itself
- Machine learning applications where the initial state corresponds to some data point.

Both of these groups have some limitations for the initial state, because the proofs for the absence of the BP problem only apply when suitable initial states are used. While these limitations differ, a minimum requirement is that the interaction between the initial state and the ansatz gets at most polynomially smaller in relation to the qubit count. In our example decomposition this would mean that the coefficients which correspond to the initial state in the polynomial subspace get at most polynomially smaller when qubits are added. This means that even though the choice of ansatz is sometimes not meaningful, it has to be deliberate in order to avoid the BP problem.

As the initial state does not store information in optimization problems, the initial state is usually simple and therefore classically simulatable. For example, in VQE the initial state is usually the starting state of the device, that usually means the state $|0\rangle\langle 0|$. In the case of QAOA the initial state is the equal superposition over all states, in the unit vector representation this would mean that all the elements in the vector have equal values of $1/\sqrt{2^n}$ [10]. In quantum kernels there can be additional limitations, such as in the fidelity quantum kernel needing the initial state to be equal to the observable [28, 35], but still the initial state is usually $|0\rangle\langle 0|$.

Next we will consider machine learning problems where the initial state corresponds to some data point. These states correspond to some data points either by being quantum data or an encoding of classical data onto a quantum computer. In the case of quantum data sets, we either have some quantum measurements or classical representations of instructions of how to operate a quantum computer to get the desired state [6, 3]. Assuming that we have real quantum data, in that we cannot classically get the result, a quantum

computer is required to gather information about the quantum state. This is usually done via measuring observables or sufficient Pauli classical shadows [14] of individual data points [6, 3]. The rest of this section goes over the case that we have classical data that is encoded into a quantum state.

When encoding classical data to a quantum computer, we commonly start from some simple state, such as the common starting state of quantum device $|0\rangle\langle 0|$, and then apply a parametrized unitary operator $U(x)$ on it, where the parameters x are dictated by the classical data. In order to measure properties about this quantum state, we use simple observables, sufficient Pauli classical shadows, or something comparable, which all use observables for measuring the quantum state [6, 14]. This means that we are interested in measuring functions of the same form as the original model (2.8) and, like in the case of the ansatz, we have to be concerned about the BP problem.

The BP problem is an issue when using a quantum computer to measure the initial state, because of exponential concentration that leads to all data points being indistinguishable from each other. Every data point being effectively the same, would make the machine learning models unusable, as we want to do something based on the data, like for example classification. This leads to the same issue as in the original model, where the model either lives with high probability given x in a polynomial subspace, or suffers from the BP problem and makes the model unusable.

Because the initial state for data encoding is usually the starting state of the device, which is easily classically computable, we do not need a quantum computer to measure the initial state of the data encoding. Additionally, as we find out in the next section, we do not need to use a quantum computer to measure information about the observable, which means that the only place to look for quantum advantage is the encoding unitary $U(x)$. Due to the structure with polynomial subspaces, we can use similar arguments as for the ansatz when talking about continuous variables, more specifically the probability to encountering data points that hit hard to simulate points is practically zero. On the other hand, for discrete variables, we can do the same as in the case of kernel methods in that we choose a model that only lives in polynomial subspaces with a high probability given x and map the classical data deliberately to the parameter values that cannot be simulated in polynomial time on classical computers.

This means that there are two known cases in which we need a quantum computer for the purpose of figuring out the needed constants from the starting state. The first case is that we have quantum data, and the second is that we encode discrete data to parameter values which lead to encoding operations that are hard to simulate classically.

While we assume in this section that the model is BP free, there is also literature that shows that initial states that encode classical data and are not classically simulatable, often lead to the BP problem [36]. This is because encoding methods that are hard to

classically simulate often lead to initial states that do not fulfill the requirements that are given by the proven BP free methods, such as the size of the interaction with the ansatz.

