THE UNIVERSITY *of* EDINBURGH

Designing and improving quantum algorithms for the NISQ era

By

Ioannis Kolotouros

School of Informatics
The University of Edinburgh

Doctor of Philosophy
Laboratory for Foundations of Computer Science

2024

Abstract

Hybrid quantum/classical algorithms are the most promising class of algorithms for achieving a practical advantage in the NISQ era. Their implementation usually requires a small (and perhaps noisy) quantum computer to work in conjunction with a powerful classical computer. The former generates and measures non-classically simulatable parameterized quantum states, while the latter post-processes the measurements to decide the parameter updates. However, limitations such as barren plateaux, local minima and sensitivity in the initialization of parameters prohibit their use in real-world applications. In this thesis, we propose several algorithms to address the aforementioned problems.

First, inspired by the Quantum Natural Gradient (QNG) [Quantum 4, 269 (2020)], we propose an information-theoretic quantum optimization algorithm called Random Natural Gradient (RNG) that considers the geometry of the underlying parameterized quantum states. Compared to the former method, we show that our approach requires 1) quadratically fewer quantum state preparations per iteration and 2) a quantum circuit with significantly smaller depth. Moreover, RNG uses random measurements and the classical Fisher information matrix as opposed to the quantum Fisher information matrix used in QNG. Additionally, we provide two estimators of the quantum Fisher information based on random measurements and highlight their connection to RNG. Then, inspired by stochastic-coordinate methods, we propose a novel approximation to the QNG called Stochastic-Coordinate Quantum Natural Gradient that optimizes only a small (randomly sampled) fraction of the total parameters at each iteration.

Additionally, we propose an objective function suited for classical combinatorial optimization problems. For this class of problems, it is crucial to ensure that the algorithm converges with high probability to a near-optimal solution in a short time. In Barkoutsos et al. [Quantum 4, 256 (2020)], an alternative class of objective function was introduced, called Conditional Value-at-Risk (CVaR), in which only a small fraction of the total energy samples are minimized. Here, we introduce a time-varying (evolving) objective function called Ascending-CVaR, which can be used for any combinatorial optimization problem. We show that this objective function allows for solutions that achieve the highest overlap with the optimal solution compared to standard objective functions or the CVaR proposed in [Quantum 4, 256 (2020)]. We also show that our method can be used as a heuristic for avoiding sub-optimal minima.

Finally, we propose a novel hybrid quantum/classical algorithm inspired by Adiabatic Quantum Computing. In our approach, we analyze how a small perturbation of a

Hamiltonian affects the position of the global minimum within a family of parameterized quantum states. We derive a set of equations that allow us to compute the new minimum by solving a constrained linear system of equations obtained from measuring a series of observables on the unperturbed system. This approach offers the advantage that it does not suffer from the barren plateaux problem as the parameters of the ansatz family are not initialized at random. Furthermore, we propose a discrete version of adiabatic quantum computing that can be implemented on a NISQ device while at the same time being insensitive to the initialization parameters and other limitations hindered in the optimization part of hybrid algorithms.

Lay Summary

Quantum mechanics is a theory that describes the physics of molecules, atoms, and elementary particles. While our day-to-day interactions with matter are based on the laws of Newtonian physics, the physics of subatomic particles contradict our basic human intuition. Feynman suggested that we could exploit some of these counterintuitive properties and construct computers based on these fundamental laws. These computers were named *quantum computers*, and their fundamental components were named *qubits*. A tremendous amount of research in the past few decades has shown that these computers can perform certain tasks much faster than conventional classical computers. Examples of such applications are breaking certain classical cryptographic schemes, simulation of quantum mechanical systems, or solving certain mathematical problems. However, these proof-of-principle devices require quantum computers with a large number of qubits that can perform millions of operations, which is far from what current quantum technologies have to offer.

In this thesis, we try to understand the power and limitations of noisy intermediate-scale quantum (NISQ) computers, i.e. quantum computers whose practical usage is limited due to technological limitations such as a restricted number of qubits or noise. We test the limits of certain algorithms that require constant cooperation between a quantum computer and a powerful classical device. At first, we identify several bottlenecks that prohibit the practicality of this framework. Then, we improve several subsequent parts of these algorithms by either reducing the calls on the quantum processor or by replacing some of their fundamental components. Furthermore, we propose new ways to utilize the continuous feedback loop between the quantum and classical processors. We benchmark these algorithms on mathematical problems that are very hard for classical computers to solve and test how their quantum counterparts can help solve these problems faster.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

The ideas presented in the current thesis are based on the following papers:

1. Ioannis Kolotouros and Petros Wallden. "Random natural gradient." Quantum 8:1503, 2024.
2. Ioannis Kolotouros and Petros Wallden. "Evolving objective function for improved variational quantum optimization." Physical Review Research, 4(2):023225, 2022.
3. Ioannis Kolotouros, Ioannis Petrongonas, Miloš Prokop, and Petros Wallden. "Simulating adiabatic quantum computing with parameterized quantum circuits." Quantum Science and Technology. 10(1):015003, 2024. .
4. Ioannis Kolotouros, David Joseph and Anand Kumar Narayanan. "Accelerating quantum imaginary-time evolution with random measurements", arXiv preprint arXiv:2407.03123, 2024. (Under Review)

Other works that are not included in this thesis are:

1. Boniface Yogendran, Daniel Charlton, Miriam Beddig, Ioannis Kolotouros, and Petros Wallden. "Big data applications on small quantum computers." arXiv preprint arXiv:2402.01529, 2024. (Presented at QCTiP 2024)

Acknowledgements

First of all, I would like to thank my supervisor, Petros Wallden. He was the first to believe in me and give me the opportunity to be a PhD student in the informatics department. All these years, we have had very productive discussions either on science or other related topics, and I want to thank him for being my mentor. He gave me the opportunity to work on the research topics that I liked and made sure that I always had all the resources I needed. I am really grateful for the things I learned working on his side, and as he was always there for me when I needed him, I feel that when I graduate, I'll say that I made one more friend.

Next, I want to thank the rest of the quantum computing gang in the informatics department. I was very lucky to meet these amazing people from every corner of the earth, working on different quantum computing topics. Although there were times we were discussing for hours instead of working, I believe that this is what makes an individual a true scientist. When things were not working, people were there to make you feel better and optimistic. I thank all of them for that.

Furthermore, I want to thank my family. First of all, my parents have always been by my side, supporting every decision I made in my life. Next, I want to thank my brother Nikos. His advice to pursue a PhD has been the best decision I have made so far. In addition, I want to thank my best friends, Giorgos, Thomas, and Angelos. Our endless discussions on Messenger, in which we mocked each other for our bad habits and personalities, were key to my good mental health. I remember always smiling in the informatics corridors as I watched their nonstop fighting in the group chat.

A big thanks to my friends here in Edinburgh as well. We had a great time, guys, and I'm sure we will continue making memories in the next years as well. Moreover, a huge thanks to my basketball teammates. Our team, Edinburgh Dragons, has been an important way out of my daily problems and research dead ends. I was always able to clear my mind when we were playing together.

Last but not least, I want to thank my most important person, the love of my life, and the person that kept me going, Sonia. She was there all these years, from the hard times when things were not going well to the happy times when everything was going great. While being 3800 km away for the first three years, I always felt that she never left my side. She may be a gangster, as we say, but I admire her, and I will always be grateful for meeting and loving her.

Table of Contents

Chapter 1

Introduction

It was in 1981 when Feynman first introduced the idea of quantum computing [1] as a way to avoid the exponential resources that are required to simulate quantum mechanical systems in classical computers. Around 40 years later, the quantum computing community entered the so-called noisy intermediate-scale quantum computing (NISQ) era [2] – the era from ≈ 50 qubits of Google's quantum advantage experiment [3] to devices with thousands of qubits that are anticipated in the near future. In this era, quantum computers scaled up from proof-of-principle devices to devices that can generate quantum states that cannot be classically simulated, and this opened the prospect of providing computational speedups in useful tasks.

In this era, quantum computers have small (but still non-negligible) error rates (close to 0.1%) that are very far from the ideal that fault-tolerant algorithms require (achieved through the means of an error-correcting code). However, significant advancements in the quantum hardware showed that error correction beyond the surface code threshold is possible [4], and indications of useful quantum computing started to appear [5]. However, as noted in [6], there is a specific error rate threshold that the devices must reach in order to achieve a computational advantage in the NISQ era.

For this reason, current quantum devices are still far apart from the fault-tolerant quantum machines where error correction and large coherent times are available. These devices inherit limited connectivity, short coherence times, noisy quantum operations, and a small number of qubits (of around hundreds of qubits), posing the question of whether and how these devices can be exploited. Thus, algorithms such as Shor's [7], Grover's [8], or HHL [9], which require large depths, cannot be implemented in these devices.

To address these issues, researchers considered using these small (and noisy) quantum

devices in conjunction with powerful, fast and reliable classical computers [10]. The former would generate and measure non-classically simulatable quantum states. At the same time, the latter would post-process the measurements with the goal of achieving quantum advantage, i.e. perform better in certain tasks than their pure classical counterparts. This approach has the advantage that the quantum circuit can remain in small depth, and the hardness is transferred to the classical computer (usually with the means of a classical optimization algorithm). Thus, the term hybrid quantum/classical computing was born.

One leading example of hybrid quantum/classical algorithms is the variational quantum algorithms [10, 11]. These algorithms can be divided into three subsequent parts. First, the targeted problem is mapped to the mathematical task that these algorithms are designed to solve, which is the search for the ground-state energy of a Hamiltonian. The second step is to employ a parameterized quantum circuit (ansatz) that is expressive enough so that it contains the ground state (or an approximation to it) and is trainable. Finally, a suitable chosen classical optimization algorithm will post-process the measured quantum states and update the parameters towards the direction that minimizes the energy of the system.

Variational quantum algorithms have gained significant research interest over the past few years. The main reason is that many problems can be easily tackled using this framework. Examples of such problems are ground state preparation of molecules [12–14], quantum compiling [15, 16], simulation of real and imaginary-time evolution of quantum mechanical systems [17, 18] or machine learning applications [19, 20] to name just a few.

Although this framework is rather promising and versatile, since the solution of most mathematical problems can be mapped onto the ground state of a Hamiltonian, there are multiple bottlenecks that prohibit its actual usefulness. Conventional quantum algorithms such as those mentioned before have strong theoretical guarantees, compared to VQAs, where strong complexity-theoretic arguments are hard to prove, and their practicality relies on heuristics.

The hardest part in VQAs is the training part, i.e. finding the optimal parameters that will generate the quantum state of interest. Understanding the emerging landscapes of VQAs has gained tremendous research interest over the past few years [21–28]. In [22], the authors proved, for the MaxCut problem, that unless $\mathsf{P} = \mathsf{NP}$, there does not exist an ansatz family (consisting of commuting generators) with a number of parameters which is polynomial in the system size, and a convex landscape. Furthermore, *overparametrization* [24, 25] for specific ansatz families can occur if the number of parameters is polynomial to the system size, and may lead to a computational phase transition where only high-quality

minima exist in the objective function landscapes. Moreover, [29] showed that analyzing the Dynamical Lie Algebra of the generators of the ansatz (and thus the controllability of the system) could help in designing trainable ansatz families.

On top of that, people have also looked at the geometry of the underlying parameterized quantum states, as in some cases, it can enhance the optimization process [30, 31]. Studying the geometry of the landscapes requires tools such as the quantum Fisher information matrix (QFIM) [32, 33]. The underlying geometry of parameterized state-space can help understand the emerging landscapes of the parameterized quantum circuits. The QFIM has been extensively used in the NISQ era either as a capacity measure [34], a generalization measure for quantum machine learning models [35], or even as a tool to construct naturally parameterized architectures [36].

At this point, we will outline *the three main bottlenecks of VQAs* that prohibit their practicality below.

The first main bottleneck comes from the fact that most classical optimization algorithms do not consider information about the underlying generated quantum states. Specifically, most of the classical optimization algorithms propose the update step according to information related to the expectation value of the Hamiltonian (or its first and higher-order derivatives). This implies that a lot of information (about changes happening in the parameterized quantum state-space) is not exploited during the optimization step. On top of that, information-theoretic methods that utilize this information usually require a significantly large number of calls on the quantum processors. For this reason, it is important that the hybrid quantum/classical schemes distribute optimally the resources between the processors. As such, the number of quantum resources needed to solve a problem must be limited enough so that we can still exploit the quantum effects, and the classical resources should be maximized as they offer speed and reliability.

The second main bottleneck is the training part. The objective function landscapes of VQAs are highly non-convex, which makes the algorithms very hard to train. This implies that multiple local minima exist where the classical optimizer may falsely converge, returning solutions that are far from optimal. As pointed out in [37, 38], the loss landscapes in shallow-depth VQAs, such as those utilized in this thesis, are filled with a vast amount of local minima, which makes them untrainable. This implies that if the parameterized quantum circuit is initialized in a configuration near a far-from-optimal local minimum, the algorithm will most likely return a bad approximation of the target solution.

Finally, the third main bottleneck arises with the choice of the ansatz family. Finding a parameterized family of gates that contain the ground state of interest is hard, a problem which is usually referred to as *ansatz expressivity* and people have introduced many ways to quantify it [39–41]. To add to the aforementioned problems, highly expressive ansatz families that span a large fraction of the total Hilbert space are hard to train. Specifically, initializing the parameters at random in a highly expressive ansatz family leads to flat energy landscapes, where identifying the descent direction requires exponentially many quantum resources; these flat valleys are named *barren plateaux* [21, 42–45].

Previously, people have tried to tackle some of the bottlenecks above using several different approaches. For example, to address the barren plateaux problem, researchers have designed parameterized architectures [46–51] that do not exhibit flat landscapes due to either correlations between parameters, clever initializations or due to the structure of the ansatz. Additionally, to address the bad convergence due to local minima, people have utilized information-theoretic classical optimization algorithms [31], employed neural networks [52] or used multi-start methods [53].

Other than that, people have developed additional hybrid frameworks with low-depth quantum circuit requirements that are suited for the NISQ era. Examples are frameworks that require measuring the generated quantum state at random [54]. Previous works have utilized these frameworks to calculate quantities such as the Rényi entropy experimentally [55, 56], to identify mixed-state entanglement [57] and to estimate the overlap of two quantum states [55, 58]. On top of that, they have been utilized to calculate certain observables of a quantum state [59], as quantum states typically carry more information.

In this thesis, we aim to advance the hybrid quantum/classical computing field and improve the existing algorithms. Our goal is to bring near-term quantum computing a step closer to a practical quantum advantage.

In this thesis, we aim to tackle each one of the three bottlenecks described above. Our contributions can be summarized as follows:

In Chapter 3, we aim to solve the first bottleneck described above. We introduce an information-theoretic optimization algorithm, called *random natural gradient* (RNG) that is suited for VQAs and draws a connection to the quantum natural gradient (QNG) [30] (and subsequently to the quantum imaginary-time evolution) but requires quadratically less quantum resources and achieves significant speedup over other classical optimization algorithms. In the same chapter, we analyze how different measurements on

parameterized quantum states approximate the underlying parameterized state-space geometry. Additionally, we introduce conditions under which preconditioning the gradient of the loss function with an information matrix will result in a decrease in the energy. Moreover, we provide estimators of the quantum Fisher information based on random measurements and show that under the extreme condition where one random measurement is used, the approximation leads to the random natural gradient. Finally, we introduce a novel approximation to the QNG, which we call *stochastic-coordinate quantum natural gradient* that utilizes only a small (randomly sampled) fraction of the total parameters of the ansatz family and benchmark it against the QNG. All optimization methods are benchmarked on the MaxCut, Number Partitioning and the Transverse-field Ising Chain model. The content of this chapter has appeared in [13, 60]

Next, in Chapter 4, we aim to solve the bad convergence bottleneck due to multiple local minima. We introduce an *evolving objective function* that starts with the conditional value at risk (CVaR) defined in [61] and gradually, in the optimization process, becomes the full energy of the quantum state. Alternative forms of this ascending-CVaR objective function are considered, and linear and sigmoid functions (that appear to perform better) are selected. We test our proposal with classical numerical simulations (using up to 20 qubits and a shot simulator), both in the settings of VQE with hardware-efficient ansatz families and in QAOA. Our analysis was done on three different classical combinatorial optimization problems, namely MaxCut, Number Partitioning, and Portfolio Optimization. The content of this chapter has appeared in [62].

In chapter 5, we aim to address the *barren plateaux* and bad initialization problem. At first, we study how small perturbations of a Hamiltonian affect the optimization landscape and how much the minima are shifted under these perturbations. This enables us to follow the trajectory of a minimum in the cost landscape as a (parameterized) Hamiltonian varies by solving a constrained linear system of equations. As such, we obtain the ground state without having to rely on hyperparameters. Additionally, we formulate an algorithm to find the best approximation of the ground state of a Hamiltonian within a family of parameterized quantum states that (i) can be applied in an early fault-tolerant device, (ii) is not sensitive to the initialization parameters, and (iii) requires fixed calls to the quantum computer with theoretical guarantees on the performance. Moreover, we quantify the quantum resources needed for our algorithm and show that finding the minimum of the perturbed Hamiltonian can be cast as a semidefinite program. Finally, we test our algorithm on small instances of both classical optimization problems (MaxCut and Number Partitioning) and on "non-classical" Hamiltonians, which include

non-diagonal terms, such as the random Transverse-Field Ising Chain. The content of this chapter has appeared in [63].

Chapter 2

Preliminaries

In this section, we will outline all the necessary information about parameterized quantum circuits, variational quantum algorithms, and all the mathematical problems (classical and quantum) that will be used to benchmark our methods in the following chapters.

2.1 Parameterized Quantum Circuits

Consider a quantum circuit of n qubits composed of a series of parameterized and non-parameterized gates. These *parameterized quantum circuits* (PQCs) can be described by a unitary operator $U(\boldsymbol{\theta})$ with $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m) \in \mathbb{R}^m$ (and $m \in \mathcal{O}(\text{poly}(n))$). The type/number of gates is usually referred to as the *ansatz family*. If K is the total number of gates of the PQC, then the ansatz family can be written in its more generic form as:

$$U(\boldsymbol{\theta}) = \prod_{k=K}^{1} e^{-i\theta_k g_k} \tag{2.1}$$

where g_k are the generators (hermitian operators $g_k^\dagger = g_k$) corresponding to every unitary of the PQC. In most cases, only a small fraction of the total K parameters are tunable, while the other $K - m$ angles correspond to fixed non-parameterized gates, usually two-qubit gates introducing entanglement, e.g. controlled-NOT or controlled-Z operations. These types of ansatz families describe most of the PQCs that appear in the literature, such as the hardware-efficient ansatz [64], the quantum alternate operator ansatz [65], the quantum optimal control ansatz [61] or the unitary coupled cluster ansatz [66].

The parameterized quantum circuit acts on a reference state $|\phi\rangle$ usually chosen to be the $|0\rangle \equiv |0\rangle^{\otimes n}$ state and generates the parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle$. As we will discuss later, the classical subroutines in VQAs require knowing the expectation

value of a Hamiltonian and its first and high-order derivatives with respect to different quantum states. The expectation value of the Hamiltonian with respect to a parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle$ is:

$$F(\boldsymbol{\theta}) = \langle 0|\, U^{\dagger}(\boldsymbol{\theta}) H U(\boldsymbol{\theta})\,|0\rangle \tag{2.2}$$

Additionally, there are methods to calculate the first and high-order derivatives for any observable O with $O^{\dagger} = O$ (and thus for any Hamiltonian H). These methods are called *parameter-shift rules*. The parameter-shift rules [67–70] state that derivatives of the expectation value of an observable O (denoted as $\langle O(\boldsymbol{\theta})\rangle$) can be calculated as a linear combination of the expectation values of the observable at two different parameter settings:

$$\frac{\partial \langle O(\boldsymbol{\theta})\rangle}{\partial \theta_j} = r\left[\langle O(\boldsymbol{\theta} + \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}})\rangle - \langle O(\boldsymbol{\theta} - \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}})\rangle\right] \tag{2.3}$$

where $\pm r$ are the eigenvalues of the generator g_j [1] (see Eq. (2.1)) corresponding to the gate of the parameter θ_j and $\hat{\boldsymbol{e}}_{\boldsymbol{j}}$ is the unit vector pointing in the j-th direction. For the case where the observable is the Hamiltonian, the equation can be written as:

$$\frac{\partial F(\boldsymbol{\theta})}{\partial \theta_j} = r\left[F\Big(\boldsymbol{\theta} + \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\Big) - F\Big(\boldsymbol{\theta} - \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\Big)\right] \tag{2.4}$$

In this thesis, we will also be interested in the second-order derivatives of the expectation value (i.e. the Hessian). The matrix elements of the Hessian can be calculated as:

$$\begin{aligned}\frac{\partial^2 F(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} = r^2\Big[&F\left(\boldsymbol{\theta} + \frac{\pi}{4r}(\hat{\boldsymbol{e}}_{\boldsymbol{j}} + \hat{\boldsymbol{e}}_{\boldsymbol{k}})\right) - F\left(\boldsymbol{\theta} + \frac{\pi}{4r}(\hat{\boldsymbol{e}}_{\boldsymbol{j}} - \hat{\boldsymbol{e}}_{\boldsymbol{k}})\right) \\ &-F\left(\boldsymbol{\theta} + \frac{\pi}{4r}(-\hat{\boldsymbol{e}}_{\boldsymbol{j}} + \hat{\boldsymbol{e}}_{\boldsymbol{k}})\right) + F\left(\boldsymbol{\theta} - \frac{\pi}{4r}(\hat{\boldsymbol{e}}_{\boldsymbol{j}} + \hat{\boldsymbol{e}}_{\boldsymbol{k}})\right)\Big]\end{aligned} \tag{2.5}$$

2.2 Variational Quantum Algorithms

As we discussed in the introduction section, variational quantum algorithms refer to a class of hybrid quantum/classical algorithms where a quantum computer works in parallel with a classical computer employed with a classical optimization algorithm. This framework offers practicality in a NISQ setting but lacks generic theoretical guarantees about its performance. The general VQA framework is illustrated in Figure 2.1

At first, the user is presented with a mathematical problem that is mapped into an interacting qubit Hamiltonian H consisting of $L = \mathcal{O}(\text{poly}(n))$ Pauli strings, where n is

[1] There are also general parameter-shift rules, see [69].