3.3 The Observable

When splitting the VQA methods based on the use of the observable, the roles are mostly reversed in comparison to the initial state:

- In the case of optimization problems the observable represents an encoding of the problem,
- while in QKMs and VQA based machine learning applications the observable has no meaning by itself.

For optimization problems the observable can represent various quantities depending on the optimization problem at hand. The observable can represent a structure of the problem, like a graph when solving a MaxCut problem using QAOA [10], or an encoding of a molecule when using VQE to solve its ground-state energy [25]. Machine learning methods on the other hand use very simplistic observables, like for example variational quantum classifiers and some quantum convolutional neural networks which measure out one qubit, leading to observables of the form $|0\rangle\langle 0|$ [29, 3]. Because kernel methods encode the information into the ansatz, both the observable and initial state can be chosen, which means that this time we group them together with machine learning applications.

Regardless of the complexity of the observable, we already have a decomposition in the Pauli basis, because, as explained in the background section, we need a Pauli decomposition for the ability to measure the loss function on a quantum computer. For the found cases, this decomposition has been sufficient for classical simulation, meaning that no quantum computer is needed for the purpose of learning information about the observable [6, 12]. Still, in theory there are cases in the known simulation algorithms, where one can choose to use a quantum computer to find out information about the initial state or observable [6]. This means that if it is more costly to evaluate the initial state on a quantum computer, we could in some cases choose to evaluate the observable instead. Even though this could lead to cases, where a quantum computer is used to measure the observable, we will say that the quantum advantage comes from the initial state, because the initial state leads to the need to use a quantum computer.

While the observable is not as interesting as the initial state when discussing the classical simulatability of models, it still needs to lead to models, which are BP free. Again the most important requirement is that the interaction between the ansatz and observable is only allowed to vanish polynomially when qubits are added [6]. This is

often done by selecting local observables, meaning that the individual components of the observable, as depicted in (2.7), only depend on fixed amounts of qubits [6]. These local observables are usually combined with ansätze and starting states that are limited in such a manner that any qubit readout effectively only depends on polynomially many fields of the quantum state [6]. As an example shallow hardware efficient ansätze with suitable starting states are limited in size with the goal to make single qubit measurement only depend on logarithmically many qubits [7]. This means that single qubit measurements depend on quantum states of the size $2^{\log(n)} = n$.

4. Discussion

Based on the currently known BP free methods, we can list avenues which allow current VQA approaches to reach exponential quantum advantage:

1. The absence of the BP problem has been correctly assumed, but we lack an analytical proof which would likely lead to the model being classically simulatable in most cases.

2. We train models with discrete parameters which correspond to non-simulatable parameter values.

3. Kernel methods are used for discrete data, where the data is deliberately mapped to non-simulatable parameter values.

4. Discrete classical data is deliberately mapped to non-simulatable parameter values in the initial state.

5. The initial state consists of quantum data.

In the first case the advantage comes from not being able to run the model classically due to not having enough information about the model. This could lead to useful quantum advantage, in the case that a useful model has been experimentally found to be BP free in some region, but there is no analytical proof that would lead to the classical simulatability of the model. In this case the quantum advantage is likely temporary, in the sense that it is likely that we will find an analytical proof for the absence of the BP problem, which then is likely to leads to classical simulatability for at least most of the parameter values θ.

In the second case we exploit the knowledge that we have parameter values which lead to the model not being classically simulatable. While the discrete nature of the parameters greatly changes the way in which we can train the model, we gave an theoretical example, which shows that in theory at least some models with discrete parameters can be trained. While we do not know of any use cases for these kinds of models, this shows that this approach could lead to interesting results.

Crucially, the first and second cases are the only ones in which a quantum-classical optimization loop is needed. In the remaining cases we use a quantum computer to figure out properties about the initial state or observable and can afterwards evaluate the model fully classically. This is a big shift for trainable VQA models, because the quantum-classical optimization loop is often considered as an integral part of VQAs. While the training loop does not need a quantum computer, in practice it could still be useful to keep a quantum computer around for the training. Because the model changes based on the parameter values, the values which we need from the initial state can change based on the parameters. Now in practice it would be sensible to only do the data acquisition when we know that specific parameter values are used as this would lead to less work than evaluating the initial state for all possible parameter values. This means that we would introduce a quantum computer back into the training loop with the role of updating the classical simulation [6].