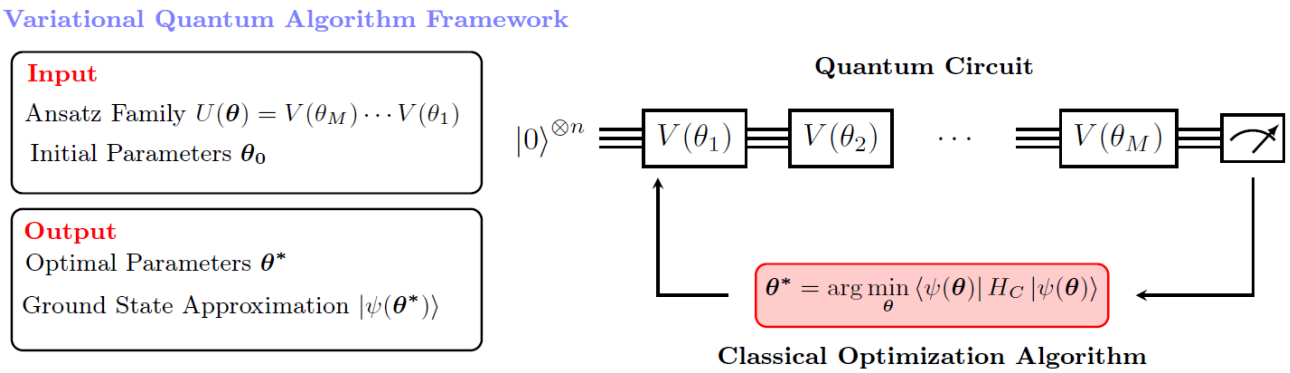

Figure 2.1: General Variational Quantum Algorithm framework. In the first step, the parameterized family of gates and the initial parameters are chosen. Then, the classical optimization algorithm iteratively updates the parameters towards the direction of the ground state energy. When the classical optimizer converges to a minimum, the algorithm stops, and an approximation of the ground state is given at the output.

the number of qubits. The Hamiltonian is chosen such that its ground state corresponds to the solution of the mathematical problem of interest. The most general way to express this Hamiltonian is:

$$H_C = \sum_{i=1}^{L} c_l P_l \tag{2.6}$$

where $c_l \in \mathbb{R}$ is the real coefficient corresponding to Pauli string P_l.

Next, the user employs a parameterized quantum circuit as discussed in Sec. 2.1 so that the ground state (or at least a good approximation to it that cannot be found efficiently by a classical computer) is contained within the ansatz. The quantum computer then iteratively generates and measures (on an appropriate basis) the parameterized quantum state so that the classical computer can post-process the measurement outcomes.

The classical computer works in parallel with the quantum computer. The user selects a classical optimization algorithm and an objective function $\mathcal{L}(\boldsymbol{\theta})$ (loss function) so that its minimum corresponds to the solution of the mathematical problem of interest. The most usual choice for the loss function is the expectation value of the Hamiltonian, i.e. $\mathcal{L}(\boldsymbol{\theta}) = F(\boldsymbol{\theta})$. In other words, the goal is to find the optimal angles $\boldsymbol{\theta^*}$ such that:

$$\boldsymbol{\theta^*} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{2.7}$$

As we will show in Chapter 4, different objective functions lead to improved performance [61, 62, 71]. When the classical optimization algorithm has converged, the VQA terminates and outputs both the optimal solution of the problem as well as the quantum state

corresponding to the solution:

$$|\psi(\boldsymbol{\theta}^*)\rangle\,, \mathcal{L}(\boldsymbol{\theta}^*)$$

2.2.1 Variational Quantum Eigensolver

One of the most used hybrid quantum/classical algorithms in the NISQ era is the Variational Quantum Eigensolver (VQE) [72]. This algorithm employs either hardware-efficient parameterized quantum circuits (usually native to the available quantum hardware) as seen in Figure 2.2 or other types of parameterized architectures such as the Hamiltonian Variational Ansatz (HVA) [51, 73] or the quantum optimal-control ansatz [74] to name a few.

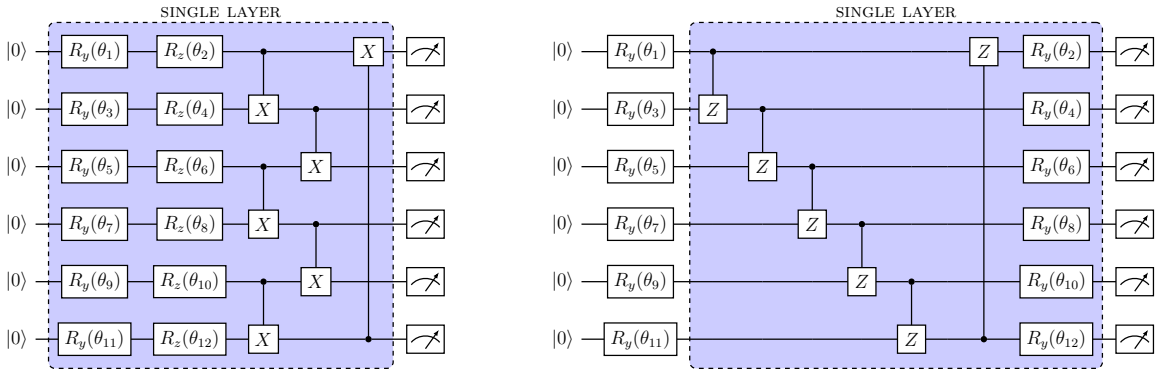

Figure 2.2: Six-qubit examples of hardware-efficient parameterized quantum circuits (with nearest-neighbours interaction) used in our experiments. The filled blue square corresponds to a single layer. Similar quantum circuits were used with an all-to-all connectivity.

VQE was originally designed to prepare ground states of molecular systems, but it can also be used to tackle optimization problems [75]. Since it is a type of variational quantum algorithm, the main idea is to map the mathematical problem into an interacting qubit Hamiltonian whose ground state corresponds to the solution of the problem at hand.

The hardware-efficient ansatz families used in VQE fall in the more general category of *problem-agnostic* ansatz families, meaning that the structure of the ansatz carries no information about the problem itself. Other problems use different ansatz families like the Unitary Coupled Cluster [76], the quantum optimal-control inspired ansatz [74] or the Hamiltonian variational ansatz [73].

The main problem with the hardware-efficient parameterized quantum circuits seen in Figure 2.2 is that as their size increases and the parameters are initialized at random, they become approximate 2-designs [77]. As such, the variance and mean of the partial

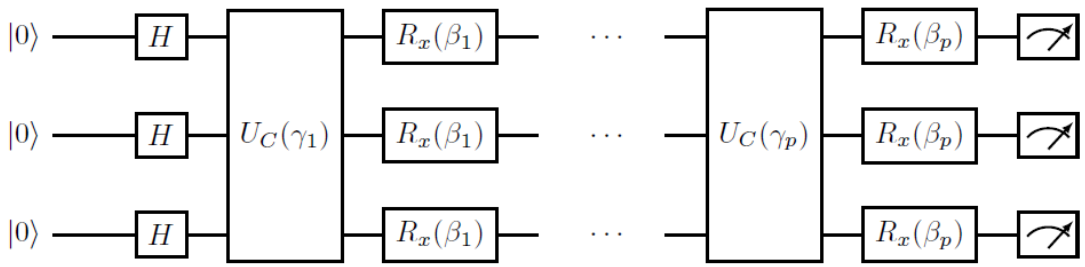

Figure 2.3: General framework of a p-layer QAOA consisting of $2p$ variational angles.

derivatives of the expectation value vanish exponentially fast (with the system size), meaning that exponentially many repetitions are needed so that the user can accurately predict the descent direction.

2.2.2 Quantum Approximate Optimization Algorithm

The *Quantum Approximate Optimization Algorithm* (QAOA) [78] is a variational quantum algorithm mostly used for combinatorial optimization problems (see Sec. 2.3). There has been a lot of research on its performance in shallow depths [79–82], which matches that of classical solvers for certain problems [83]. However, its actual limits in intermediate depths are still unknown.

The QAOA algorithm applies an alternation of two unitary transformations, one encoding the cost function H_C (i.e. the Hamiltonian of interest), $U(H_C) = e^{-i\gamma H_C}$, and the other a mixer Hamiltonian $H_B = \sum \sigma_i^x$, $U(H_B) = e^{-i\beta H_B}$, where γ and β are *variational angles* specifying the "time" for which the unitary transformations are applied. The system is initialized at the ground state of H_B, and the alternating ansatz of $U(H_B)$ $U(H_C)$ is applied p-times, with p defining the number of layers of the algorithm (see Figure 2.3), producing the state:

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_B} e^{-i\gamma_1 H_C} |+\rangle \tag{2.8}$$

where $|+\rangle$ is the uniform superposition state, $\boldsymbol{\gamma} = (\gamma_1 \dots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1 \dots, \beta_p)$. QAOA has also been generalized for different choices of unitaries in [65].

With sufficient repetitions of the algorithm, the expectation value is calculated as:

$$F_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \boldsymbol{\beta}, \boldsymbol{\gamma}| H_C |\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle \tag{2.9}$$

until the $2p$ optimal parameters $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$ are found. If C_{opt} is the optimal cost function, then the target of the algorithm is to maximize the approximation ratio, defined as:

$$r^* = \frac{F_p(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)}{C_{opt}} \tag{2.10}$$

As we already discussed, finding the optimal parameters is far from trivial since the loss function landscape is highly non-convex, filled with local minima where a classical optimizer could easily get stuck. The hardest part of QAOA, and in general of a variational quantum algorithm, is finding the optimal parameters that will lead to a high overlap with the optimal (or near-optimal) bit-string or low expectation value. Ways to avoid "getting stuck" in local minima is using multi-start methods [53], clever initializations [84, 85] or the Ascending-CVaR objective function that we describe in Chapter 4.

2.3 Combinatorial Optimization Problems

In this thesis, we will investigate problems with discrete but exponentially many possible solutions, namely *combinatorial optimization problems* [86]. These are all important problems in their own right, so improving the performance of hybrid quantum/classical algorithms in these problems is of independent interest. Moreover, testing our proposals on different types of combinatorial optimization problems demonstrates that the improvements observed are generic and motivate further use for different applications.

Initially, we map any combinatorial optimization problem to *a Quadratic Unconstrained Binary Optimization* (QUBO) problem. This is what we will do for all our examples. This is a necessary step in the case of QAOA but not in the case of VQE, where we can solve an optimization problem without mapping it to QUBO. QUBO problems seek to solve (find the $\boldsymbol{x} \in \{0,1\}^n$ that minimizes the expression):

$$\min_{\boldsymbol{x}} \left(\boldsymbol{b}^T \boldsymbol{x} + \boldsymbol{x}^T A \boldsymbol{x} \right) \tag{2.11}$$

where $\boldsymbol{b} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. These cost functions can easily be mapped to an Ising Hamiltonian [87] by first transforming the binary variables $x_i \in \{0,1\}$ according to:

$$x_i = \frac{1 - z_i}{2} \tag{2.12}$$

where $z_i \in \{-1, +1\}$ are spin variables, and then turning the cost function to a Hamiltonian by promoting these variables to Pauli σ_i^z operators, one for each qubit i. The QUBO problem then transforms to

$$\min_{z} \boldsymbol{c}^T \boldsymbol{z} + \boldsymbol{z}^T Q \boldsymbol{z} \tag{2.13}$$

where the new $\boldsymbol{c} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n \times n}$ are easily computable.

Then, by replacing the spin variable z_i with the Pauli σ_i^z operator with corresponding eigenvalues $\{-1, +1\}$, the problem translates into finding the ground state, i.e. the spin configuration of an n-qubit system interacting with the Hamiltonian:

$$H = \sum_{i=1}^{n} c_i \sigma_i^z + \sum_{i=1}^{n} Q_{ij} \sigma_i^z \sigma_j^z \tag{2.14}$$

2.3.1 MaxCut Problem

The first problem is *MaxCut*. It is one of the most studied combinatorial problems in the context of variational quantum algorithms due to the simplicity and guaranteed performance, at least for some instances [78, 79, 81, 88, 89].

Let $G(V, E)$ be a non-directed n-vertex graph, where V is the set of vertices, E is the set of edges, and w_{ij} are the weights of the edges. A *cut* is defined as a bipartion of the set V into two disjoint subsets P, Q, i.e. $P \cup Q = V$ and $P \cap Q = \emptyset$. Equivalently, we label every vertex with either 0 or 1, where it is understood that the vertex belongs to set P if it takes the value 0 and to set Q if it takes the value 1. The aim is to maximize the following cost function:

$$C(\boldsymbol{x}) = \sum_{i,j=1}^{n} w_{ij} x_i \left(1 - x_j\right) \tag{2.15}$$

This intuitively corresponds to finding a partition of the vertices into two disjoint sets that "cuts" the maximum number of edges. By applying the transformation, Eq. (2.12), the cost function transforms into:

$$C(\boldsymbol{z}) = \sum_{\langle i,j \rangle \in E} \frac{w_{ij}}{2} \left(1 - z_i z_j\right) \tag{2.16}$$

Maximizing the cost function above corresponds to finding the ground state of the Hamiltonian[2]:

$$H_{\mathrm{MC}} = -\sum_{\langle i,j \rangle \in E} \frac{w_{ij}}{2} \left(1 - \sigma_i^z \sigma_j^z\right) \tag{2.17}$$

It is known that it is NP-hard to achieve an approximation ratio (see Eq. (2.10)) of $r^* \geq \frac{16}{17}$ for MaxCut on all graphs [90]. The best classical approximation algorithm is

[2]Note the overall minus sign that turns the maximization of the cost function to finding the minimum energy for the Hamiltonian.

that of Goemans and Williamson (GW) [91], which uses semi-definite programming to achieve an approximation ratio $r^* \approx 0.87856$ for all graphs. Note that being NP-hard implies that we do not expect to have an efficient quantum algorithm (poly-time) to solve the problem for its hardest instances[3], but we may be able to get improvements using quantum algorithms (either by smaller speed-ups or by heuristics that could solve more instances than classical heuristics).

Although it was proven that constant-depth QAOA does not outperform GW for certain classes of problems [92], there are instances where the approximation ratio of the former is larger than the latter [88]. Note here that QAOA beats random guessing even at $p = 1$ [78], while Machine Learning techniques have been used to classify for which graph types it is better to use QAOA instead of GW [93]. In general, however, the performance of QAOA in intermediate depths is still unexplored.

2.3.2 Number Partitioning

The second problem is *Number Partitioning* and is stated as follows. Given a set of N positive integers $S = \{n_1, n_2, ..., n_N\}$, the target is to find a bipartion of the set S into two disjoint subsets P, Q, where $P \cup Q = S$ and $P \cap Q = \emptyset$ so that the difference between the sum of the elements on the set P and the set Q is minimized. We thus want to minimize the cost function:

$$C(\boldsymbol{x}) = \left(\sum_{i=1}^{N} (2x_i - 1) n_i \right)^2 \tag{2.18}$$

The binary string $\boldsymbol{x} = x_1 x_2 \dots x_n$ corresponds to one configuration where a number n_i is placed in the P set ($x_i = 0$) or in the Q set ($x_i = 1$). The cost function can easily be mapped to the Ising Hamiltonian [87]:

$$H_{\mathrm{NP}} = \left(\sum_{i=1}^{N} \sigma_i^z n_i \right)^2 \tag{2.19}$$

By expanding the cost function Eq. (2.19), the Hamiltonian can be expressed as:

$$H_{\mathrm{NP}} = \sum_{i \neq j} (n_i n_j) \sigma_i^z \sigma_j^z + \sum_{i=1}^{N} n_i^2 \tag{2.20}$$

If we neglect the constant term, we can see that the Number Partitioning problem can be easily mapped to the *Sherrington-Kirkpatrick model*, which is an energy minimization problem with all-to-all random couplings that was recently analyzed on [94].

[3] NP is strongly believed not to be included in BQP

Although the problem is known to be NP-Hard, it is also known as the "easiest hard problem". That is because there exists a "hard-easy" phase transition [95] where instances of the easy phase can be efficiently tackled using heuristics [96]. Interestingly, it appears that one may be able to tackle some of the instances in the "hard phase" using VQAs.

2.3.3 Portfolio Optimization

The third problem is *Portfolio Optimization* [97, 98] and is stated as follows. Given a set of n assets $\{0, \cdots, n\}$, corresponding expected returns μ_i and covariances Σ_{ij}, a risk factor $q > 0$ and a budget $B \in \{1, \ldots, n\}$, the considered portfolio optimization problem tries to find a subset of assets $P \subset \{1, \ldots, n\}$ with $|P| = B$ such that the resulting *q-weighted-mean-variance*, i.e. $\sum_{i \in P} \mu_i - q \sum_{i,j \in P} \Sigma_{ij}$, is maximized. In other words, we want to maximize the cost function:

$$C(\boldsymbol{x}) = \sum_{i=1}^{n} \mu_i x_i - q \sum_{i,j=1}^{n} \Sigma_{ij} x_i x_j \tag{2.21}$$

along with the constraint

$$\sum_{i=1}^{n} x_i = B \tag{2.22}$$

The *portfolio vector* $x \in \{0, 1\}^n$, consisting of n binary decision variables, indicates whether an asset is picked ($x_i = 1$) or not ($x_i = 0$). The constraint in (2.22) is translated as an extra penalty term in the Hamiltonian $(\sum_{i=1}^{n} x_i - B)^2$.

The problem is known to be NP-complete [99]. We apply the transformation, Eq. (2.12), so the cost function transforms into:

$$\begin{aligned} C(\boldsymbol{z}) = -q \sum_{i,j=1}^{n} \frac{\Sigma_{ij}}{4} z_i z_j + \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{q \Sigma_{ij} z_i}{2} - \frac{\mu_i z_i}{2} \right) \\ + \sum_{i=1}^{n} \left(\frac{\mu_i}{2} - \sum_{j=1}^{n} \frac{q \Sigma_{ij}}{4} \right) \end{aligned} \tag{2.23}$$

which, along with the extra penalty term, corresponds to minimizing the Hamiltonian:

$$\begin{aligned} H_{\mathrm{PO}} = \sum_{i,j=1}^{n} \frac{q \Sigma_{ij}}{4} \sigma_i^z \sigma_j^z - \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{q \Sigma_{ij} \sigma_i^z}{2} - \frac{\mu_i \sigma_i^z}{2} \right) \\ - \sum_{i=1}^{n} \left(\frac{\mu_i}{2} - \sum_{j=1}^{n} \frac{q \Sigma_{ij}}{4} \right) + \left(\sum_{i=1}^{n} \sigma_i^z + \frac{n}{2} - B \right)^2 \end{aligned} \tag{2.24}$$

Portfolio optimization, as given in Eq. (2.21), has been tackled using variational quantum algorithms [100], using warm-starting QAOA [101] and on D-wave systems using quantum annealing [102]. Prior to our work, [103] developed a quantum-walk-based optimization algorithm and [104] considered a more general setting of portfolio optimization, called dynamic portfolio optimization, where one has to allocate weights to a number of assets in a period of time in order to maximize the overall return.

2.4 Quantum many-body problems

Other than classical combinatorial optimization problems, we will also investigate quantum many-body problems. These problems correspond to spin-systems that interact with Hamiltonians with extra off-diagonal terms and have been used as benchmarks in the VQA literature [17, 73, 105–107]. This type of Hamiltonians, compared to classical optimization Hamiltonians, have eigenvectors that do not correspond to the computational basis vectors.

The Transverse-Field Ising Chain (TFIC) describes a quantum system that interacts under a Hamiltonian with extra off-diagonal terms. The Hamiltonian describing the TFIC model (with periodic boundary conditions) is:

$$H_{\text{TFI}} = -\sum_{k=1}^{n} J_k \sigma_k^z \sigma_{k+1}^z - h \sum_{k=1}^{n} \sigma_k^x \tag{2.25}$$

where $(J_k,\, h)$ are coupling coefficients. Other than the TFIC, we will also look at the Heisenberg model. The following Hamiltonian describes the 1D XXX Heisenberg model:

$$H_{XXX} = J \sum_{i=1}^{n-1} \left(\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z\right) + h \sum_{i=1}^{n} \sigma_x^i \tag{2.26}$$

Chapter 3

Random Natural Gradient

3.1 Introduction

In order to make a hybrid quantum/classical framework practical (and thus useful), certain conditions must be met. At first, we need to distinguish the classical from the quantum resources required to solve a problem. For the parameterized quantum circuits of thousands of parameters and their corresponding matrices (e.g., Hessian or information matrices), classical computers are powerful enough to perform standard linear algebra calculations with perfect accuracy. On the other hand, quantum computers are still imperfect, with slow compilation times. As such, the number of quantum resources required to solve a problem must be limited enough so that we still exploit the quantum effects, and the classical resources must be maximized as they offer speed and reliability.

In the past few years, a lot of interest has been focused on all subsequent parts of VQAs. People have utilized both gradient-based methods that exploit parameter-shift rules [67–69] or gradient-free methods [108, 109] that treat the quantum circuit as a black box. On the other hand, [27] proposed that one should construct a quadratic model of the loss landscape by performing measurements on the quantum computer and then minimize this quadratic approximation on the classical computer, reducing the overall quantum resources. Additionally, [110, 111] argued on whether we can construct algorithms that will allow VQAs to be trained as efficiently as classical neural networks by reducing the quantum overhead required for the gradient calculation. In this chapter, we focus on the classical optimization part and especially on the information that the classical computer receives in order to update the parameters of the quantum circuit.

In [30], the authors generalized the idea of the *natural gradient* [112] in the quantum setting and introduced a novel classical optimization algorithm that takes into consideration how small changes in the parameter space affect the generated quantum states; the algorithm was named *quantum natural gradient* (QNG). However, such updates require the calculation of the quantum Fisher information matrix (QFIM) [32] at each step, which in general is computationally expensive to calculate, and thus, using it in VQAs becomes impractical. The QNG has also been further extended in the case of noisy and nonunitary circuits [113].

The QNG algorithm is also related to the quantum imaginary-time evolution (QITE) [12, 30]. QITE [114] is a promising tool to prepare thermal or ground states of Hamiltonians, as convergence is guaranteed when the imaginary-evolved state has a non-zero overlap with the ground state [13]. As we will show, a user can employ a parameterized quantum circuit and dynamically change the parameters of the circuit so that the generated state follows the imaginary time evolution, a method called VarQITE [12].

In this chapter, we give two new optimization methods, improving on efficiency over the QNG optimizer. We first introduce a method called *random natural gradient*. As we will show, preparing a quantum state and measuring it on a random basis (by applying a random unitary and measuring it on the computational basis) offers a significant speedup in a VQA optimization framework. Random measurements have previously been used to construct classical shadows of quantum states [54, 59, 115], to extend the size of quantum computation beyond the physical qubits of a device [116] or to measure properties of quantum mechanical systems [55, 56]. Moreover, they have also been used to discriminate two quantum states [117]. Then, we provide two estimators of the quantum Fisher information matrix (that is used in QNG) and show how our method is related to the quantum imaginary-time evolution. Finally, inspired by classical coordinate-descent methods [118–120], we propose an approximation to the quantum natural gradient that requires only a fraction of the total resources at every iteration that we call stochastic-coordinate quantum natural gradient.

Our Contributions:

- We introduce a novel optimization technique, which we call *random natural gradient* that is quadratically faster (in terms of quantum state preparations) than the *quan-*

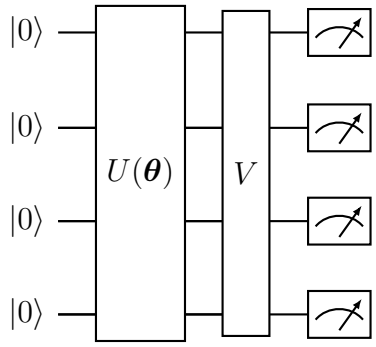

Figure 3.1: General random-measurement framework rotating the parameterized quantum state by a unitary $V \sim \nu$ where $\nu \subseteq U(2^n)$ and then measuring in the computational basis.

tum natural gradient and achieves significant speedup over classical optimization algorithms used.

- We analyze how different measurements on parameterized quantum states can approximate the underlying geometry in the state space.
- We introduce two estimators of the quantum Fisher information matrix based on random measurements.
- We introduce conditions under which preconditioning the gradient of the loss with an information matrix will result in a descent direction.
- We introduce a novel approximation to the quantum natural gradient called *stochastic-coordinate quantum natural gradient* that utilizes only a portion of the total parameters of the parameterized quantum circuit and benchmark it against the quantum natural gradient.

3.1.1 Distance of Probability Distributions

As we discussed in Sec. 2.2 the quantum computer prepares a parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle$. Once the state has been prepared, a measurement basis is chosen, and the system of qubits is measured. The measurement basis can be changed by first applying a unitary matrix V and then performing projective measurements on each qubit (see Figure 3.1).

Once the measurement basis is selected, the system of qubits is prepared and measured a constant number of times with respect to a measurement basis $\mathcal{M}$, where, in our case,

the basis is determined by the unitary V. The probability $p_{\boldsymbol{s}}^V$ of each outcome $\boldsymbol{s} \in \{0,1\}^n$ is given as:

$$p_{\boldsymbol{s}}^V = \mathrm{Tr}\big(V\rho(\boldsymbol{\theta})V^\dagger \Pi_{\boldsymbol{s}}\big) \tag{3.1}$$

where $\Pi_{\boldsymbol{s}} = |\boldsymbol{s}\rangle\langle\boldsymbol{s}|$ is the projection operator on the $\boldsymbol{s}$-th eigenspace and $\rho(\boldsymbol{\theta}) = |\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})|$. As a result, a different choice of basis $\mathcal{M}$ gives rise to a different probability distribution $p_{\mathcal{M}}(\boldsymbol{\theta}) = \{p_{\boldsymbol{s}}^V(\boldsymbol{\theta})\}$ with $p_{\mathcal{M}}(\boldsymbol{\theta}) \succcurlyeq 0$, $\|p_{\mathcal{M}}(\boldsymbol{\theta})\|_1 = 1$. The number of different probability outcomes for a n-qubit state is upper bounded by 2^n and depends on the measurement basis $\mathcal{M}$. However, if a quantum state is measured only K times (with $K \ll 2^n$) then this number is bounded by K. For now, we will assume that the number of different measurement outcomes is K and so $p_{\mathcal{M}}(\boldsymbol{\theta}) \in \Delta^{K-1}$, where Δ^{K-1} is the probability simplex of dimension $K-1$.

It will be very useful to introduce a measure that quantifies distances in the space of probability distributions. Let $\boldsymbol{u}, \boldsymbol{v} \in \Delta^{K-1}$ with $\|\boldsymbol{u}\|_1 = \|\boldsymbol{v}\|_1 = 1$ be two probability distributions. The (*Kullback-Leibler*) KL-divergence (or else the relative entropy) is defined as:

$$\mathrm{KL}(\boldsymbol{u}||\boldsymbol{v}) = \sum_{j=1}^{K} u_j \log \frac{u_j}{v_j} \tag{3.2}$$

The KL-divergence is not a metric since it is not symmetric under the interchange of $\boldsymbol{u}$ and $\boldsymbol{v}$ but satisfies all the properties of a monotonic distance measure. Specifically, the KL-divergence satisfies [121]:

- $\mathrm{KL}(\boldsymbol{u}||\boldsymbol{v}) = 0 \implies \boldsymbol{u} = \boldsymbol{v}$.
- $\mathrm{KL}(\boldsymbol{u}||\boldsymbol{v}) \geq 0$ for all $\boldsymbol{u}, \boldsymbol{v} \in \Delta^{K-1}$.
- $\mathrm{KL}(T(\boldsymbol{u})||T(\boldsymbol{v})) \leq \mathrm{KL}(\boldsymbol{u}||\boldsymbol{v})$ for every stochastic map T. [1]

3.1.1.1 Classical Fisher Information Matrix

In our analysis below, we will assume that the measurement basis $\mathcal{M}$ is fixed. Thus, the resulting probability distribution will only depend on the choice of parameters $\boldsymbol{\theta}$. Let $p_{\mathcal{M}}(\boldsymbol{\theta})$ be the probability distribution after measuring the state $|\psi(\boldsymbol{\theta})\rangle$ and $p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})$ be the probability distribution after measuring the state $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$. If the shift vector $\boldsymbol{\epsilon}$

[1] A stochastic map T is defined as the map between probability distributions. Stochastic maps preserve or reduce the amount of information available to distinguish the distributions.

is small, we can Taylor expand the KL-divergence as:

$$\begin{aligned}\mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) = \mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta})) + \sum_{i=1}^{m}\epsilon_i \frac{\partial \mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon}))}{\partial \epsilon_i}\Big|_{\boldsymbol{\epsilon}=0} \\ + \frac{1}{2}\sum_{i,j=1}^{m}\epsilon_i\epsilon_j \frac{\partial^2 \mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon}))}{\partial \epsilon_i \partial \epsilon_j}\Big|_{\boldsymbol{\epsilon}=0} + \mathcal{O}(\|\boldsymbol{\epsilon}\|_1^3) \implies \\ \mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) \approx \frac{1}{2}\boldsymbol{\epsilon}^T[\mathcal{F}_C^V(\boldsymbol{\theta})]\boldsymbol{\epsilon} = \frac{1}{2}\|\boldsymbol{\epsilon}\|^2_{\mathcal{F}_C^V}\end{aligned} \tag{3.3}$$

where the first term in Eq. (3.3) is zero since $\mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta})) = 0$ and the second term is also zero since it corresponds to a minimum[2]. We also neglected the third-order terms in the last line. We can show that, for the choice of KL-divergence as a distance measure, the elements of the CFIM can be calculated as:

$$[\mathcal{F}_C^V(\boldsymbol{\theta})]_{ij} = \sum_{\boldsymbol{s}} \frac{1}{p_{\boldsymbol{s}}^V(\boldsymbol{\theta})} \frac{\partial p_{\boldsymbol{s}}^V(\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial p_{\boldsymbol{s}}^V(\boldsymbol{\theta})}{\partial \theta_j} \tag{3.4}$$

For completeness, we have added the CFIM derivation in Appendix A.1, but one can also find it in [32]. It is also worth noting that if a different distance measure was chosen, then it would always return a constant multiple of the CFIM as long as the choice of the distance measure is *monotonic* [122].

3.1.1.2 Measuring the Classical Fisher Information Matrix

Consider the parameterized shift rules discussed in Sec. 2.1 and also consider the observable $O = \Pi_{\boldsymbol{s}} = |\boldsymbol{s}\rangle\langle\boldsymbol{s}|$. Then, the CFIM elements, given by Eq. (3.4), can be calculated as:

$$\begin{aligned}[\mathcal{F}_C^V]_{ij} = \sum_{\boldsymbol{s}} \frac{r^2}{p_{\boldsymbol{s}}^V(\boldsymbol{\theta})} \Big[p_{\boldsymbol{s}}^V(\boldsymbol{\theta} + \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{i}}) - p_{\boldsymbol{s}}^V(\boldsymbol{\theta} - \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{i}})\Big] \\ \times \Big[p_{\boldsymbol{s}}^V(\boldsymbol{\theta} + \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}}) - p_{\boldsymbol{s}}^V(\boldsymbol{\theta} - \frac{\pi}{4r}\hat{\boldsymbol{e}}_{\boldsymbol{j}})\Big]\end{aligned} \tag{3.5}$$

We can see that the elements of the CFIM can be expressed as products of first-order derivatives. As such, we can introduce Corollary 3.1 that quantifies the classical and quantum resources needed for the calculation of the CFIM.

Corollary 3.1. *Consider a parameterized quantum circuit consisting of m parameterized quantum gates. Any classical Fisher information matrix (CFIM) requires $\mathcal{O}_Q(m)$ different quantum state preparations and $\mathcal{O}_C(m^2)$ classical resources to post-process the measurements and store the matrix.*

[2]The KL-divergence is non-negative in general, and zero at $\boldsymbol{\epsilon} = 0$.

As we discuss later, this results in the CFIM requiring quadratically less quantum resources than the QFIM.

3.1.2 Distance of pure density operators

Just as we defined a measure of distance in the space of probability distributions, we could also measure distances in the space of density operators. In this thesis, we focus only on *pure* quantum states ($\operatorname{tr}\rho^2 = 1$).

As we discussed for the classical case, there is a unique underlying metric in the space of probability distributions independent of the choice of distance measure. Different distance measures will always yield a constant multiple of the CFIM. However, this is not true in the quantum case as Petz [123] proved that there exist infinitely many metrics. If we restrict ourselves to the space of *pure* quantum states there is a unique underlying metric, independent of the choice of the distance measure [30]. As a distance measure, we choose the *infidelity* between pure quantum states which for two quantum states $|\psi(\boldsymbol{\theta})\rangle$, $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$) is defined as:

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) = 1 - |\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle|^2 \tag{3.6}$$

3.1.2.1 Quantum Fisher Information Matrix

Let $|\psi(\boldsymbol{\theta})\rangle$ and $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$ be two parameterized quantum states. If the shift vector $\boldsymbol{\epsilon}$ is small, then we can Taylor expand the infidelity as:

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) = d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta})\rangle) + \sum_{i=1}^{m} \epsilon_i \frac{\partial d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle)}{\partial \epsilon_i}\bigg|_{\boldsymbol{\epsilon}=0} + \frac{1}{2}\sum_{i,j=1}^{m} \epsilon_i\epsilon_j \frac{\partial^2 d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle)}{\partial \epsilon_i \partial \epsilon_j}\bigg|_{\boldsymbol{\epsilon}=0} + \mathcal{O}(\|\boldsymbol{\epsilon}\|_1^3)$$

where if we neglect third order terms we can write

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) \approx \frac{1}{4}\boldsymbol{\epsilon}^T[\mathcal{F}_Q(\boldsymbol{\theta})]\boldsymbol{\epsilon} = \frac{1}{4}\|\boldsymbol{\epsilon}\|^2_{\mathcal{F}_Q} \tag{3.7}$$

where similarly to the classical relative entropy the first two terms vanish because $d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta})\rangle)$ corresponds to a minimum with a value equal to zero. The matrix $\mathcal{F}_Q(\boldsymbol{\theta})$ is called the *quantum Fisher information matrix* (QFIM) and acts as a metric in the space of quantum states giving information about the geometry of states near the

vicinity of the state $|\psi(\boldsymbol{\theta})\rangle$. The matrix elements of the QFIM are calculated to be the real part of the *Fubini-Study metric* (see [30, 32]):

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = 4 \operatorname{Re}\left[\frac{\partial \langle\psi(\boldsymbol{\theta})|}{\partial\theta_i}\frac{\partial |\psi(\boldsymbol{\theta})\rangle}{\partial\theta_j} - \frac{\partial \langle\psi(\boldsymbol{\theta})|}{\partial\theta_i}|\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})|\frac{\partial |\psi(\boldsymbol{\theta})\rangle}{\partial\theta_j}\right] \tag{3.8}$$

Intuitively, the QFIM acts as a metric in the space of quantum states. It provides a description of the geometry of the underlying state, giving information on how the parameterized quantum state changes if we vary a given parameter. Large eigenvalues of the QFIM will result in significant changes in the quantum state (with respect to a distance measure) even for small variations towards the direction of the corresponding eigenvector. On the other hand, zero eigenvalues will correspond to *singularities*, i.e. points in the space of parameters where changes will have no effect on the underlying quantum state [34].

3.1.2.2 Measuring the Quantum Fisher Information Matrix

So far in the literature, there has been extensive research on how to calculate the elements of QFIM given by Eq. (3.8). In [67] explained how to use parameter-shift rules to calculate QFIM while [70] introduced stochastic parameter-shift rules. On the other hand, [34] proposed an alternative way with Hadamard-overlap using an extra ancilla qubit. Overall, the quantum and classical resources needed for the calculation of the QFIM are stated in Corollary 3.2.

Corollary 3.2. *Consider a parameterized quantum circuit, consisting of m parameterized quantum gates. The quantum Fisher information matrix at any parameter configuration $\boldsymbol{\theta}$ requires $\mathcal{O}_Q(m^2)$ different quantum state preparations and $\mathcal{O}_C(m^2)$ classical resources to post-process the measurement and store the matrix.*

3.1.3 Quantum Natural Gradient

The Quantum Natural Gradient (QNG) [30] is an optimization algorithm suited for variational quantum algorithms. Instead of updating the parameters in the direction of the negative gradient (of the loss function), the algorithm considers the changes happening in the space of parameterized quantum states. Specifically, at each iteration, the parameters are changed according to the update rule:

$$\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} - \eta\mathcal{F}_Q(\boldsymbol{\theta_k})^{+}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta_k}) \tag{3.9}$$

where $\mathcal{F}_Q(\boldsymbol{\theta_k})^+$ is the *Moore-Penrose inverse* (pseudoinverse) [3] of the *quantum Fisher information matrix* (QFIM). Intuitively, QNG performs large steps in the directions where the state changes by a small amount and takes smaller steps in the directions where the state changes by a large amount. Although the simulated experiments in [30] showed that the convergence speed (in terms of optimization iterations) is improved significantly compared to first-order local optimizers, the bottleneck is that it requires $\mathcal{O}_Q(m^2)$ state preparations at each iteration which results in a big drawback for near term devices. This has the immediate implication that the quantum resources needed to implement the algorithm are quite large, limiting its actual practicality. We will discuss that thoroughly in the next sections.

3.1.3.1 Steepest Descent

At this point, it would be fruitful to provide a more geometric explanation behind the update rule (3.9) of QNG and its approximation through the update rule of Eq. (3.37). Consider the first-order Taylor expansion of loss function $\mathcal{L}(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta} + \boldsymbol{v}$:

$$\mathcal{L}(\boldsymbol{\theta} + \boldsymbol{v}) \approx \mathcal{L}(\boldsymbol{\theta}) + \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})^T \boldsymbol{v} \tag{3.10}$$

Steepest descent [124] aims to find a direction $\boldsymbol{v}$ such that the directional derivative becomes as small as possible. Since $\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})^T \boldsymbol{v}$ is linear in $\boldsymbol{v}$, it can be made as small as we desire. So to make the question sensible, we define the *normalized steepest descent* as:

$$\Delta\boldsymbol{\theta}_{\text{nsd}} = \arg\min_{\boldsymbol{v}:\|\boldsymbol{v}\|=1} \{\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})^T \boldsymbol{v}\} \tag{3.11}$$

where $\|\cdot\|$ can be any vector norm. For example, choosing $\|\cdot\| = \|\cdot\|_2$ (the l_2-norm) the steepest descent becomes gradient descent or choosing $\|\cdot\| = \|\cdot\|_1$ (the l_1-norm) becomes *coordinate descent* [124]. We are interested in the case where $\|\cdot\|$ is chosen to be $\|\cdot\|_P$, where $P \succcurlyeq 0$ describes the intrinsic geometry in the parameterized space (or at least an approximation of it). Notice that the underlying geometry of the parameters of a parameterized quantum circuit is not *Euclidean*, but follows a *Riemannian structure*. That is, the distance between vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \delta\boldsymbol{\theta}$ is:

$$d(\boldsymbol{\theta}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})^2 = \delta\boldsymbol{\theta} G(\boldsymbol{\theta}) \delta\boldsymbol{\theta} = \|\delta\boldsymbol{\theta}\|^2_{G(\boldsymbol{\theta})} \tag{3.12}$$

where $G(\boldsymbol{\theta})$ is the *Riemannian metric*. Since parameterized quantum circuits generate parameterized quantum states $|\psi(\boldsymbol{\theta})\rangle$, the actual geometry of the parameters is characterized by changes happening in the quantum state. As a result, the corresponding

[3] The *pseudoinverse* of a matrix A corresponds to the inverse of A that is defined on the space orthogonal to the kernel.

metric that describes the geometry of the parameters at a point $\boldsymbol{\theta}$ is the QFIM, i.e. $G(\boldsymbol{\theta}) = \mathcal{F}_Q(\boldsymbol{\theta})$.