The second and third case of the list are similar to each other, in that they carefully choose how to encode discrete data onto the quantum computer, in such a manner, that it leads to classically non-simulatable operations. These circuits use the quantum state in a very deliberate manner in order to not suffer from the BP problem while not being classically simulatable. This very careful use of the quantum state makes it so that these algorithms use quantum computers in a similar manner as non-variational algorithms such as Shor's and Grover's algorithm, and indeed all the examples that we could find of these kinds of algorithms are based on the ability to solve the discrete logarithm problem by using Shor's algorithms [19, 6].

In the last case we can either have data from an experiment, or know how to prepare the quantum state in question. When the data comes from an experiment, one has to use a quantum computer to measure the properties that are need to create a classical representation of the state. As we need to make a lot of measurements, the quantum experiment has to be repeatable so that we get a sufficient amount of shots, but afterwards the classical representation can be used for classical simulation [6]. In the case that quantum data is prepared using a quantum computer, we trivially need a quantum computer if we assume that the data is truly quantum in the sense that we cannot simulate it classically.

With this understanding of found cases where VQAs can lead to exponential advantage, we can look back at the idea of them being a good candidate for reaching practical quantum advantage on small and erroneous quantum computers. For this it is crucial to find an use case that gives practical advantage, but currently we can only speculate about the possibility that someone finds one. Another important part is the suitability of VQAs for small and erroneous devices. About this we can have discussions based on old results, which claim that VQAs are suitable for these devices, but do not take into account the

BP problem. In addition, we will go over some benefits that can come from researching the BP problem, but are outside of VQAs. In the end we will also go over some caveats that help to clarify the scope of this work.

4.1 Running on Small and Erroneous Devices

In this section we discuss the common sentiment that VQAs are suitable for small and erroneous devices, which are often referred to as noisy intermediate-scale quantum (NISQ) devices [5, 26]. The ideas of VQAs being suitable for such devices seems to come from ideas that did not take the BP problem into account [22, 21, 30] and some success of running VQAs on very low qubit count devices [34, 25]. As the barren plateau problem shows itself at higher qubit counts as unfavorable scaling, we are interested in knowing if the arguments for VQAs being suitable for NISQ devices still hold when we have to avoid the BP problem while aiming for exponential quantum advantage.

The claims about the error resilience of VQAs can be split into resilience against coherent and incoherent errors [5]. In the case of coherent errors, the used operators have some unknown constant errors. The idea that leads to the idea of VQAs being resilient to this kind of errors stems from the ability to parametrize the ansatz in such a way that the ansatz is of the form $U(\theta + e)$, where e is a vector containing the constant, but unknown errors [22, 5]. Assuming, that we can train continuous parameters θ, this would lead to the ability to train θ to be such that $\theta + e$ leads to the optimal ansatz.

In addition to the theoretical results, there are experimental results that show that very small models can be run in the presence of coherent errors [34]. Unfortunately, when increasing the amount of qubits, we have to take the BP problem into account. As this idea needs trainable parameters, the only case for exponential quantum advantage would be that in which we have correctly observed the absence of the BP problem and lack a proof that leads to classical simulatability. For the remaining cases, we need to use a quantum computer in a very deliberate way, and if we allow a constant error e, the quantum computer is not used in this deliberate way anymore, and we can pick some small e that allows for classical simulatability.

For incoherent errors there are results that show that under some specific errors it is possible that the model moves away from areas of the landscape that contain these errors [21]. There is also a result that shows, that under certain kinds of errors, some models can still find the optimal parameters, even when the value of the loss function is wrong [30]. These results assume that there are trainable parameters, and because we assume incoherent errors, we can add small errors to e to avoid non-simulatable parameter values.