Lemma 3.1. *Consider a parameterized quantum circuit parameterized by a vector $\boldsymbol{\theta}$ that generates a quantum state $|\psi(\boldsymbol{\theta})\rangle$. The normalized steepest descent direction of the loss function $\mathcal{L}(\boldsymbol{\theta})$ is:*

$$-\frac{\mathcal{F}_Q^{-1}(\boldsymbol{\theta})\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})}{\left\|\mathcal{F}_Q^{-1/2}(\boldsymbol{\theta})\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})\right\|_2} \tag{3.13}$$

where $\mathcal{F}_Q(\boldsymbol{\theta})$ is the QFIM at point $\boldsymbol{\theta}$

Proof. In the case of a parameterized quantum circuit, the local geometry of the parameterized space is described by the QFIM. As such, distances (near a point $\boldsymbol{\theta}$) are measured with respect to the $\|\cdot\|_P$, with $P = \mathcal{F}_Q(\boldsymbol{\theta})$. Then, the normalized steepest descent direction is given by:

$$\Delta\boldsymbol{\theta}_{\text{nsd}} = \arg\min_{\boldsymbol{v}:\|\boldsymbol{v}\|_{\mathcal{F}_Q}=1} \{\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})^T\boldsymbol{v}\} \tag{3.14}$$

We can make use of the substitution $\boldsymbol{u} = \mathcal{F}_Q^{1/2}\boldsymbol{v}$. In this case, the constraint $\|\boldsymbol{v}\|_{\mathcal{F}_Q} = 1$ is replaced by:

$$\|\boldsymbol{v}\|_{\mathcal{F}_Q} = 1 \implies \|\boldsymbol{u}\|_2 = 1 \tag{3.15}$$

As such, solving Eq. (3.14) is equivalent to:

$$\Delta\boldsymbol{\theta}_{\text{nsd}} = \arg\min_{\boldsymbol{u}:\|\boldsymbol{u}\|_2=1} \{\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})^T\mathcal{F}_Q^{-1/2}\boldsymbol{u}\} \tag{3.16}$$

Thus, by solving Eq. (3.16) and making again the substitution $v = \mathcal{F}_Q^{-1/2}u$ we find that the steepest descent direction is:

$$-\frac{\mathcal{F}_Q^{-1}(\boldsymbol{\theta})\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})}{\left\|\mathcal{F}_Q^{-1/2}(\boldsymbol{\theta})\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})\right\|_2}$$

□

3.1.4 Quantum Imaginary-Time Evolution

In Quantum Mechanics, when a system is initialized in the quantum state $|\phi(0)\rangle$ (at time $t = 0$) and its dynamics are described by a time-independent Hamiltonian H, it will evolve under the unitary e^{-iHt}, i.e.:

$$|\phi(t)\rangle = e^{-iHt}|\phi(0)\rangle \tag{3.17}$$

Such an evolution can be simulated in a gate-based quantum computer by "trotterizing" the unitary evolution e^{-iHt} into small time intervals δt.

If we allow the time to take imaginary values ($\tau \equiv it, \tau \in \mathbb{R}$) then the operator $e^{-H\tau}$ is no longer unitary and the evolution is called *quantum imaginary-time evolution*. As a first step, we will derive the mathematical equation that governs the imaginary-time evolution. Consider the imaginary-time evolved state $|\phi(\tau)\rangle$:

$$|\phi(\tau)\rangle = A(\tau)e^{-H\tau} |\phi(0)\rangle \tag{3.18}$$

where:

$$A(\tau) = \left(\frac{1}{\sqrt{\langle\phi(0)| e^{-2H\tau} |\phi(0)\rangle}} \right) \tag{3.19}$$

is a normalization factors so that $\langle\phi(\tau)|\phi(\tau)\rangle = 1$. The evolution under the imaginary-time evolution is governed by the *Wick-Schrödinger* equation. To see this, we take the time derivative:

$$\frac{\partial |\phi(\tau)\rangle}{\partial \tau} = \frac{\partial}{\partial \tau}\Big(A(\tau)e^{-H\tau} |\phi(0)\rangle \Big) =$$
$$\frac{\partial A(\tau)}{\partial \tau} e^{-H\tau} |\phi(0)\rangle + A(\tau)\frac{\partial e^{-H\tau}}{\partial \tau} |\phi(0)\rangle$$

Computing the derivative in the first term, we obtain:

$$\frac{\partial A(\tau)}{\partial \tau} = \frac{\partial}{\partial \tau}\left(\frac{1}{\sqrt{\langle\phi(0)| e^{-2H\tau} |\phi(0)\rangle}} \right) = A(\tau)E_\tau \tag{3.20}$$

where $E_\tau = \langle\phi(\tau)| H |\phi(\tau)\rangle$. Thus, putting everything back together we obtain the *Wick-Schrödinger equation*:

$$\frac{\partial |\phi(\tau)\rangle}{\partial \tau} = (E_\tau - H) |\phi(\tau)\rangle \tag{3.21}$$

As we discussed, imaginary-time evolution is a very interesting tool that allows the preparation of thermal states [114, 125, 126] or ground states [12, 17, 127]. The necessary condition is that the initial state is prepared with a non-zero overlap with the ground state of the Hamiltonian of interest. To see this, consider a Hamiltonian H (with non-negative spectrum) and an initial state $|\phi(0)\rangle$ that has a non-zero overlap with the ground state $|\phi_0\rangle$. We can write the initial state in the energy eigenbasis as:

$$|\phi(0)\rangle = a_0 |\phi_0\rangle + \sum_{j\neq 0} a_j |\phi_j\rangle \tag{3.22}$$

where $|\phi_0\rangle$ is the ground state. Evolving the state according to the quantum imaginary-time evolution will result in the quantum state:

$$\begin{aligned} |\phi(\tau)\rangle &= A(\tau)\Big[a_0 e^{-H\tau}|\phi_0\rangle + \sum_{j\neq 0} a_j e^{-H\tau}|\phi_j\rangle\Big] \\ &= A(\tau)\Big[a_0 e^{-E_0\tau}|\phi_0\rangle + \sum_{j\neq 0} a_j e^{-E_j\tau}|\phi_j\rangle\Big] \end{aligned} \tag{3.23}$$

As a result, in the limit of $\tau \to \infty$ the system reaches the ground state because the terms corresponding to higher energies (i.e. $e^{-E_j\tau}$) go faster to zero.

In our case, we are equipped with a small-scale (and perhaps noisy) quantum computer and we aim to approximate the exact imaginary-time evolution described by the states $|\phi(\tau)\rangle$ by a family of parameterized state $|\psi(\boldsymbol{\theta}(\tau)\rangle$ [12] which approximate the former states as much as possible. In other words, we aim to find the parameter dynamics $\boldsymbol{\theta}(\tau)$ so that the parameterized state approximates the imaginary-time evolution. Starting from McLachlan's variational principle:

$$\delta\|(d/d\tau + H - E_\tau)\,|\psi(\boldsymbol{\theta}(\tau))\rangle\|_2 = 0 \tag{3.24}$$

and introducing a time-dependent global phase in the calculation [128], we find (see [12, 128] for details) that the parameters must satisfy:

$$\mathcal{F}_Q(\boldsymbol{\theta}(\tau))\dot{\boldsymbol{\theta}} = -2\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_\tau(\boldsymbol{\theta}(\tau)) \tag{3.25}$$

where $\mathcal{F}_Q$ is the quantum Fisher information matrix defined in Eq. (3.8). As we previously discussed, one of the major drawbacks is that the evaluation of Eq. (3.25) at a certain point requires the preparation of $\mathcal{O}(m^2)$ quantum states, scaling quadratically with the number of parameters. As it is clear, updating the parameters according to Eq. (3.25) for a sufficiently small time step is equivalent to the QNG update rule in Eq. (3.9).

3.2 Random Natural Gradient

In this section, we will outline our first main result, which is a novel optimization algorithm called *Random Natural Gradient* (RNG). The update rule of RNG is given by the formula:

$$\begin{gathered} \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta[\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta}_k)]^{+}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta}_k) \\ \text{Random measurement } \mathcal{M} \end{gathered} \tag{3.26}$$

where the random measurement $\mathcal{M}$ is performed by first applying a random unitary V on the parameterized state $|\psi(\boldsymbol{\theta}_k)\rangle$ and then measuring on the computational basis. We can assume that this unitary is parameterized by a $k = \mathcal{O}(\text{poly}(n))$-dimensional vector (i.e., it is comprised of a series of parameterized gates) $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_k)$. In our analysis, we choose the random unitaries to be hardware-efficient parameterized quantum circuits for which we uniformly sample the parameters of the parameterized gates. In other words, our algorithm replaces the QFIM ($\mathcal{F}_Q$) in the QNG update rule in Eq. (3.9) by a random CFIM ($\mathcal{F}_C^{\mathcal{M}}$).

The intuition behind this update rule is relatively simple and is thoroughly explained in the following sections. As we will explain, the optimization part in VQAs requires an update rule that can be calculated efficiently for any hope of quantum advantage. Additionally, the learning rate should carry some information about the underlying parameterized state and how an infinitesimal change in a parameter would change it.

First of all, we can visualize measuring on a random basis and constructing the CFIM as a way to approximate the QFIM. Similarly, the authors of [30] suggested that instead of calculating the QFIM (as it would result in a tedious calculation), one could calculate only block-diagonal elements. This strategy would require fewer quantum resources but carries no physical intuition on why such an approximation would be valid, especially when there are high correlations between elements of different blocks.

Finding the measurement basis that would result in the optimal step is not a task that can be calculated efficiently. For that reason, one could draw a measurement basis at random and then construct the CFIM on that basis. Our experiments showed that a random CFIM has an increased rank (e.g. compared to measurements in Z-basis), and as such, more directions can be explored during the optimization (see Figure 3.4). This has the further implication that the classical optimization part may avoid local minima as was explored in the QNG case in [31]. Similar findings were also found in [129], where the noise may increase the rank of QFIM, so more directions can be explored.

From a practical perspective, the update rule in Eq. (3.26) can be calculated efficiently (see Corollary 3.1) and, as we will show, improves the convergence dramatically. At each time step, the quantum resources (the number of quantum state preparations) needed are $2m$ for the gradient calculation and $2m + 1$ for the calculation of the CFIM on a random basis (see Eq. (3.5)) for a total of $4m + 1$ quantum states. Then, the classical computer post-processes the $2m + 1$ random basis measurements and utilizes a classical

memory of size $m \times m$ for the random CFIM. This update rule is iteratively applied with a different measurement basis until convergence to a local (or a global) minimum occurs.

Furthermore, our method inherits the advantage that the depth of the quantum circuit required to calculate the matrix elements of the random CFIM is less than that of the QFIM. In RNG, the depth of the additional unitary that is required for the calculation of the random CFIM is user-dependent and as we show in our experiments in Sec. 3.8, the user can select shallow random quantum circuits and converge significantly faster than QNG.

Specifically, as we discussed in Sec. 3.1.2.2, for the calculation of QFIM one requires quantum circuits of depth twice the one needed to generate the parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$. This imposes a bottleneck for parameterized quantum states that require large depths. However, for RNG, the additional quantum circuit is user-specified and depends on the architecture required for the random measurement. Our algorithm is outlined in Algorithm 1.

Finally, we would like to stress how RNG avoids singularities in the parameter space [130]. Consider a fixed measurement basis $\mathcal{M}$. There are points in the parameters space where a small displacement in the parameters may not result in any change in the probabilities observed. This would result in a CFIM with degenerate zero eigenvalues. In a practical scenario close to such a point, a natural gradient optimizer will make very large steps, prohibiting it from convergence. However, by switching the basis we can now avoid the singularities as for the new observables, the previous point may result in completely different probability distributions, and as such the optimizer will continue making small steps.

3.3 Local Optimization

In this section, we provide the motivation and theoretical intuition behind the update rule of RNG in Eq. (3.26) and argue why such an update is desirable. *Local optimization* refers to the technique of following a trajectory in a region of the loss landscape and converging to a (possibly local) minimum. Consider the loss function to be the expectation value of the energy of a parameterized quantum state:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathrm{tr}(\rho(\boldsymbol{\theta})H) \tag{3.27}$$

Algorithm 1: Random Natural Gradient

Input : Problem Hamiltonian H;
Ansatz family $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle$;
Total iterations K;
Loss function $\mathcal{L}(\boldsymbol{\theta})$;
Initial parameters $\boldsymbol{\theta} = \boldsymbol{\theta_0}$;
Learning rate η;
for $k = 1, 2, \ldots, K$ **do**
  Calculate derivatives $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_i}\ \forall\ i \in [M]$;
  Shuffle a measurement basis $\mathcal{M}$;
  Calculate the Classical Fisher Information Matrix $\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta})$;
  Update $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta[\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta})]^{+}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})$;
end
return $\boldsymbol{\theta}$

where H is the Hamiltonian of the problem. Vanilla Gradient Descent (GD) iteratively updates the parameters $\boldsymbol{\theta}$ by following the direction of the negative gradient. The update rule is given by:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \boldsymbol{\nabla}\mathcal{L}(\theta_k) \tag{3.28}$$

where $\eta > 0$ is a tunable hyperparameter that is crucial for both the speed and convergence of the algorithm. The biggest bottleneck in applying GD in a VQA setting is that a small choice of η will require numerous quantum state preparations in the quantum computer, especially in the region where $\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta}) \to 0$, while a large choice of η may result in "missing" the minimum. In the former case, the quantum computer will require a very large number of circuit repetitions to acquire the desired accuracy, but also multiple and different quantum state preparations. A useful tool from the classical optimization literature is the *proximal point method* [131] where the update rule is given by:

$$\boldsymbol{\theta_{k+1}} = \mathrm{prox}_{\mathcal{L},\lambda}(\boldsymbol{\theta_k}) = \arg\min_{\boldsymbol{\theta}} \left[\mathcal{L}(\boldsymbol{\theta}) + \lambda q(\boldsymbol{\theta}, \boldsymbol{\theta_k})\right] \tag{3.29}$$

where $q(\boldsymbol{\theta}, \boldsymbol{\theta_k})$ is a *dissimilarity function* that measures distance between the two vectors $\boldsymbol{\theta}, \boldsymbol{\theta_k}$. When q is chosen to be the squared Euclidean distance:

$$q(\boldsymbol{\theta}, \boldsymbol{\theta_k}) = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta_k}\|_2^2 \tag{3.30}$$

then the proximal update becomes the ordinary GD given by the update rule in Eq. (3.28) with $\eta = \lambda^{-1}$. In that case, the dissimilarity function acts as a penalty term that prohibits big steps in the space of parameters.

In [30] the authors claimed that the classical optimization algorithm should adapt the updated parameters according to the changes happening in the state-space i.e. update $\boldsymbol{\theta}$ according to:

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} \left[\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2\eta} \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}}\|^2_{\mathcal{F}_Q} \right] \tag{3.31}$$

where the term $\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}}\|^2_{\mathcal{F}_Q} \equiv (\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}})^T \mathcal{F}_Q (\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}})$ penalizes large steps in the state-space[4]. In this case, the update rule (3.29) is reformulated to the *Quantum Natural Gradient* (QNG) where the parameters are iteratively changed according to the rule given by Eq. (3.9).

At this point, we would like to state that the QNG update rule in Eq. (3.9) falls into the more general category of *preconditioning*. In general, preconditioning the GD update in Eq. (3.28) by a positive definite matrix A:

$$\begin{gathered} \boldsymbol{\theta}_{\boldsymbol{k+1}} = \boldsymbol{\theta}_{\boldsymbol{k}} - A^{-1} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}}) \\ A \succ 0 \end{gathered} \tag{3.32}$$

results in a *descent direction*. To see this, consider the Taylor expansion of the loss function (3.27) around the current iterate $\boldsymbol{\theta}_{\boldsymbol{k}}$:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}}) + \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}})^T (\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}}) + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}})^T \mathbf{H} (\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}}) + \mathcal{O}(\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{k}}\|^3) \tag{3.33}$$

where $\mathbf{H}$ is the Hessian at the point $\boldsymbol{\theta}_{\boldsymbol{k}}$ and let the updated point be $\boldsymbol{\theta}_{\boldsymbol{k}} - A^{-1} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}})$. Keeping only the first-order terms (this would be valid for example if we are in a region with small gradients or if we scale the matrix A by a small factor η), then the loss function is:

$$\begin{gathered} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}} - A^{-1} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}})) = \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}}) - \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}})^T A^{-1} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}}) \\ \implies \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}} - A^{-1} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}})) < \mathcal{L}(\boldsymbol{\theta}_{\boldsymbol{k}}) \end{gathered} \tag{3.34}$$

since the second term in the first line is negative for $A^{-1} \succ 0$. By keeping second-order terms, the condition so that the preconditioner A points towards a descent direction becomes [5]:

$$A^{-1} \mathbf{H} A^{-1} \prec 2A^{-1} \tag{3.35}$$

The above analysis can readily be formulated in the case where the preconditioner is *a positive semidefinite matrix*, but the inverse is replaced by the Moore-Penrose inverse. In

[4] Here $\mathcal{F}_Q$ acts a metric, stretching the vectors in the state-space accordingly.

[5] We write $B \prec C$ to denote the matrix $B - C$ being negative definite.

this case, we observe two things. The first is that moving towards a descent direction is feasible when the matrix A^{-1} is small (with respect to a matrix norm), which can always be done by multiplying by a sufficiently small scalar η. However, choosing a matrix that is computationally expensive to calculate and then scaling it by a small factor η (see QNG update in Eq. (3.9) and Corollary 3.2) may prohibit any advantage of using the preconditioner in the first place. On the other hand, condition (3.35) filters a large amount of positive definite matrices that allow for a descent direction, but testing the condition in an online setting is impractical since it requires the calculation of the Hessian at every iterate.

We argue here that the preconditioner should carry information about the intrinsic geometry of the parameters (just like in QNG) but at the same time, it should be relatively fast to calculate. As we propose in Sec. 3.2, a clever way to feed information about changes happening in the quantum state in a positive semidefinite matrix is to use random measurements. This alternative allows for a fast calculation of a positive semidefinite matrix, which is intrinsically meaningful and improves the convergence in a VQA framework.

3.3.1 Classical Natural Gradient

In Sec. 3.3 we introduced the idea of proximal updates where the parameters are updated in a way that considers a distance measure of the parameters. Similarly to QNG, one could prepare a quantum state, measure it (e.g. on the computational basis), and, with the measurement outcomes, approximate the probability distribution of different outcomes. In that case, we can choose the dissimilarity function to be the KL-divergence (see Sec. 3.1.1) between the probability distributions after measuring the quantum states at the computational basis.

However, nothing prevents us from using a different dissimilarity function by switching the measurement onto a different basis (possibly a random one). As such, if $\mathcal{M}$ is the measurement basis, then the proximal point method will become:

$$\boldsymbol{\theta_{k+1}} = \text{prox}_{\mathcal{L},\lambda,\mathcal{M}}(\boldsymbol{\theta_k}) = \arg\min_{\boldsymbol{\theta}} \left[\mathcal{L}(\boldsymbol{\theta}) + \lambda q_{\mathcal{M}}(\boldsymbol{\theta}, \boldsymbol{\theta_k})\right] \tag{3.36}$$

In that case, the update rule will be reformulated to:

$$\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} - \eta \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})^{+} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta_k}) \tag{3.37}$$

where $\mathcal{F}_C^{\mathcal{M}}$ is the classical Fisher information matrix constructed by performing measurements on the $\mathcal{M}$ basis. When the measurement basis is chosen to be the computational basis state, then the update rule (3.37) is referred to in the classical optimization literature as the *Natural Gradient Descent* (NGD). However, in a quantum setting, the CFIM is basis-dependent, and this property offers a significant computational advantage over QNG.

From the previous discussion, we can immediately introduce Corollary 3.3, which provides a condition so that the update (3.37) results in a descent direction. The condition under which a CFIM (constructed by measurements on a basis $\mathcal{M}$) preconditions a gradient descent direction and results in a decrease in the loss function is as follows.

Corollary 3.3. *Consider the update rule* (3.37) *and let* $\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})$ *be the CFIM constructed by performing measurements in the* $\mathcal{M}$ *basis. Then, the updated direction will result in a descent direction (*$\mathcal{L}(\boldsymbol{\theta_{k+1}}) \leq \mathcal{L}(\boldsymbol{\theta_k})$*) as long as:*

$$\eta \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})^+ \boldsymbol{H} \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})^+ \preccurlyeq 2\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})^+ \tag{3.38}$$

for some $\eta > 0$*, where* $\boldsymbol{H}$ *is the Hessian of the loss function* $\mathcal{L}$ *at the point* $\boldsymbol{\theta_k}$*.*

Corollary 3.3 provides the condition so that a tuple $(\mathcal{F}_C^{\mathcal{M}}, \eta)$ will result in a decrease of the loss function. However, testing the condition in Eq. (3.38) cannot be feasibly implemented in a practical setting as it requires the calculation of the Hessian at each point $\boldsymbol{\theta_k}$. As such, we would have to rely on empirical choices for the choice of the hyperparameter η.

It is true that all directions that leave the quantum state invariant under translations of the parameters will also leave all probability distributions unaffected. The former directions correspond to the eigenvectors of the QFIM that belong in its *null space* while the latter directions correspond to eigenvectors of different classical Fisher information matrices. We can thus show (see Proposition 3.1) that all zero eigenvalues of the QFIM correspond to zero eigenvalues of any CFIM (but not vice versa).

Proposition 3.1. *The null space of the quantum Fisher information matrix* $\mathcal{N}(\mathcal{F}_Q)$ *is a subspace of the null space of the classical Fisher information matrix* $\mathcal{N}(\mathcal{F}_C^{\mathcal{M}})$ *over any measurement collection* $\mathcal{M}$ *at a fixed point* $\boldsymbol{\theta}$

$$\mathcal{N}(\mathcal{F}_Q) \subseteq \mathcal{N}(\mathcal{F}_C^{\mathcal{M}}) \tag{3.39}$$

Proof. The infidelity between two parameterized quantum states $|\psi(\boldsymbol{\theta})\rangle$ and $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$ is given by (see Eq. (3.7)):

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) = \frac{1}{2}\boldsymbol{\epsilon}^T \mathcal{F}_Q \boldsymbol{\epsilon}$$

where $\mathcal{F}_Q$ is the QFIM at point $\boldsymbol{\theta}$. Consider now the eigenvalue decomposition of $\mathcal{F}_Q$:

$$\mathcal{F}_Q = UDU^T \tag{3.40}$$

where U is a unitary matrix with its columns being the normalized eigenvectors of $\mathcal{F}_Q$ and D the diagonal matrix with the eigenvalues of $\mathcal{F}_Q$ as its entries. The distance between the two states can then be written as:

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) = \frac{1}{2}\boldsymbol{\epsilon}^T UDU^T \boldsymbol{\epsilon} \tag{3.41}$$

Now, if we assume that $\boldsymbol{\epsilon} = d\alpha \boldsymbol{v_i}$ where $\boldsymbol{v_i}$ is an eigenvector of $\mathcal{F}_Q$ with eigenvalue λ_i, then the distance can be written as:

$$d_f(|\psi(\boldsymbol{\theta})\rangle, |\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle) = \frac{\lambda_i}{2}(d\alpha)^2 \tag{3.42}$$

Thus, all infinitesimal changes $\delta\boldsymbol{u} \in \mathcal{N}(\mathcal{F}_Q)$ that belong in the null space of $\mathcal{F}_Q$ will not result in a change of the underlying quantum state. A direct consequence is that since the quantum states remain invariant under these translations, the probability distributions over any measurements will also remain unchanged. As such, since the probability distributions do not change for small displacements $\delta\boldsymbol{u}$, then $\delta\boldsymbol{u} \in \mathcal{N}(\mathcal{F}_C^{\mathcal{M}})$ and thus:

$$\mathcal{N}(\mathcal{F}_Q) \subseteq \mathcal{N}(\mathcal{F}_C^{\mathcal{M}})$$

□

At this point, we would like to stress that the converse is not true. CFIM (obtained by a general measurement) may have zero eigenvalues that are not zero eigenvalues of the QFIM. This also implies that different measurements lead to CFIM having different null spaces, with only a guarantee that all of them contain the null space of the QFIM. It follows that some measurements lead to CFIMs that carry more information about changes happening in the quantum state and are closer to the QFIM than other measurements.

Finally, we can see that as in our case, a direction $\propto \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta_k})^+ \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})$ points towards the steepest descent direction (see sec. 3.1.3.1) in the Riemannian space whose metric is

the CFIM constructed by performing measurements in the $\mathcal{M}$ basis. But as we discuss in Sec. 3.4, all $[\mathcal{F}_C^{\mathcal{M}}]$ are information matrices that carry partial local information of the quantum state with respect to the choice of measurements. In other words, all CFIMs can be seen as providing *local approximations of the geometry of the underlying state-space* with the quality of the approximation determined by its distance from the QFIM.

3.4 Optimal Measurement

Consider again two parameterized quantum states, $|\psi(\boldsymbol{\theta})\rangle$, $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$ that differ by a small shift vector $\boldsymbol{\epsilon} \in \mathbb{R}^m$. Suppose we perform a measurement on a basis $\mathcal{M}$. As we have already seen, the distance of the corresponding probability distributions $p_{\mathcal{M}}(\boldsymbol{\theta})$, $p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})$ can be written as:

$$\mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) = \frac{1}{2}\boldsymbol{\epsilon}^T \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta})\boldsymbol{\epsilon} \tag{3.43}$$

Our goal is to choose a measurement basis $\mathcal{M}$ that will extract the maximum information from the state $|\psi(\boldsymbol{\theta})\rangle$. In the context of this chapter, *maximum information* refers to a measurement that will approximate as much as possible what happens locally in the space of quantum states.

Definition 3.1. (**Optimal measurement**) *We define the optimal measurement $\mathcal{M}^*$ as the measurement that when applied on the states $|\psi(\boldsymbol{\theta})\rangle$ and $|\psi(\boldsymbol{\theta}+\boldsymbol{\epsilon})\rangle$ it maximizes the distance between the probability distributions $p_{\mathcal{M}}(\boldsymbol{\theta})$ and $p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})$.*

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} \mathrm{KL}(p_{\mathcal{M}}(\boldsymbol{\theta})||p_{\mathcal{M}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) \tag{3.44}$$

As we discussed in Sec. 3.1.1 we can describe the possible measurements $\mathcal{M}(\boldsymbol{\phi})$ by first applying a unitary $V(\boldsymbol{\phi})$ on the parameterized state $|\psi(\boldsymbol{\theta})\rangle$ and then performing projective measurements on each qubit. If $\boldsymbol{\phi}^*$ are the angles that maximize the distance between probability distributions, then the following inequality holds

$$\begin{aligned} &\mathrm{KL}(p_{\mathcal{M}(\boldsymbol{\phi}^*)}(\boldsymbol{\theta})||p_{\mathcal{M}(\boldsymbol{\phi}^*)}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) \\ &\leq \mathrm{KL}(p_{\mathcal{M}^*}(\boldsymbol{\theta})||p_{\mathcal{M}^*}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) \end{aligned} \tag{3.45}$$

where the equality is true whenever there exists $\boldsymbol{\phi}^*$ such that $\mathcal{M}(\boldsymbol{\phi}^*) = \mathcal{M}^*$. We can use Eq. (3.43) and show that any CFIM can be upper bounded by the QFIM (see [32] for details) as :

$$\mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta}) \preccurlyeq \mathcal{F}_Q(\boldsymbol{\theta}) \; \forall \boldsymbol{\theta} \in \mathbb{R}^m \tag{3.46}$$

A natural question to ask is *"What is the appropriate measurement basis so that the resulting CFIM is optimal, in the sense that the CFIM approaches the QFIM with the least amount of error."*. The answer to this question is discussed below.

Consider a parameterized quantum circuit that generates parameterized quantum states $|\psi(\boldsymbol{\theta})\rangle$. Consider also the set of measurements that are generated by applying a unitary $V(\boldsymbol{\phi})$ on the state $|\psi(\boldsymbol{\theta})\rangle$ and then measuring in the computational basis.

Our starting point is Eq. (3.46). For every angle configuration $\boldsymbol{\phi}$, the matrix $(\mathcal{F}_C^{\mathcal{M}(\boldsymbol{\phi})}(\boldsymbol{\theta}) - \mathcal{F}_Q(\boldsymbol{\theta}))$ is negative semidefinite for every $\boldsymbol{\theta}$. As such, by taking the trace:

$$\begin{aligned} \operatorname{tr}\left(\mathcal{F}_C^{\mathcal{M}(\boldsymbol{\phi})}(\boldsymbol{\theta}) - \mathcal{F}_Q(\boldsymbol{\theta})\right) &\leq 0 \\ \operatorname{tr}\left(\mathcal{F}_C^{\mathcal{M}(\boldsymbol{\phi})}(\boldsymbol{\theta})\right) &\leq \operatorname{tr}\left(\mathcal{F}_Q(\boldsymbol{\theta})\right) \end{aligned} \tag{3.47}$$

As a result, the trace of QFIM provides an upper bound on the trace of every CFIM. We can thus conclude that the solution of the optimization problem:

$$\max_{\boldsymbol{\phi}} \operatorname{tr}\left(\mathcal{F}_C^{\mathcal{M}(\boldsymbol{\phi})}(\boldsymbol{\theta})\right) \tag{3.48}$$

will result in the CFIM corresponding to the optimal measurement (or else the optimal approximation of QFIM).

At this point, it is important to stress that except for the one-parameter case, there does not always exist a measurement basis $\mathcal{M}$ so that the CFIM is equal to QFIM [33, 132]. Specifically, in [133], the authors provide conditions under which there exists measurement so that the QFIM is saturated. However, if no such measurement exists, then the solution of Eq. (3.48) will result in a CFIM that approximates QFIM with the least amount of error:

$$\min_{\mathcal{M}} \left\| \mathcal{F}_C^{\mathcal{M}}(\boldsymbol{\theta}) - \mathcal{F}_Q(\boldsymbol{\theta}) \right\| \tag{3.49}$$

where $\|\cdot\|$ is any matrix norm. Moreover, as we discussed in Sec. 3.3, in a classical optimization scheme, achieving the optimal measurement may not have an immediate effect on the speed of convergence. Ideally, we would want to increase the number of directions in the probability distribution space so that the optimizer can have more directions to move. As we will discuss in Sec. 3.2, this is accomplished when the measurement basis is chosen at random.

The immediate advantage of identifying the optimal measurement is that provided that the resulting CFIM coincides with QFIM, the quantum natural gradient [30] can be

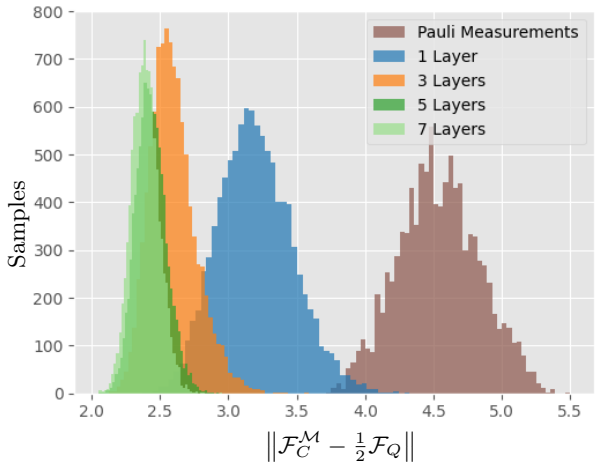

Figure 3.2: Distance of random CFIMs from QFIM. As the number of layers increases, sampling a random measurement tends to have a small distance from the QFIM and, as such, carries more information. A total number of 10000 CFIMs were calculated for each choice of measurement basis.

implemented using $\mathcal{O}(m)$ quantum states/resources instead of $\mathcal{O}(m^2)$. This results in a quadratic advantage in quantum resource requirement compared to previous methods. However, when the measurement operators are parameterized by a unitary $V(\boldsymbol{\phi})$, the optimization problem becomes non-convex. On top of that, the optimal measurement is $\boldsymbol{\theta}$-dependent, and as such, after every optimization iteration, the optimal measurement must be re-evaluated.

Remark. As we observed in our experiments, increasing the expressivity of the random parameterized measurements results in CFIMs that 1) tends towards $\frac{1}{2}\mathcal{F}_Q$, i.e. the error $\left\|\mathcal{F}_C^{\mathcal{M}} - \frac{1}{2}\mathcal{F}_Q\right\|$ is reduced and 2) the variance of the error (of any random CFIM) goes to zero.

Consider, for example, the parameterized quantum circuit in the left-hand side of Figure 2.2 for a system of 8 qubits and 3 layers. As it is illustrated in Figure 3.2, it is clear that a Pauli measurement cannot encapsulate the intrinsic changes happening in the quantum state. However, as we introduce random measurements, the random CFIM starts to approximate the QFIM, with the approximation becoming better when more expressive ansatz families are used. For the random measurements, we used the same type of circuit but with different Pauli rotations.

3.5 Estimators of the quantum Fisher information matrix

At this point, it would be illustrative to understand what happens when a collection of random measurements is performed on a parameterized state $|\psi(\boldsymbol{\theta})\rangle$. In this section, we will generalize our previous results in the case where multiple random measurements are used. We will show that multiple random measurements allow the estimation of the full QFIM and show how the user can use these estimators in the optimization process. As such, we will outline two estimators of the QFIM. For the second estimator, the Random Natural Gradient will correspond to a special case in which one random measurement is used. Experiments on how these estimators can be used in an optimization setting can be found in [13].

As we already discussed, QNG requires a fast calculation of the quantum Fisher information matrix (QFIM) or an approximation of it. In this section, we outline our results on the approximation of the QFIM using random measurements [54] and defer the proofs to the appendix A.3. Throughout the rest of the chapter, we will denote $\mathcal{F}_Q$ the quantum Fisher information matrix and $\tilde{\mathcal{F}}_Q$ any approximation to it. Our first estimator is presented in Theorem 3.1.

Theorem 3.1. *For every parameterization $\boldsymbol{\theta} \mapsto |\psi(\boldsymbol{\theta})\rangle$, the matrix elements of the quantum Fisher information matrix can be inferred as*

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = 2(2^n + 1) \sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H} \left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial \theta_j} \right], \tag{3.50}$$

where $\mathbb{E}_{U\sim\mu_H}[\cdot]$ is the ensemble average over random unitary U drawn from the Haar distribution μ_H and $p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) := \langle\psi(\boldsymbol{\theta})| U^\dagger \Pi_{\boldsymbol{s}} U |\psi(\boldsymbol{\theta})\rangle$ is the probability of the outcome $\boldsymbol{s}$ when measuring $U|\psi(\boldsymbol{\theta})\rangle$ with respect to the computational basis projectors $\{\Pi_{\boldsymbol{s}}\}$.

Proof. For a detailed proof, see Appendix A.3. □

Each sample requires at most $2m$ quantum state preparations in total. To sample, it suffices to compute the partial derivatives of the probability outcomes in Eq. (3.50), and those can be easily calculated using parameter-shift rules (see Preliminaries Sec. 2). The immediate result is that, in practice, this estimator requires significantly fewer quantum

states to approximate the QFIM since it can be written as a product of first-order derivatives.

In general, generating Haar random unitaries on a quantum computer is a computationally exhaustive task since most unitary operators require a number of gates that scale exponentially to the number of qubits [134]. On the other hand, k-designs are distributions that match the Haar moments up to the k-th order (see Definition 3.2). The advantage is that k-designs can be generated efficiently.

Definition 3.2. *(Unitary k-design) A probability distribution ν supported over a set of unitaries $S \subseteq U(d)$ is defined to be a unitary k-design if and only if*

$$\mathbb{E}_{V\sim\nu}[V^{\otimes k} O V^{\dagger\otimes k}] = \mathbb{E}_{U\sim\mu_H}[U^{\otimes k} O U^{\dagger\otimes k}] \tag{3.51}$$

for all $O \in \mathcal{L}((\mathbb{C}^d)^{\otimes k})$.

We next prove a corollary of Theorem 3.1, recasting Haar random unitaries with 2-designs.

Corollary 3.4. *For U drawn from a 2-design ν, the elements of the quantum Fisher information matrix satisfy*

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = 2(2^n+1)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\nu}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right] \tag{3.52}$$

where $p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) := \langle\psi(\boldsymbol{\theta})| U^\dagger \Pi_{\boldsymbol{s}} U |\psi(\boldsymbol{\theta})\rangle$ is the probability of the outcome $\boldsymbol{s}$ when measuring $U|\psi(\boldsymbol{\theta})\rangle$ with respect to the computational basis projectors $\{\Pi_{\boldsymbol{s}}\}_{\boldsymbol{s}}$.

Proof. The expectation $\mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right]$ in the statement of theorem 3.1 expands as

$$\mathrm{Tr}\left[\mathbb{E}_{U\sim\mu_H}[U^{\otimes 2}\Pi_{\boldsymbol{s}}^{\otimes 2}U^{\dagger\otimes 2}]\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\otimes\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right], \tag{3.53}$$

which, by specializing definition 3.2 to 2-designs equals

$$\mathbb{E}_{U\sim\nu}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right],$$

thereby proving the corollary. □

The 2-design estimator: Corollary 3.4 suggests natural inference procedures for estimating the QFIM by sampling unitaries that come from a k-design with $k \geq 2$ (since a k-design is also a 2-design if $k \geq 2$). We next derive a simple estimator in this motif and call it the *the 2-design estimator*. To efficiently sample the unitary, we draw from the ensemble of the n-qubit Clifford group $Cl(n)$, which forms a 3-design. Elements from the n-qubit Clifford group can be generated by a circuit with at most $\mathcal{O}(n^2/\log n)$ elementary gates [135], showcasing that our 2-design estimator can be implemented in a NISQ setting. As before, computing the full gradient vector (and hence, drawing one sample of the estimate in 3.4) only takes $O(m)$ quantum state preparations. To obtain the estimate in Corollary 3.4, we repeat this sampling procedure K times, where K is a hyper-parameter considered as a design choice. The state preparation cost $\mathcal{O}(Km)$ of the 2-design estimator significantly improves on the $\mathcal{O}(m^2)$ required by previous known algorithms for QFIM when K is small.

For most parameterizations $\boldsymbol{\theta} \mapsto |\psi(\boldsymbol{\theta})\rangle$, the measurement probability $p_{\boldsymbol{s}}^{V}(\boldsymbol{\theta})$ distribution is likely concentrated around its expectation. Therefore, a small K likely suffices for most applications. As a curiosity to determine the accuracy limits of the 2-design estimator in the worst case, it would be interesting to construct parameterizations requiring a large K, perhaps by planting pathological high dimensional singularities. The hyperparameter can also be adaptively tuned, keeping it small at the early stages for rapid descent and increasing it closer towards the end of the evolution to converge to a more accurate ground state. We leave these interesting questions open for future research. We do demonstrate empirically in small examples that, in practice, K is much smaller than m. In essence, K offers a tradeoff between rapid and accurate descent of QITE incorporating the 2-design estimator. In most cases, a small choice of K is sufficiently accurate while facilitating rapid descent.

Next, we provide a second estimator to the QFIM, which we name *the average classical Fisher information matrix estimator* and is connected to the *random natural gradient*. As such, we provide a second definition, which is the average (over an ensemble $\nu \subseteq U(2^n)$) classical Fisher information matrix.

Definition 3.3. (Average classical Fisher information matrix). Consider an ensemble of unitary operators $\nu \subseteq U(2^n)$ from which we uniformly sample. We can define the average (over the unitary ensemble ν) classical Fisher information matrix as:

$$\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U] \tag{3.54}$$

The idea is that one can get information about the underlying quantum states by measuring them on a specific basis, just as we discussed in previous sections. If this procedure is performed repeatedly, one can get an accurate picture of the geometry of the parameterized quantum states. We were able to make an important observation that is true for all parameterized quantum states that were investigated in this chapter.

Conjecture. *If the unitaries are drawn from the Haar-distribution $\nu = \mu_H$, then the average Fisher defined in Eq.* (3.54) *approximates the quantum Fisher information matrix, i.e.*

$$\mathbb{E}_{U\sim\mu_H}[\mathcal{F}_C^U(\boldsymbol{\theta})] = \frac{1}{2}\mathcal{F}_Q(\boldsymbol{\theta}) \tag{3.55}$$

for any $\boldsymbol{\theta}$.

The proof of the previous conjecture is a very challenging task. The reason is that the unitaries that appear in the Haar-integral on the left-hand side of Eq. (3.55) enter in a non-linear fashion. As such, certain results from random matrix theory cannot be directly applied in this scenario. We discuss that thoroughly in Appendix A.3. The above conjecture indicates that by choosing the appropriate unitary ensemble to sample from (in this case, the Haar distribution), one can get an accurate description of the underlying geometry in the space of parameterized quantum states. However, as we later show (see Lemma 3.3), replacing the quantum Fisher information matrix in Eq. (3.25) by the average classical Fisher information matrix for any subset $\nu \subseteq U(2^n)$ will always result in a descent direction for a sufficiently small time step.

Each one of the two estimators that we propose (in Eq. (3.50) and Eq. (3.55)) have different advantages compared to the other. As we verified, the random CFIM estimator in Eq. (3.55) outperforms the former in Eq. (3.50) in terms of speed of convergence, i.e. it converges much faster to the QFIM. This means that with fewer sampled unitaries (and measurements) we can approximate the quantum Fisher information to great accuracy. An illustrative example is given in Figure 3.3. In this figure, we visualize how the quantum Fisher information matrix can be approximated using random measurements with either of the two estimators in Eq. (3.50) and Eq. (3.55) for a random 8-qubit parameterized quantum state at a random configuration $\boldsymbol{\theta}$. For the average CFIM, the unitaries were drawn from the Haar distribution.

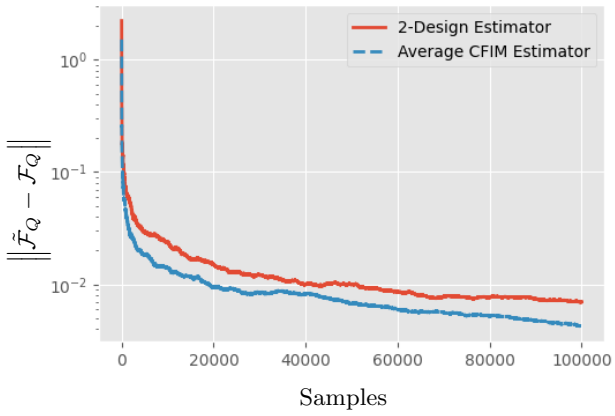

Figure 3.3: Distance of the quantum Fisher information from its corresponding estimators (in logarithmic scale). The red line corresponds to the estimator in Eq. (3.50) while the blue line to the estimator in Eq. (3.55).