Because the findings about coherent and incoherent errors are based on training

models, they are only applicable in the cases in which we train discrete parameters or we have observed that a model is BP free, but lack an analytical proof that would lead to classical simulatability. Unfortunately, the case with discrete parameters is not applicable for the cases with incoherent errors, because they are model specific with continuous parameters, and for the case with coherent errors, we would need to have coherent errors that play along with the discretization of parameters.

For the cases in which we have correctly observed that a model is BP free, but lack an analytical proof which would lead to classical simulatability, the results mentioned above are also speculation at best because these results are very model specific. In the case of incoherent errors, the results are trivially model specific as they assume a specific kinds of models. In the case of coherent errors, the results appear more general, but they are not, because we need to add parameters, which change the behavior of the model. Depending on the model and errors, the additional parameters can change the behavior of the model in such a way that we suffering from the BP problem.

There are also claims about VQAs being suitable for erroneous hardware due to the ability to implement the operation on quantum computers in a short amount of time and few physical operations [34, 37]. Having an operation that can be implemented in a short program on quantum computers would improve the error resilience, because a big part of the errors come from individual physical operations [32] and decoherence over time [4].

This idea about VQAs leading to short programs seems to come from the freedom that some algorithms have in choosing the initial state and ansatz. For example in the case of VQE and VQC, we can choose these freely. While the choice of initial state and ansatz effects the quality of results, it has been shown with small examples, that selecting an ansatz and initial state that is tailored for the device, instead of the problem, can lead to some success [16]. Unfortunately, these hardware efficient solutions are only BP free when we choose to scale the amount of used qubits in such a way that we do not use all of the given qubits [20, 7]. This leads to models that are mathematically BP free and classically simulatable, but if we would calculate the variance scaling in such a way, that we only take into account the qubits that effect the value of the loss function, the model suffers from the BP problem.

As the examples of models that are BP free and non-simulatable are based on the ability to solve the discrete logarithm problem with Shor's algorithm [19, 6], we do not see any reason to believe that the program sizes of VQAs would be considerably shorter when compared to programs like Shor's algorithm itself, which is not considered to be a short program. The knowledge about the trade-off between classical simulatability and the BP problem can lead to shorter programs on quantum computers, because measuring only the initial state is often shorter than calculating the loss function using a quantum computer [6]. Unfortunately, this does not mean that the resulting program is short, as

for that we would need to assume that the calculation of the loss function would also be short.

As we did not find any results that would show that the current approach for VQAs would be suitable for small erroneous devices, we would appreciate it, if the research community would stop treating VQAs as being suitable for these devices. If one wants to make these kinds of claims, they should have proper reasoning that takes the BP problem into account. We would go as far as to say that common algorithms like Shor's and Grover's, are more error resilient due to the ability to filter out erroneous shots. In Shor's algorithm one can filter out erroneous shots by using the the knowledge that the result is periodic, and in Grover's we are able to confirm classically if a shot gives the correct answer.

4.2 Benefits From Researching The Barren Plateau Problem

While the results shared in this work do not paint a promising future for VQAs, we argue that proofs that show the absence of the BP problem can lead to advances elsewhere. We split the benefits into two groups: quantum inspired classical algorithms, and learning for which problems quantum computers are better than classical ones.

Based on the existing cases, we can use proofs of the absence of the BP problem to find out how to simulate said models classically for at least most parameters θ [6]. This means that by increasing the amount of proven BP free models, we can increase the amount of models that can be classically simulated. These classically simulatable models could potentially lead to quantum inspired classical models given that they are found to be better than the previous state of the art classical models.

Additionally, the research into the BP problem gives us a lot of information about the capabilities of quantum computers for all use cases which use quantum computers to evaluate functions which can be written in the form of the general VQA model (2.8). When a model suffers from the BP problem, we know that even a quantum computer cannot solve the problem in polynomial time. On the contrary, when a model is BP free, we often find that classical computers are able to evaluate the model. The only found cases, in which classical computers are not able to evaluate BP free models in polynomial time, are some specific parameters in models which live for high probability given θ in effective polynomial subspaces. These models and parameter values are of great interest, as they separate the capabilities of quantum and classical computers.

4.3 Caveats

The connection between the absence of the BP problem and polynomial subspaces has only been shown for currently proven BP free models [6]. The findings have some mathematical backing, because in the case that we use an exponential space, the loss function can be written as an inner product between two exponentially large spaces and these kind of inner product lead on average to exponentially small and concentrated results [6]. However, there is a theoretical possibility that some model, that has not yet been inspected, does not follow the assumptions which are based on the currently known proven BP free models.

In this work we only consider exponential quantum advantage which we define as the ability to solve something in polynomial time on an quantum computer while state of the art classical algorithms require exponential time. In practice, even a polynomial advantage could lead to a situation where it makes sense to run a model on a quantum computer. As a lot of the literature focuses on exponential advantage, and therefore ignores polynomial terms, an inspection into polynomial advantage has to to be very thorough. As an example, one has to look at the scaling of the variance differently, in this work we are only interested in knowing if a model suffers from the BP problem, but when inspecting polynomial scaling, we are interested in the scaling more precisely. This is because the scaling of the variance effects the amount of needed shots, and therefore contributes to the scaling of the model.

5. Conclusion

Variational Quantum Algorithms (VQA) and quantum kernel methods, which we group together with VQAs, have been regarded as candidates for practical quantum advantage. These algorithms use quantum computers to calculate a quantity, for which it was found, that often as qubits are added the gradients vanish exponentially and the results concentrates exponentially. This problem called the Barren Plateau (BP) problem is a problem, because quantum computers are used to sample values, which leads to the need to take exponentially many samples in order to keep the statistical uncertainty smaller than the value of interest which gets exponentially small due to the BP problem. This means that the BP problem leads to exponential scaling, which goes against the goal of using quantum computers to avoid exponential scaling.

In order to avoid the BP problem, there has been research into BP free models. When inspecting common models that have been proven to be BP free, it was found that the proofs lead to the ability to, on average, simulate these models classically in polynomial time. When taking this into account the avenues for exponential quantum advantage through VQAs get scarcer. With this in mind we found five avenues in which exponential quantum advantage seems possible: there is no proof for the absence of the BP problem although its absence has been experimentally found, we find a useful trainable model which uses discrete parameters that are mapped to parameter values that lead to the model being hard to simulate classically, we use kernel methods that map discrete data to points of the model that are hard to simulate classically, we encode discrete data to initial states that are hard to simulate classically, the initial state consists of quantum data which is hard to simulate classically.

As VQAs have been claimed to be suitable for small an erroneous devices, we looked into papers that have arguments that support this notion [22, 21, 30, 34]. We found that these were outdated results in the sense, that they do not take the BP problem into account. The results about error resilience are based on the notion that we are optimizing continuous parameters, which only happens in one of our the five cases, and even the only case where models are trained the results are very model specific, meaning that general claims would be speculative at best. Claims about small program sizes could lessen the amount of errors, but the idea of short program size seems to come from the freedom to

choose parts of the program to be short, but these short choices do not seem to lead to useful models. We encourage that, if there is no new research that says otherwise, the research community stops treating VQAs as being suitable for small erroneous devices.

While the BP problem and its ramifications paint a grim future for VQAs, the generality of the loss function means that the findings about the BP problem can become useful when trying to find quantum advantage from other algorithms. Given that a suitable use case is found, the classical simulatability of the BP free models could also lead to VQAs becoming useful classical algorithms which do not need quantum computers. For these algorithms it could still be possibly that quantum computers are polynomially faster, which we did not consider in this work, because we focused on exponential advantage.

Bibliography

[1] A. Angrisani, A. Schmidhuber, M. S. Rudolph, M. Cerezo, Z. Holmes, and H.-Y. Huang. Classically estimating observables of noiseless quantum circuits. *arXiv*, 2024.