Given the same number of samples, we can see that the average CFIM estimator (blue line) is able to approximate with a smaller error the QFIM, compared to the estimator in Eq. (3.50). However, both estimators are able to achieve a good approximation to QFIM with only a small number of samples. Similar performance was observed for all ansatz families used in this chapter.

On the other hand, the 2-design estimator in Eq. (3.50) comes with other benefits. Specifically, its implementation can be performed by sampling unitaries from a 2-design, which implies that the unitaries have an exponentially smaller depth than that of the unitaries that are sampled from the Haar distribution. As such, the estimator in Eq. (3.50) can even be experimentally realized in the early fault-tolerant (or NISQ) era where the number of qubits remains small, but we can execute longer circuits.

Recapitulating, one can replace the QFIM with either of the two proposed estimators (where both require measuring the state on a random basis). In that case, the QFIM $\mathcal{F}_Q$ is replaced by its estimator $\tilde{\mathcal{F}}_Q$ and the partial differential equation in Eq. (3.25) is transformed to:

$$\tilde{\mathcal{F}}_Q[\boldsymbol{\theta}(\tau)]\dot{\boldsymbol{\theta}} = -2\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_\tau(\boldsymbol{\theta}) \tag{3.56}$$

Specifically, the parameterized quantum state is rotated by a global unitary V, sampled by the appropriate ensemble $\nu \subseteq U(2^n)$ ($V \sim \nu$), and then is measured on the computational basis. Finally, the QFIM in Eq. (3.25) can be estimated by post-processing

the measurement and using any of the proposed estimators.

For example, in the case where we use the average CFIM estimator $\tilde{\mathcal{F}}_Q$, the partial differential equation in imaginary-time evolution Eq. (3.56) can be replaced by:

$$\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U(\boldsymbol{\theta}(\tau)]\dot{\boldsymbol{\theta}} = -2\boldsymbol{\nabla}_{\boldsymbol{\theta}}E_\tau(\boldsymbol{\theta}) \tag{3.57}$$

It is important to stress that in the case where only a single unitary is used in Eq. (3.55), then the solution of the partial differential equation:

$$\mathcal{F}_C^U(\boldsymbol{\theta}(\tau))\dot{\boldsymbol{\theta}} = -2\boldsymbol{\nabla}_{\boldsymbol{\theta}}E_\tau(\boldsymbol{\theta}) \tag{3.58}$$

is equivalent to the *Random Natural Gradient*, introduced in Sec. 3.2. The error in the updated $\boldsymbol{\theta}$ is characterized by the distance between the operators, i.e. the estimator of the QFIM ($\tilde{\mathcal{F}}_Q$) and the QFIM ($\mathcal{F}_Q$) and is quantified in Lemma 3.2.

Lemma 3.2. *Assuming that both the quantum Fisher information matrices and its estimator are full-rank and that* $\dot{\boldsymbol{\theta}}_Q$ *and* $\dot{\tilde{\boldsymbol{\theta}}}_Q$ *are given by Eqs.* (3.25), (3.56), *then the relative error* $\frac{\left\|\dot{\boldsymbol{\theta}}_Q - \dot{\tilde{\boldsymbol{\theta}}}_Q\right\|}{\left\|\dot{\boldsymbol{\theta}}_Q\right\|}$ *can be upper bounded as:*

$$\frac{\left\|\dot{\boldsymbol{\theta}}_Q - \dot{\tilde{\boldsymbol{\theta}}}_Q\right\|}{\left\|\dot{\boldsymbol{\theta}}_Q\right\|} \leq \frac{\lambda_{\max}(\mathcal{F}_Q)}{\lambda_{\min}(\tilde{\mathcal{F}}_Q)} - 1 \tag{3.59}$$

Proof. For a detailed proof, see Appendix A.4. □

Moreover, as stated in Lemma 3.3, the resulting update (using Eq. (3.57)) will always decrease the energy of the system provided that we choose the appropriate timestep (see Corollary 3.3).

Lemma 3.3. *Updating the parameters of a parameterized quantum circuit according to Eq.* (3.57) *will result in a descent direction.*

Proof. Consider the expectation value of the Hamiltonian H of a parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$:

$$E_\tau(\boldsymbol{\theta}) = \langle\psi[\boldsymbol{\theta}(\tau)]|\, H\, |\psi[\boldsymbol{\theta}(\tau)]\rangle$$

Its time derivative is then:

$$\begin{aligned}
\frac{d}{d\tau}E_\tau(\boldsymbol{\theta}) &= 2\,\mathrm{Re}\left(\langle\psi[\boldsymbol{\theta}(\tau)]|\,H\frac{d\,|\psi[\boldsymbol{\theta}(\tau)]\rangle}{d\tau}\right) \\
&= 2\,\mathrm{Re}\left(\langle\psi[\boldsymbol{\theta}(\tau)]|\,H\sum_{j=1}^{m}\frac{\partial\,|\psi[\boldsymbol{\theta}(\tau)]\rangle}{\partial\theta_j}\dot{\theta}_j\right) \\
&= \sum_{j=1}^{m}2\,\mathrm{Re}\left(\langle\psi[\boldsymbol{\theta}(\tau)]|\,H\frac{\partial\,|\psi[\boldsymbol{\theta}(\tau)]\rangle}{\partial\theta_j}\dot{\theta}_j\right) \\
&= (\boldsymbol{\nabla}_{\boldsymbol{\theta}}E_\tau(\boldsymbol{\theta}))^\intercal\dot{\boldsymbol{\theta}} \\
&= -(\boldsymbol{\nabla}_{\boldsymbol{\theta}}E_\tau(\boldsymbol{\theta}))^\intercal[\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U(\boldsymbol{\theta}(\tau)]]^{-1}\boldsymbol{\nabla}_{\boldsymbol{\theta}}E_\tau(\boldsymbol{\theta})
\end{aligned}$$

Since any classical Fisher information matrix is a positive semi-definite matrix, its average will also be positive semidefinite:

$$\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U(\boldsymbol{\theta}(\tau)] \succcurlyeq 0 \tag{3.60}$$

which implies that its inverse is also positive semidefinite. As a result,

$$\frac{d}{d\tau}E_\tau(\boldsymbol{\theta}) \leq 0 \tag{3.61}$$

and so we move into a *a descent direction.* □

Furthermore, we would like to notice how one could approximate quantities such as $\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U]$ in practice. In a real-world setting, the user would select an ensemble of unitaries $\{U_i\}$ from which they would uniformly sample from. Then, they would choose the number of unitaries K per iteration to calculate the average. Finally, the average can be approximated as:

$$\mathbb{E}_{U\sim\nu}[\mathcal{F}_C^U] \approx \frac{1}{K}\sum_{j=1}^{K}\mathcal{F}_C^{U_j} \tag{3.62}$$

Recapitulating, we can see that multiple random measurements are required to remain close to the trajectory that the QNG optimizer would follow. However, a single random measurement still suffices to move into a descent direction and converge much faster. As we illustrate in the Results Section (see Sec. 3.8), such an optimization technique is favourable, offering great speedups with much fewer quantum resources while at the same time performing much better than naive non-information-theoretic methods.

3.6 Stochastic-Coordinate Quantum Natural Gradient

In this section, we will outline the second optimization algorithm, which is based on a lower-rank approximation of the QFIM. During the past few years, researchers have proposed a number of ways to approximate the QFIM by reducing the quantum resources in order to make the QNG more applicable in real-world settings. As we previously mentioned, Stokes *et al* [30] proposed that instead of calculating the full QFIM, one could calculate a block-diagonal approximation of the QFIM. However, such an approximation may not be valid when off-block diagonal terms are highly correlated.

On the other hand, in [136], the authors suggested that one could apply the 2-SPSA algorithm (which is used to calculate the Hessian of a loss function) to approximate the QFIM. The authors suggested that this strategy is efficient as it requires a constant number of quantum states independent of the number of parameters. However, this approximation requires a larger number of shots as the number of parameters increases (or a smaller step size during the optimization) in order to achieve the same accuracy.

In this section, we provide a new approximation that is inspired by coordinate descent algorithms [118]. In these algorithms, the user determines a coordinate [119], or a block of coordinates [120] that will update on each iteration and keeps all other directions fixed. At this point, we need to take a step back and discuss a redundancy measure that was introduced in [34]. Consider a PQC C with m parameters and let its parameter dimension D_C. As introduced in [34], the parameter dimension D_C quantifies the number of independent parameters that the PQC can express in the space of quantum states. Let also $G_C(\boldsymbol{\theta})$ be the rank of QFIM at point $\boldsymbol{\theta}$. The authors numerically verified that for PQCs in which their parameterized gates follow a $(0, 2\pi)$ gate periodicity:

$$G_C(\boldsymbol{\theta}) \approx D_C \tag{3.63}$$

for randomly chosen $\boldsymbol{\theta}$. As such, for a given PQC, by measuring the rank of QFIM at random points, we can calculate the redundancy of the parameters

$$R = \frac{m - G_C(\boldsymbol{\theta})}{m} \tag{3.64}$$

In Figure 3.4 we illustrate the ranks of both the QFIM and CFIMs compared to the total number of parameters. We can visualize that the redundancy measure is large and

that only a fraction of the total parameters contribute to changing the quantum state in an independent way.

Consider the QFIM at a given configuration $\boldsymbol{\theta}$, with rank $m - k$ where k is the dimension of its kernel and m is the total number of parameters. In the ideal case where the eigenvectors and the eigenvalues are known, identifying the parameters that can change the quantum state can be performed using the following procedure.

Let the kernel of $\mathcal{F}_Q$ spanned by the eigenvectors $\{\boldsymbol{v_1}, \ldots, \boldsymbol{v_k}\}$ with $k < m$. If we project the parameters onto that subspace, we can identify which parameters matter most. For example, if a given parameter θ_i has the largest projection onto that subspace, then we can deduct that varying this parameter will result in minor (if not negligible) changes in the underlying quantum state. Moreover, if the projection onto the space orthogonal to the kernel is zero, then varying this parameter will have no effect on the quantum state. Mathematically, as also noted by [34], one would have to calculate the quantity:

$$g_i = \sum_{j=1}^{k} |v_j^i|^2 \tag{3.65}$$

for every $i \in [m]$ and the largest g_i would correspond to parameters whose variation will not lead to any change of the state. However, the problem is that such a procedure is inefficient in practice, as different parameters may matter most in different parameter settings, and the calculation of the full QFIM is needed. Instead, we propose the following solution to this problem, which is computationally cheaper but may not always find all the parameters that result in an independent change.

Now, consider a subset $L \subseteq [m]$ (of cardinality $|L| = l \leq m$) of the total number of parameters and let $\mathcal{F}_{RQ}$ be the *reduced* QFIM with elements defined as:

$$[\mathcal{F}_{RQ}]_{ij} = \begin{cases} [\mathcal{F}_Q]_{ij} & \text{if } i, j \in L \\ 0 & \text{otherwise} \end{cases} \tag{3.66}$$

where $[\mathcal{F}_Q]_{ij}$ are the elements of the QFIM defined in Eq. (3.8). We can immediately see that the first advantage of this approximation is that the cost of calculating the reduced QFIM immediately drops down to $\mathcal{O}_Q(l^2)$ quantum state preparations (but the same classical memory resources). The second advantage is that increasing the size of the subset L, i.e. considering more parameters, improves the accuracy of the approximation, but increases the cost, with the method essentially becoming QNG when $L = [m]$. The

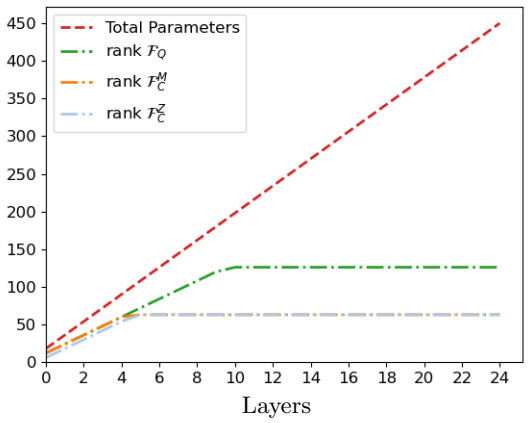

Figure 3.4: Ranks of QFIM $\mathcal{F}_Q$, CFIM with Z-basis measurements $\mathcal{F}_C^Z$, and CFIM with measurements on a random basis $\mathcal{F}_C^{\mathcal{M}}$ compared to the total number of parameters of the parameterized quantum circuit on the left of Figure 2.2.

physical intuition behind this approximation is that the reduced QFIM carries information about how the quantum state changes (with respect to a distance measure) if we vary only a portion of the total parameters. However, the question that naturally arises is how one can pick any such coordinate subset.

A straightforward way to choose the coordinate subset is to sample uniformly a subset (of user-specified cardinality) of the total number of parameters. The probability that this subset includes all independent parameters is given by Lemma 3.4.

Lemma 3.4. *Consider a parameterized quantum circuit C composed of m parameters. Consider also the unknown subset $L \subseteq [m]$ (of cardinality $|L| = l$) of parameters whose variation results in an independent change of the underlying quantum state at point $\boldsymbol{\theta}$. Let also $S_k \subseteq [m]$ be the set of (uniformly) randomly sampled parameters of cardinality $|S_k| = l + k \leq m$. Then, the probability of $L \subseteq S_k$ is*

$$\mathbb{P}[L \subseteq S_k] = \frac{(l+k)!}{k!l!} \frac{l!(m-l)!}{m!} \tag{3.67}$$

Proof. Let $S = [m]$, be the set of parameters that parameterize a quantum circuit. Let also $L \subseteq S$ with $|L| = l$ be the target subset of $l < m$ parameters that result in an independent change in the quantum state. Our goal is to quantify the probability of

sampling a subset $S_k \subseteq S$ (of cardinality $|S_k| = l + k$) so that $L \subseteq S_k$.

Consider at first the case where $k = 0$. In that case, the probability that we sample the target subset can be calculated as:

$$\Pr[L = S_0] = \left(\frac{l}{m}\right)\left(\frac{l-1}{m-1}\right)\cdots\left(\frac{1}{m-l+1}\right) = \frac{l!(m-l)!}{m!} \tag{3.68}$$

where the right-hand side corresponds to the condition probability of sampling the first parameter at random and being in the target subset $\frac{l}{m}$, then sampling the second parameter at random and being also one of the remaining $(l-1)$ parameters in the target subset $\frac{l-1}{m}$ and so on until the last parameter that we sample to be also on the target subset $\frac{1}{m-l+1}$. Then, consider $k = 1$. In that case, the probability of sampling the subset is calculated as:

$$\begin{aligned}\Pr[L \subseteq S_k] = \frac{m-l}{m}\frac{l}{m-1}\frac{l-1}{m-2}\cdots\frac{1}{m-l} + \frac{l}{m}\frac{m-l}{m-1}\frac{l-1}{m-2}\cdots\frac{1}{m-l} \\ + \ldots + \frac{l}{m}\frac{l-1}{m-1}\cdots\frac{1}{m-l+1}\frac{m-l}{m} = (l+1)\Pr[S_0 = L]\end{aligned} \tag{3.69}$$

where the first term corresponds to the conditional probability that the first parameter that we sample is not in the target subset, but the rest are. Then, the second term corresponds to the first parameter being in the target subset, the second not being in the target subset and the rest of parameters again being in the target subset and so on for the remaining terms. In the exact same manner, we can calculate that for the general case $k = l + n$, the probability is:

$$\Pr[L \subseteq S_n] = \frac{(l+n)!}{n!l!}\frac{l!(m-l)!}{m!} \tag{3.70}$$

□

One can use this approximation and construct an approximation to the QNG. At each iteration, the user samples (uniformly at random) a different subset $L_i \subset [m]$ of the total coordinates and calculates the reduced QFIM given by Eq. (3.66). Then, the parameters are updated according to the rule:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta[\mathcal{F}_{RQ}(\boldsymbol{\theta}_k)]^{+}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta}_k) \tag{3.71}$$

Algorithm 2: Stochastic-Coordinate Quantum Natural Gradient

Input : Problem Hamiltonian $\mathcal{H}$;
Ansatz family $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle$;
Total iterations K;
Loss function $\mathcal{L}(\boldsymbol{\theta})$;
Initial parameters $\boldsymbol{\theta} = \boldsymbol{\theta_0}$;
Learning rate η;
for $k = 1, 2, \ldots, K$ **do**
  Shuffle random subset of coordinates $L_k \subset [m]$;
  Calculate derivatives $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_i}\ \forall\ i \in L_k$;
  Calculate the reduced QFIM $\mathcal{F}_{RQ}(\boldsymbol{\theta})$;
  Update $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta[\mathcal{F}_{RQ}(\boldsymbol{\theta})]^{+}\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})$;
end
return $\boldsymbol{\theta}$

3.7 Method Evaluation

In the first part of this section, we discuss the mathematical problems used for our experiments. In the second part, we discuss the appropriate figures of merit used to benchmark our proposed algorithms (see Random Natural Gradient in Sec. 3.2 and Stochastic-Coordinate Quantum Natural Gradient in Sec. 3.6). The choice of quantum circuits used in our experiments are illustrated in Figure 2.2.

3.7.1 Evaluation Metrics

In order to fairly evaluate our proposed algorithms, we choose to benchmark based on two different metrics. In a hybrid quantum-classical setting, one would have to carefully evaluate both the classical and the quantum resources needed to execute an optimization algorithm.

As we compare classical optimization algorithms, a fair choice of metric is to count the number of optimization iterations (or else how many times we have to update the parameters of the quantum circuit) until convergence.

The second choice of metric is how many (different) quantum states we have to prepare until we converge. For example, as we already discussed, although the QNG may converge faster in terms of optimization iterations, it actually requires $\mathcal{O}_Q(m^2)$ quantum states at each iteration. So a careful analysis may prohibit any practical advantage, as

the resources may be larger than performing naive gradient descent.

On top of that, it is beneficial to benchmark our method to QNG and GD on a large number of instances in order to evaluate the overall performance. For this reason, we choose two additional metrics. The first (which quantifies the quality of the output solution) is the average probability of sampling the optimal solution (i.e. the overlap with the ground state) of the output state (when the optimization is terminated). The second (which quantifies the speed of methods) is the average relative error (over all different instances) per iteration. It quantifies how much, on average, at a given iteration, the current state differs from the optimal state.

3.8 Results

In this section, we will illustrate how the two proposed methods perform in different optimization settings. In the first part, we will investigate the performance of the *Random Natural Gradient* (see Sec 3.2) while in the second part, the *Stochastic-Coordinate Quantum Natural Gradient* (see Sec. 3.6).

3.8.1 Technical Details

In this chapter, all simulations were performed using Qiskit's exact *Statevector simulator*, which allows noiseless executions of the quantum circuits. For all classical optimization algorithms used, we calculated the gradients using the parameter shift rules [68] and used the same learning rate $\eta \in [0.001, 0.1]$ for all algorithms. For the classical combinatorial optimization problems, we used the quantum circuit on the right of Figure 2.2 with nearest-neighbours interactions and 4 layers, while for the Transverse-Field Ising problem, the quantum circuit on the left of Figure 2.2 with all-to-all connectivity and 3 layers. Moreover, for the random measurements, we used the same type of circuits with different Pauli rotations and with smaller depths than the circuits that generated $|\psi(\boldsymbol{\theta})\rangle$. Finally, in order to calculate the Moore-Penrose inverses for the update steps, we set a threshold of 10^{-4} as a cutoff for the singular values in order to prohibit very large steps.

3.8.2 Random Natural Gradient

3.8.2.1 MaxCut

We can visualize the overall performance of RNG compared to GD and QNG in Table 3.1 and Figure 3.5. The three methods were compared on random weighted 3-regular graphs. This class of graphs have only two optimal solutions, where each one can be acquired from the other by flipping all qubits (due to the $\mathbb{Z}_2$ symmetry of the MaxCut problem).

In Table 3.1, we can illustrate the probability of sampling the optimal solution when the three different optimization methods were used with the same initial angles and the same step size ($\eta = 0.05$). We can see that the Random Natural Gradient and Quantum Natural Gradient perform almost similarly. The Gradient Descent method, however, always performs worse than the former information-theoretic methods. Overall, we expect RNG and QNG to perform similarly and, in the limit of infinitely many iterations, to converge to the same point (provided that the approximation of RNG to QNG is sufficient).

On the other hand, in Figure 3, we plot the average relative error of the output solution for the three methods (over the 30 different instances on 12 qubits). As can be clearly seen, the QNG and RNG return solutions that, on average, are always closer to the optimal solution. This highlights the fact (in agreement with [31]) that information-theoretic methods perform significantly better than methods that do not consider the information of the underlying state space. For additional experiments that illustrate the actual resources needed for RNG compared to QNG, we point the reader to Appendix A.2.

3.8.2.2 Number Partitioning

As in MaxCut, we illustrate the overall performance of RNG compared to GD and QNG in Table 3.2. For our experiments, we chose to sample integers from the $[0, 25]$ set and set the step-size of $\eta = 0.001$ (a smaller step-size is needed to guarantee convergence as the cost values are much larger than MaxCut). Similar to MaxCut, Number Partitioning also has a $\mathbb{Z}_2$ symmetry. As it can be clearly visualized, the superiority of information methods is also present in the Number Partitioning problem. Clearly, the two information-theoretic optimization methods are able to return high-quality outputs, achieving significantly larger overlap with optimal solutions (on an average of 60 instances).

MaxCut	Optimal Solution Overlap (%)	
	12 Qubits	14 Qubits
Gradient Descent	37.3	24.99
Random Natural Gradient	41.89	32.02
Quantum Natural Gradient	42.4	29.99

Table 3.1: Probability of sampling the optimal solution for the MaxCut problem for 60 different instances (30 for 12 qubits and 30 for 14 qubits). All instances correspond to random 3-regular weighted graphs. Both QNG and RNG outperform GD.

Number Partitioning	Optimal Solution Overlap (%)	
	12 Qubits	14 Qubits
Gradient Descent	20.83	17.62
Random Natural Gradient	27.54	25.54
Quantum Natural Gradient	28.17	26.31

Table 3.2: Probability of sampling the optimal solution for the Number Partitioning problem for 60 different instances (30 for 12 qubits and 30 for 14 qubits). All instances correspond to a set of integers drawn from the $[1, 25]$ interval. Both information-theoretic methods outperform GD.

3.8.2.3 Heisenberg Model

You can visualize the performance of the Random Natural Gradient on a Heisenberg-model instance of 10 qubits (with couplings $J = h = 1$) in Figure 3.6. For this instance, we used the hardware-efficient ansatz seen on the left side of Figure 2.2, with a nearest-neighbour connection.

On the left side of Figure 3.6, we illustrate the number of optimization iterations (or else how many times we update the parameters) until convergence (we stop the optimization after 500 iterations). We see that the RNG is able to reach the region of the local minimum much faster (in terms of optimization iterations) compared to the QNG. On the other hand, the GD optimizer, as it doesn't carry any information about the underlying Riemannian space, gets stuck in a local minimum, performing significantly worse than QNG and RNG.

However, the biggest advantage is illustrated on the right-hand side of Figure 3.6. There, we can visualize the actual (quantum) resources needed until convergence. It is clear that the RNG offers a significant advantage in the number of quantum calls, reducing the overall overhead in current quantum devices (requiring almost ten times less quantum state preparations than the QNG until convergence).

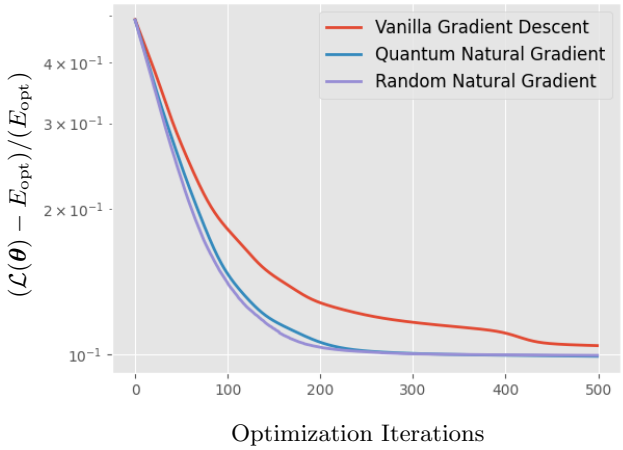

Figure 3.5: Relative error for Gradient Descent, Random Natural Gradient and Quantum Natural Gradient for 30 12-qubit random weighted 3-regular graphs (in logarithmic scale).

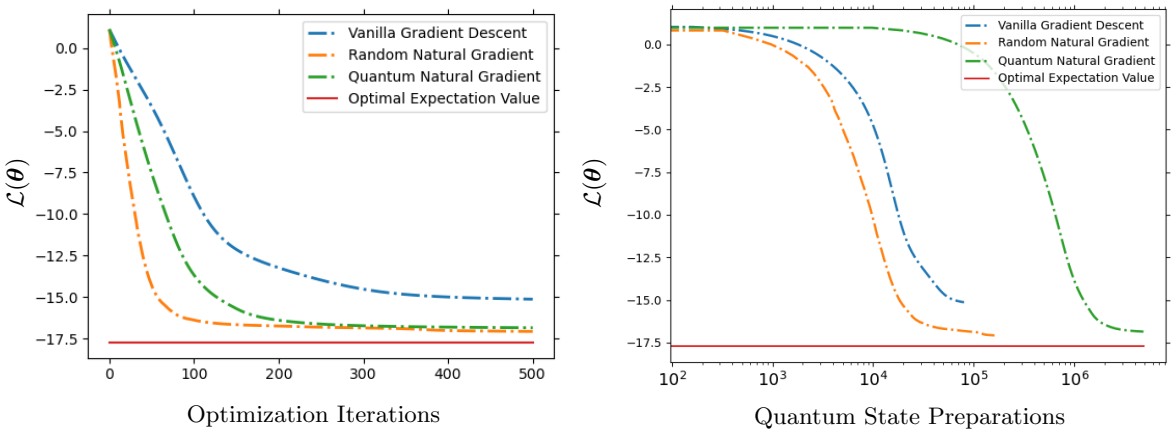

Figure 3.6: Performance of the Random Natural Gradient optimizer on a Heisenberg model of 10 qubits compared to the Quantum Natural Gradient and Gradient Descent on both the optimization iterations (left figure) and on quantum resources (right figure) (in logarithmic scale). The RNG and GD methods require fewer quantum resources to converge (compared to QNG), but, as seen in both figures, the GD method converges to a bad quality minimum.

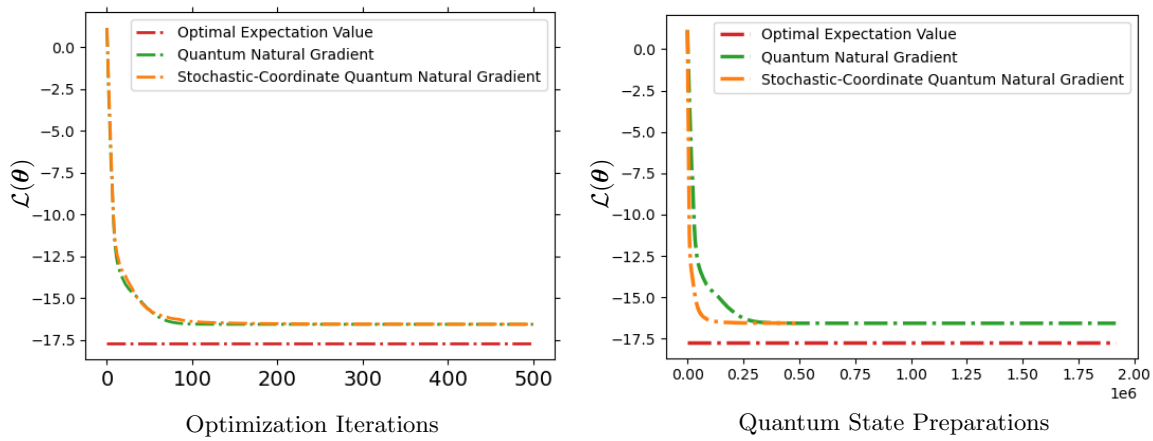

Figure 3.7: Comparison of SC-QNG (with sampling half of the total parameters at each iteration) with QNG both in terms of optimization iterations (left) and quantum calls (right).

3.8.3 Stochastic-Coordinate Quantum Natural Gradient

In figure 3.7, we can visualize the performance of the Stochastic-Coordinate Quantum Natural Gradient on a Heisenberg-model instance of 10 qubits (again with couplings $J = h = 1$ but with different random initial angles).

An important hyperparameter in the SC-QNG optimizer is the cardinality of the random subset S_k that we uniformly sample at each iteration. In this work, we make the naive choice that the user samples $m/2$ parameters at random in each iteration. As we can see in Figure 3.7, both QNG and SC-QNG require the same number of optimization iterations to converge. However, the latter results in a significant reduction in the actual quantum resources needed, requiring only a fraction of the quantum states that we need to prepare for QNG.

Chapter 4

Evolving objective function for improved variational quantum optimization

4.1 Introduction

As we already discussed in the previous sections, the goal of VQAs is to find the optimal parameters that generate quantum states that are close to the ground state of interest. For classical combinatorial optimization problems (as introduced in Sec. 2.3), the optimal solution is one (or many) computational basis state. Preparing a quantum state that has a big overlap with that state is sufficient to give a good and quick approximation of the ground state.

For example, if the user can achieve a constant but likely small overlap with the correct solution, it is guaranteed that after sampling this state a sufficient number of times to obtain at least one sample of the true ground state. In [61], the authors used this idea, and instead of evaluating the proximity of a quantum state to the desired (ground state) by minimizing the (overall) energy, they aimed to minimize the energy of the lowest tail of a quantum state. This, intuitively, would succeed quicker in finding a quantum state that has a non-negligible overlap with the solution (but not necessarily very high overlap). This state, however, suffices to solve the problem. This intuition was also confirmed with numerical simulations. In other words, the objective (loss) function used in the classical optimizer in order to find the optimal parameters was not the energy

of the quantum state but the tail of the corresponding distribution.

In this chapter, we consider an evolving objective function. In our proposal, the way a quantum state's loss is computed dynamically changes during the classical optimization process. Our starting point is the objective function in [61] focusing on a small tail, but during the optimization process, we gradually increase the tail (or else the fraction of the distribution we "count") until we reach a point that all the distribution is included, i.e. we measure the full expectation value of the energy.

Our contributions can be summarized as follows:

- We introduce an evolving objective function that starts with the CVaR defined in [61] and gradually, during the optimization process, becomes the full energy of the quantum state. Alternative forms of this Ascending-CVaR objective functions are considered, and linear and sigmoid functions (that appear to perform better) are selected.
- We test our proposal with classical numerical simulations (using up to 20 qubits), both in the setting of VQE with hardware-efficient ansatz and in QAOA. Our results suggest that our proposal leads to faster convergence with a bigger overlap with the ideal solution than prior works, while crucially, it succeeds in obtaining the solution in (many) instances in which other techniques fail altogether.
- Our analysis is done for three different combinatorial optimization problems, namely MaxCut, Number Partitioning and Portfolio Optimization (see details of the problems in Sec. 2.3). We consider many different instances and problem sizes where the conclusions persist in all cases. This has importance in its own right, since these problems are important by themselves, and our proposal gives an approach to improve the performance and bring closer a "useful" quantum advantage. Interestingly, our method offered greater advantage in "hard instances" of the problems, where the other methods frequently failed to find the solutions altogether.

4.1.1 Conditional Value-at-Risk (CVaR)

At first, it is essential to understand the CVaR objective function introduced by Barkoutsos et al. in [61]. They demonstrated that their proposal performed better than minimizing the expectation value. The key observation is that for optimization problems, the optimal

solution is a computational basis state. For a general quantum state $|\psi(\boldsymbol{\theta})\rangle$ one can prepare and measure it (multiple times) on the computational basis, and the expectation value of the energy is simply the average of the individual computational basis state energies. To find the overlap of this state with the optimal solution (ground state), one can simply observe the frequency of the computational basis state with the smallest energy. Naturally, if that overlap is too small (or even zero), it is possible that none of the measurement outcomes will give the solution. On the other hand, it is also clear that the overlap of this state with computational basis vectors with high energy is irrelevant when finding the ground state. The idea of [61] was to use this observation, and instead of using all the measurement outcomes and computing the expectation value, they used as objective function the lower tail of the distribution of energies obtained, i.e. ignored all but a small fraction (with smallest energy) of their measurement outcomes.

They then demonstrated that their technique succeeded in getting a quantum state that has a sufficiently large overlap with the ground state quicker. This, in turn, is sufficient to actually find this ground state since, as a final step, once the optimal $\boldsymbol{\theta}^*$ is found, one can keep the computational vector that has the smallest energy only. Specifically, let H_k be the energy corresponding to a computational basis vector, and let us order them in such a way that larger k corresponds to larger energy. For each state, one repeats the measurement K-times, so there are (up to) K distinct values H_k. In [61], a new parameter α was introduced. Let $\alpha \in (0, 1]$ be the fraction (part of the tail) that we want to keep. This fraction, typically, needs to be non-negligible (we can assume, for simplicity, that it is constant). Then, the objective function that was used was the average of the smallest αK samples, i.e.

$$CVaR_\alpha = \frac{1}{\lceil \alpha K \rceil} \sum_{k=0}^{\lceil \alpha K \rceil} H_k \tag{4.1}$$

In order to achieve the same accuracy when evaluating this objective function as the accuracy achieved when computing the expectation value using K shots, it is clear that the number of runs of the preparation circuit needs to be increased to K/α.

As it was proven by [61], the angles $\boldsymbol{\theta}^*$ that minimize CVaR_α do not (in general) correspond to minima of the expectation value. As a result, the angles that lead to the smallest possible α-tail differ from the angles that minimize the average of the samples. This fact motivates us to introduce a lower α-tail optimization so as to achieve an overlap

with the optimal state of at least α, i.e. find optimal $\boldsymbol{\theta}^*$ that satisfies:

$$|\langle\psi(\boldsymbol{\theta}^*)|\psi_{\text{opt}}\rangle|^2 \geq \alpha \tag{4.2}$$

4.2 Ascending-CVaR

The CVaR cost function of [61] was shown to perform better in general than the "standard" expectation value. There are three observations, however, that motivate our proposal. First, as noted in [61], the choice of α is arbitrary, and importantly, for different problems and even for different instances of the same class of problems, the optimal choice of α varies in a non-obvious (e.g. monotonic) way. The performance of the algorithm's speed, as well as whether it finds the solution at all, depends on that choice. The second point is that optimizing with a fixed small α has further disadvantages: (i) it "finds" parameters $\boldsymbol{\theta}$ that result in a state that does not have the greatest overlap with the solution and (ii) the true running time of the algorithm to achieve same accuracy is larger, in other words for each iteration one requires $1/\alpha$ times more measurements to achieve the same accuracy in estimating the cost function (since only the lower α fraction of the measurements are used). Finally, the third observation is that the $CVaR_\alpha$ objective functions with different α have a different energy landscape. For any fixed choice of α, the optimizer could "get stuck" at a local minimum. Interestingly, if one varies α during the optimization, while they still ensure that if the algorithm finds the true ground state, it remains there, we also avoid getting stuck at local minima since those are different for different choices of α. Therefore, if the optimizer reaches a point that has a local minimum for one value of α, when α changes, this point (may) no longer be a local minimum and thus could continue "moving" towards the true global minimum (ground state).

Let's say that an optimization problem has an optimal solution, which is a computational basis state, and we denote it as $|\psi_{\text{opt}}\rangle$. Let's also assume that a parameterized family of gates, $U(\boldsymbol{\theta})$, acts on the $|0\rangle^{\otimes n}$ state and produces the state

$$|\psi(\boldsymbol{\theta})\rangle = a_{\text{opt}}(\boldsymbol{\theta})\,|\psi_{\text{opt}}\rangle + a_{\text{other}}(\boldsymbol{\theta})\,|\psi_{\text{other}}\rangle \tag{4.3}$$

where $|\psi_{\text{other}}\rangle$ is the superposition of all sub-optimal computational basis states. Let's also assume that this parameterized family of states can achieve *a maximum overlap* κ

with the optimal solution[1]. We can write the state $|\psi\rangle$, corresponding to the state with the highest overlap, without loss of generality as:

$$|\psi\rangle = \sqrt{\kappa}\,|\psi_{opt}\rangle + \left(\sqrt{1-\kappa}\right)|\psi_{other}\rangle \tag{4.4}$$

Proposition 4.1. *For the family of states in Eq.* (4.3) *and for all* $\alpha \leq \kappa$*:*

$$\min_{\boldsymbol{\theta}} CVaR_{\alpha}(\boldsymbol{\theta}) = \min_{|\phi\rangle} \langle\phi|\, H\, |\phi\rangle \tag{4.5}$$

i.e. all $CVaR_{\alpha}$ *with* $\alpha \leq \kappa$ *share the same minimum objective function value, which is the smallest eigenvalue of the Hamiltonian* H*.*

It is clear from Proposition 4.1 that *all* $CVaR_{\alpha}(\boldsymbol{\theta})$ *with* $\alpha \leq \kappa$ *share the same ground state*, which is the true optimum of the optimization problem. Thus all angles $\boldsymbol{\theta}^*$ that correspond to a global minimum of $CVaR_{\alpha_1}$ will also correspond to a global minimum of $CVaR_{\alpha_2}$ if $\alpha_2 \leq \alpha_1 \leq \kappa$. For example, for an ansatz family $U(\boldsymbol{\theta})$ that is able to attain 10% overlap with the optimal computational basis state, if one is able to find the global minimum of $CVaR_{0.1}$, which means that 10% of the measurements correspond to the ground state, then it is clear that all $CVaR_{\alpha}$ with $\alpha < 0.1$ will also be minimized by the same angles.

Proposition 4.2. *Let an optimization problem with an optimal solution* $|\psi_{opt}\rangle$ *corresponding to a computational basis state. For any parameterized family of gates* $U(\boldsymbol{\theta})$ *that can achieve a maximum overlap* κ *with the optimal solution, the angles* $\boldsymbol{\theta}^*$ *that correspond to the global minimum of* $CVaR_{\alpha_2}$ *will also correspond to a global minimum for* $CVaR_{\alpha_1}$ *if* $\alpha_1 \leq \alpha_2 \leq \kappa$*. The converse does not necessarily hold.*

In other words, Proposition 4.2 states that if $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} CVaR_{\alpha_2}(\boldsymbol{\theta})$ then also $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} CVaR_{\alpha_1}(\boldsymbol{\theta})$ for all $\alpha_1 \leq \alpha_2 \leq \kappa$. This indicates why decreasing α may not seem like a good choice. If, for example, the optimizer is able to find the optimal angles that minimize $CVaR_{\alpha_1}$ with $\alpha_1 \leq \kappa$, then for all $\alpha_2 < \alpha_1$ they will still remain optimal angles and thus will not be able to achieve a higher overlap state.

Proposition 4.3. *A local minimum for* $CVaR_{\alpha_1}$ *does not necessarily correspond to a local minimum for* $CVaR_{\alpha_2}$ *if* $\alpha_1 \neq \alpha_2$*.*

[1] In other words, the complex coefficient $a_{opt}(\boldsymbol{\theta})$ corresponding to the probability of sampling the optimal solution $Pr(\text{opt}) = |a_{\text{opt}}(\boldsymbol{\theta})|^2$ has a maximum value: $\max_{\boldsymbol{\theta}} |a_{\text{opt}}(\boldsymbol{\theta})|^2 = \kappa$)

Proposition 4.3 was proven using a counterexample in [61]. All these Propositions are important for introducing a non-stationary optimization technique that avoids local minima. We know from Proposition 4.1 that all $CVaR_\alpha$ objective functions with $\alpha \in (0, \kappa]$ share the same minimum objective value, which is the ground state energy of the Hamiltonian. We also know from Proposition 4.2 that many of the global minima for α_1 may not be a global minimum for α_2 if $\alpha_1 < \alpha_2$ and thus increasing α introduces extra information about the optimality of states. Finally, Proposition 4.3 indicates that different objective functions are associated with different energy landscapes as they do not agree on the local minima.

However, knowing the maximum overlap κ in advance is not always possible. In the case of VQE with hardware-efficient ansatz families, it can be shown that $\kappa = 1$ and so $\min_{\boldsymbol{\theta}} CVaR_\alpha(\boldsymbol{\theta}) = \min_{|\phi\rangle} \langle\phi| H_C |\phi\rangle$ for every $\alpha \in (0, 1]$. On the other hand, for the QAOA ansatz, our experiments showed that κ is usually small for small-depth circuits but increases with the number of layers as the ansatz family becomes more expressive.

The loss functions used in variational quantum algorithms, to our knowledge, are "constant in time", meaning that the whole optimization is run with a fixed loss function. To solve the issue of "selecting the best α" and the other reasons listed above, we propose to use a dynamically evolving cost function that essentially passes through a fixed set of α values. In the case of VQE, it is initialized in a very small value and the optimization ends with $\alpha = 1$ that is the standard expectation value of the Hamiltonian. We call all these cost functions *Ascending-CVaR*. This also has a great(er) number of free choices since we can now freely choose the (ascending) function. However, all choices we tried for the ascending function performed (in general) better than fixed α, which indicates that the evolving cost function is a promising approach. For the remainder of the chapter, we focused on two functions that performed better:

The *linear ascending* in which the parameter α_t is iteratively and discretely increased by the rule:

$$\begin{aligned} \alpha_{t+1} &= \alpha_t + \lambda \\ CVaR_{\alpha_t} &= \frac{1}{\lceil \alpha_t K \rceil} \sum_{k=0}^{\lceil \alpha_t K \rceil} H_k \end{aligned} \tag{4.6}$$

where $\lambda \in [0.025, 0.045]$ is the *ascending factor* and $0 < \alpha_t \leq 1$.