[2] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles. Equivalence of quantum barren plateaus to cost concentration and narrow gorges. *Quantum Science and Technology*, 7(4):045015, Aug. 2022.

[3] P. Bermejo, P. Braccia, M. S. Rudolph, Z. Holmes, L. Cincio, and M. Cerezo. Quantum convolutional neural networks are (effectively) classically simulable. *arXiv*, 2024.

[4] H. E. Brandt. Qubit devices and the issue of quantum decoherence. *Progress in Quantum Electronics*, 22(5-6):257–370, Sept. 1999.

[5] 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*, 3(9):625–644, Aug. 2021.

[6] M. Cerezo, M. Larocca, D. GarcÃa-MartÃn, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes. Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing. *arXiv*, 2024.

[7] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1), Mar. 2021.

[8] I. Cong, S. Choi, and M. D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, Aug. 2019.

[9] J. Dborin, F. Barratt, V. Wimalaweera, L. Wright, and A. G. Green. Matrix product state pre-training for quantum machine learning. *Quantum Science and Technology*, 7(3):035014, May 2022.

[10] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *arXiv*, 2014.

[11] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum-enhanced measurements: Beating the standard quantum limit. *Science*, 306(5700):1330–1336, Nov. 2004.

[12] M. L. Goh, M. Larocca, L. Cincio, M. Cerezo, and F. Sauvage. Lie-algebraic classical simulations for variational quantum computing. *arXiv*, 2023.

[13] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1), July 2019.

[14] H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, June 2020.

[15] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. S. Derks, P. K. Faehrmann, and J. J. Meyer. Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, 106(4), Oct. 2022.

[16] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, Sept. 2017.

[17] O. Koska, M. Baboulin, and A. Gazda. A tree-approach pauli decomposition algorithm with application to quantum computing. In *ISC High Performance 2024 Research Paper Proceedings (39th International Conference)*, pages 1–11. IEEE, May 2024.

[18] J. W. Z. Lau, K. H. Lim, H. Shrotriya, and L. C. Kwek. Nisq computing: where are we and where do we go? *AAPPS Bulletin*, 32(1):27, Sep 2022.

[19] Y. Liu, S. Arunachalam, and K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, July 2021.

[20] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, Nov 2018.

[21] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Physical Review A*, 95(4), Apr. 2017.

[22] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, Feb. 2016.

[23] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3), Sept. 2018.

[24] H. Mujtaba. Amd's next gen epyc & radeon instinct powered lumi supercomputer announced for 2021, 550 petaflops peak horsepower. Technical report, Wccftech, 2020. Accessed on 17.01.2025.

[25] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), July 2014.

[26] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, Aug. 2018.

[27] M. S. Rudolph, J. Miller, D. Motlagh, J. Chen, A. Acharya, and A. Perdomo-Ortiz. Synergistic pretraining of parametrized quantum circuits via tensor networks. *Nature Communications*, 14(1), Dec. 2023.

[28] M. Schuld. Supervised quantum machine learning models are kernel methods. *arXiv*, 2021.

[29] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3), Mar. 2020.

[30] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles. Noise resilience of variational quantum compiling. *New Journal of Physics*, 22(4):043006, Apr. 2020.

[31] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct. 1997.

[32] Acharya et al. Quantum error correction below the surface code threshold. *Nature*, Dec. 2024.

[33] Morvan et al. Phase transitions in random circuit sampling. *Nature*, 634(8033):328–333, Oct. 2024.

[34] O'Malley et al. Scalable quantum simulation of molecular energies. *Phys. Rev. X*, 6:031007, Jul 2016.

[35] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, 15(1):5200, Jun 2024.

[36] S. Thanasilp, S. Wang, N. A. Nghiem, P. Coles, and M. Cerezo. Subtleties in the trainability of quantum machine learning models. *Quantum Machine Intelligence*, 5(1), May 2023.

[37] D. Wecker, M. B. Hastings, and M. Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4), Oct. 2015.