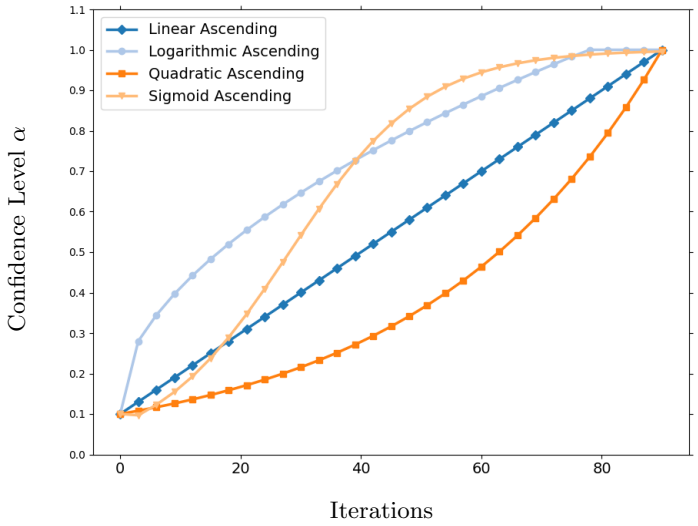

Figure 4.1: Different choices for the ascending function. All functions start from the same initial point, $\alpha_0 = 0.01$ and ascend until $\alpha_f = 1$ is reached.

The *sigmoid ascending* in which the parameter α_t is discretely increased according to the function:

$$\alpha_t = \frac{1}{1 + e^{5-\lambda t}} \tag{4.7}$$

where $\lambda \in [0.3, 0.4]$ is again the ascending factor and $0 < \alpha_t \leq 1$.

To reach this conclusion, we tested four different functions, a *sigmoid*, a *linear*, an *exponential*, and a *logarithmic* (see Figure 4.1) on VQE-CVaR$_{\alpha_t}$ with various different ascending rates. All functions were tested on all three problems. The metrics used were the magnitude of the overlap with the optimal solution, the success rate (i.e. the number of times where it succeeds in achieving a non-negligible overlap) as well as the average time taken to achieve at least 10% overlap (for details see Sec. 4.4).

The linear ascending, Eq. (4.6), and the sigmoid ascending, Eq. (4.7), functions have the most steady behaviour as it can be seen in Figure 4.2, outperforming the other two types on the majority of instances. The sigmoid was slightly slower in terms of speed, which is why we mainly used the linear one. However, as we will discuss in the next section, it appears that it may be better in some classes of problems, especially on harder instances with ~ 50 qubits. It seems that in those cases, the optimizer is doing better,

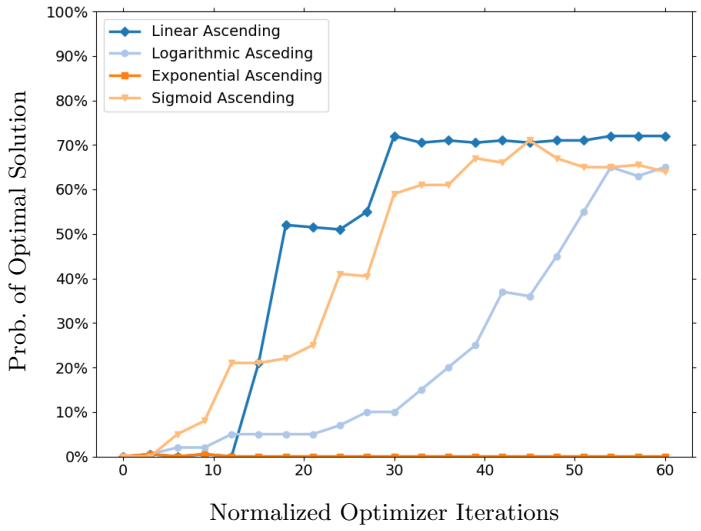

Figure 4.2: Portfolio optimization instance for 18 assets and different ascending functions. The blue line (down-pointing marker) indicates the linear ascending and always achieves a high overlap with the optimal solution in contrast to the orange line (line marker), the exponential ascending, which fails in almost any instance.

spending the majority of its iterations on low α values, and thus, the sigmoid performs better.

It is worth noting that increasing α to $\alpha = 1$, where it becomes the expectation value, is not necessary. The whole point of variational algorithms is to achieve a constant, non-negligible overlap with the optimal or near-optimal solution. For that reason, one could only vary α until it reaches a certain value, truncating the optimization and reducing the number of iterations by a considerable amount.

We remark here that *Ascending-CVaR* is fundamentally different from an adaptive strategy that selects the optimal value for the parameter α. Specifically, by looking at the results in Sec. 4.5, we can see that our method is able to reach quantum states that result in a very high overlap with the optimal state (almost equal to unity) that no constant choice of α would be able to attain. Even when the question is whether we find the solution with at least some small probability, Ascending-CVaR succeeds in cases where all of the fixed α failed.

In order to get the intuition of why our method works, one should think of what varying α actually does. The optimizer, at each iteration, moves towards a local (or a

global) minimum corresponding to the instantaneous value of α. By increasing α at every step, the optimization is able to "see" a larger part of the energy distribution of the quantum state. This translates to the optimizer gaining additional information about the quantum state, which modifies the objective function landscape.

This extra information alters the landscape and is thus able to "erase" false local minima; while using the previous step as "initialization" it is unlikely that the optimizer will get stuck in new sub-optimal minima. Moreover, since only the global minima are invariant under α transformations, this change in the landscape will not affect any correct moves of the optimizer. In other words, if the optimization algorithm did converge in a sub-optimal local minimum for a value of α, it may not still be in a local minimum by switching into a different value of α. Hence, the *Ascending-CVaR* objective function guides the trajectory of the optimizer in the highly parameterized space until it reaches a (nearly) optimal solution.

Finally, the reason our method is sensitive to the choice of the ascending factor λ is related to the speed at which the optimizer is receiving this extra information (see Sec. 4.6.2 on how we numerically test the ascending factors for the different optimization problems).

The pseudocode for the Ascending-CVaR algorithm is outlined in Algorithm 3.

Algorithm 3: Ascending-CVaR Optimization Algorithm

Require: Cost Function $C(\boldsymbol{\theta})$;
$\boldsymbol{\theta}^{(0)} \leftarrow$ Random initial parameters in the domain of $C(\boldsymbol{\theta})$;
$\alpha_0 \leftarrow$ Initial α;
$g(\alpha) \leftarrow$ Ascending function;
$U(\boldsymbol{\theta}) \leftarrow$ Ansatz Family
for $i = 1, 2, \ldots$ **do**
  $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} CVaR_{\alpha_{i-1}}(\boldsymbol{\theta})$ with initial parameters $\boldsymbol{\theta}^{(0)}$;
  if *stopping condition is met* **then**
    return $\boldsymbol{\theta}^*$;
  end
  $\alpha_i \leftarrow g(\alpha_{i-1})$;
  $\boldsymbol{\theta^{(0)}} = \boldsymbol{\theta}^*$;
end

4.3 Why our method works: An example

It would be illustrative to describe how local minima may vanish when the objective function is changed during the optimization. In Figure 4.3, we plot the $CVaR_\alpha$ objective function landscape for different values of α. We choose to draw the landscape for the QAOA algorithm with depth $p = 1$ because the two parameters β, γ make it suitable to visualize in a 2D plot. On the contrary, VQE with a hardware-efficient ansatz, even on depth $p = 1$, would require $2n$ parameters.

It can be easily seen that the positions of the local minima change, but the position of the true global minimum remains the same while the condition $\alpha \leq \kappa$ holds (see Proposition 4.1). However, in order to make it clearer for the reader, we choose to circle the position of a local minimum, located at $\gamma = 0.15, \beta = 1.75$. In this case, we can see how the local minima vanishes during the variation of the objective function. An optimization algorithm that could stuck during the optimization on a fixed value of α could "unstuck" with the change of α.

The problem corresponding to the figures is a small instance of the Number Partitioning problem with size $n = 8$. Even in a small-size instance like this, the landscape is full of sub-optimal local minima where the optimizer can falsely converge. This case problem, however, does not constitute an example to prove the value of our method; it is only used to visualize the changes in the energy landscape. The biggest improvements were observed in high-dimensional expressive ansatz families like VQE with hardware-efficient parameterized gates or larger depth QAOA, which cannot be plotted in a two-dimensional contour.

4.4 Evaluation Metrics

As we discussed in Sec. 2.2.2, a common metric used for combinatorial optimization problems is the approximation ratio as given in Eq. (2.10). However, as noted earlier, the true aim of variational quantum algorithms for combinatorial optimization is to obtain a sufficiently high (but not necessarily close to unity) overlap with the optimal solution. The CVaR method, for example, is constructed in a way that the maximum overlap achieved is not unity but determined by the risk (confidence level) α. While our approach does achieve a high approximation ratio, to make a fair and more complete comparison with prior works and, importantly, with [61], we use different metrics. Specifically, to

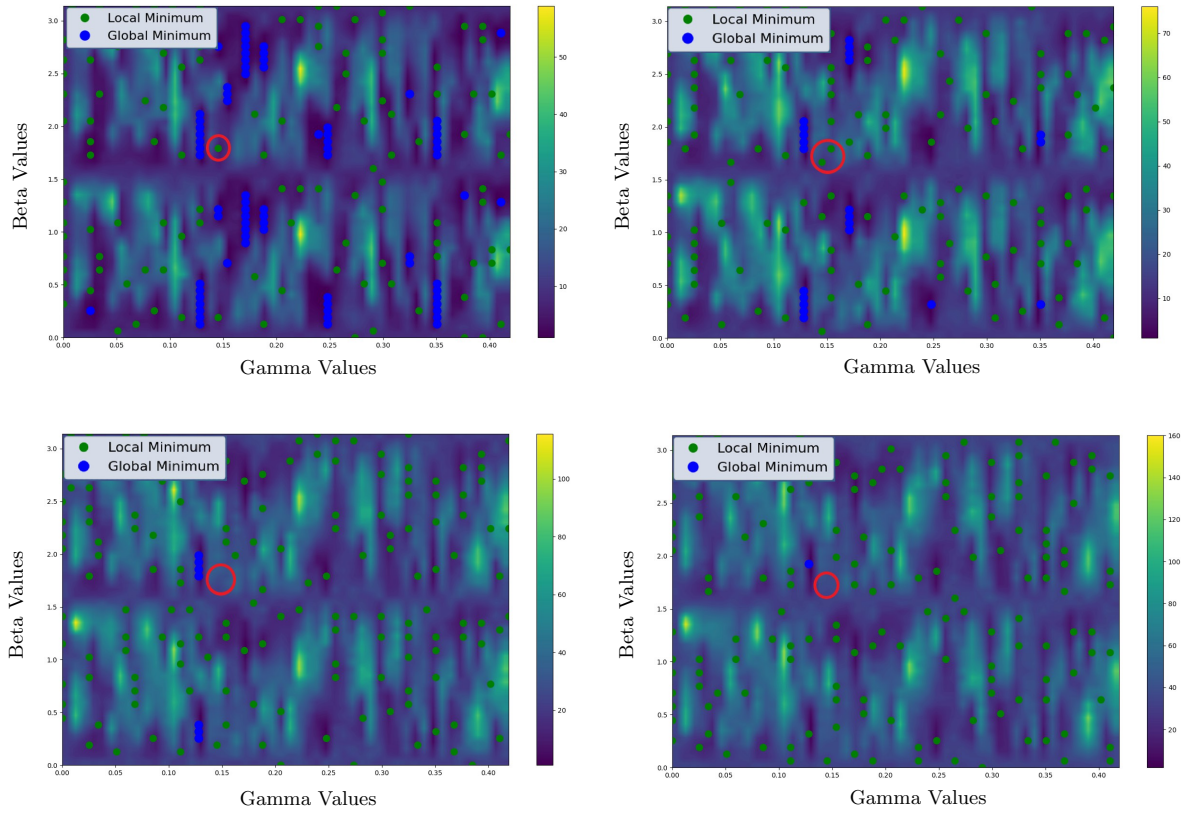

Figure 4.3: Visualization of local and global minima for different $CVaR_\alpha$ objective functions. You can see the local minima drawn in the red circle on the top two figures, corresponding to $\alpha = 0.05$ on the left and $\alpha = 0.08$ on the right. However, on the bottom figures, corresponding to $\alpha = 0.11$ on the left and $\alpha = 0.14$ on the right, the local minima no longer exist.

benchmark and test our proposed method, we used three different types of metrics. The first is the overlap with the optimal solution. If $|\psi_{\text{opt},i}\rangle$ is a d-degenerate ground state of the problem Hamiltonian, then the overlap is defined as:

$$\sum_{i=1}^{d} |\langle\psi(\boldsymbol{\theta})|\psi_{\text{opt},i}\rangle|^2 \tag{4.8}$$

i.e. the probability of obtaining the optimal solution, given the parameters $\boldsymbol{\theta}$. It follows that the parameterized state with the highest overlap with the optimal solution leads to sampling that optimal solution with the least number of circuit executions.

The second metric we want to test is the time taken to reach a given fixed overlap. We set a threshold of 10% probability of obtaining the optimal solution, and we tested

which method achieves at least that probability faster. We note, however, that in order to test which method converges to a 10% overlap faster, we have to use $\alpha \geq 0.1$ because all $\alpha < 0.1$ are not guaranteed to converge in an overlap of 10% since the parameters $\boldsymbol{\theta}$ than minimize α lead in an overlap smaller than 0.1.

To summarize the results and compare the different approaches better, for each loss function, we divided the problem instances into those that the loss function is successful and those that it fails. The meaning of what constitutes a "successful" run or a "failed" run cannot be unambiguously defined. For our work, we consider that an optimizer is successful at a given instance of a problem if it achieves at least 10% overlap with the optimal solution. It is clear that as the size of the problem instances increases, achieving a fixed 10% overlap becomes harder[2]. In our analysis, we chose 10% since this leads to interesting behaviour where the methods analysed differ in their performance.

In our experiment, for comparing with fixed α we used four different choices: $\alpha = 0.1, 0.2, 0.5, 1$. The $\alpha = 1$ choice corresponds to a non-CVaR objective function. Specifically, $\alpha = 1$ refers to the expectation value (it includes all the measurement outcomes), and it is the objective function that has been used in the overwhelming majority of the existing literature on variational quantum algorithms. In [61], they made an extensive comparison of CVaR with the expectation value (on the same combinatorial problems we make our analysis). For that reason, we choose to make the comparison of all different choices of α with our proposed α_t and plot our results in one section (see Sec. 4.5).

We also note that ascending factors $\lambda \in [0.025, 0.045]$ and $\lambda \in [0.3, 0.4]$ were found to be a good choice for the three different problems on instances with 15 to 20 qubits for the linear and sigmoid ascending respectively. (see Sec. 4.6.2). However, we would like to stress that if the sizes of the instances increase or even if the problems change, but the sizes remain the same, one would have to readjust the hyperparameter λ. Investigating theoretically the choice of both the ascending factor and the ascending function given the characteristics of the problem, as well as possible connections of our method to adiabatic quantum computing, are left for future work.

In the QAOA algorithm, we tested instances using depth $p = 1$ to $p = 6$, while on VQE, we used the circuit seen on the left of Figure 2.2 with all-to-all connection and

[2]We should note that even a much smaller overlap is sufficient to find at least once the solution, provided that the number of "shots" is sufficiently large.

worked only on the depth $p = 1$ since this depth was sufficient to get very good accuracy. In near-term devices for the QAOA algorithm, increasing the depth even more becomes impractical due to noise and decoherence. For this reason, we did not consider greater depth despite the fact that, theoretically, this could lead to better performance. This means that the variational ansatz for QAOA has only 2 to 12 parameters, i.e., only a fraction of the total parameters present in the hardware-efficient ansatz used for VQE in depth-1.

To account for the different sizes of problem instances and to make a fair comparison for the speed of convergence, we used the normalized optimizer iterations [137]. Note that this choice is made in order to be able to compare the performance of the algorithm among instances that involve different number of qubits, and see how the improvement offered by Ascending-CVaR is independent of the instance size. Concretely, the normalized optimizer iterations are defined as the number of times the optimizer evaluates the objective function divided by the function's number of parameters, i.e. the number of parameters of the ansatz. In the case of the VQE, the number of parameters is $n(1 + p)$, while on QAOA, it is $2p$. We note, however, that the real time of convergence could be used as seen in Sec. 4.6.1, where we compare the performance with respect to the total number of circuit repetitions. However, as we show below, there are instances where the constant CVaR does not achieve even a small overlap with the optimal solution, and in those cases, the time taken becomes irrelevant.

We ran our experiments on IBM's *Qiskit Aer* simulator, allowing noiseless multi-shot executions of our circuit. We set the number of executions of our circuit to $K = 1000$, which scaled up as K/α with the choice of α. All instances were given a maximum of $(66 \times parameters)$ optimizer iterations, which is more than enough iterations for an optimizer to converge to a minimum in the problems we implemented. They were initialized with a random choice of parameters, but the same for all different choices of α. We used the same gradient-free optimizer, COBYLA [138], for all different problems and instances as it was shown to outperform other classical optimizers [75].

4.5 Results

We will analyze the results for each of the three combinatorial optimization problems separately. For each of them, we will first present the results for VQE with hardware-efficient ansatz and then the results for QAOA. We note that for all three combinatorial

MaxCut	Successful Instances					Average Overlap				
	α_t	0.1	0.2	0.5	1	α_t	0.1	0.2	0.5	1
Random Graphs	96	84	81	68	53	64.69	12.13	21.45	39.28	36.24

Table 4.1: Results table for the *MaxCut* problem (VQE) for 100 random non-regular unweighted graph instances with 15 to 19 vertices.

optimization problems and for all methods used (Ascending-CVaR, "constant" CVaR [61] or the expectation value), VQE performs (much) better than QAOA, at least for the sufficiently shallow circuits that we consider. Our method improves the performance in both cases (VQE, QAOA), but since VQE gives much better results for these problems, in the comparison and discussion, we will focus on VQE instances only.

4.5.1 MaxCut

For the *MaxCut* problem, we worked on unweighted graphs with 15-19 vertices drawn from different graph classes and sampled them using the NetworkX library [139].

4.5.1.1 CVaR$_{\alpha_t}$-VQE

For regular graphs, CVaR$_{\alpha_t}$-VQE behaved equally well with constant-α's optimization and the expectation value. All of the methods reached the chosen threshold of 10% overlap with the optimal state at almost equal times without any difficulty. For that reason, we focused on harder, non-regular instances where our method outperformed the latter methods. In Table 4.1, we summarize the results for 100 random non-regular unweighted graph instances with 15 to 19 vertices. We can see that our method succeeds in more instances while the overlap achieved is also much higher.

There are many reasons why non-regular random graphs are "harder" than regular graphs. The first is that the ground state of a regular graph, due to its symmetry, is highly degenerate, where the optimizer could easily reach without converging in a sub-optimal minimum. The second is that the Hamiltonian corresponding to a random graph has more distinct eigenvalues, and as it was shown numerically by [75], the number of distinct eigenvalues correlates inversely with the performance of hardware-efficient ansatz.

Indicatively, in Figure 4.4, we plot the probability of sampling the optimal solution over the normalized number of iterations for two random graphs with 17 vertices. For the left figure, we can see how the optimizer for the Ascending-CVaR optimization can

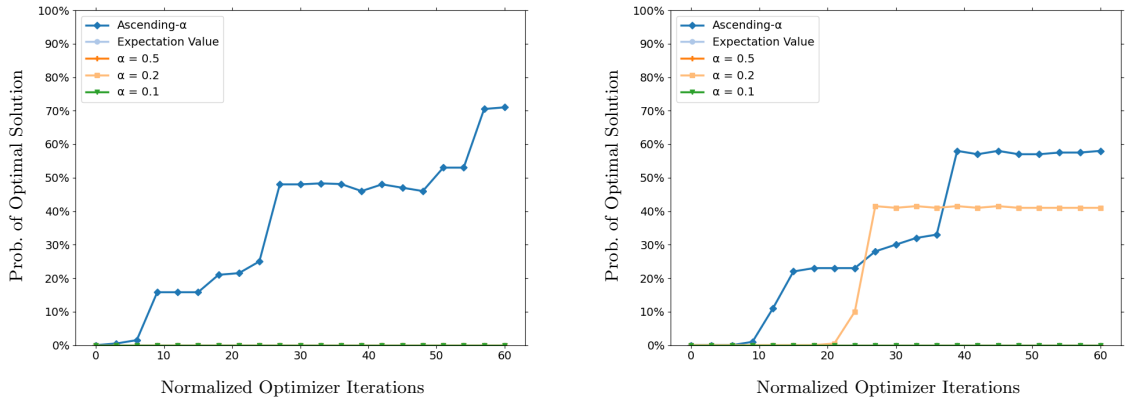

Figure 4.4: *MaxCut* instances with 17 vertices for random non-regular unweighted graphs. *Ascending-CVaR*, drawn with a blue line (diamond marker), results in a fast and high overlap with the optimal solution in contrast to constant CVaR.

find the optimal solution in under 10 normalized iterations, which, by the end of the optimization, is able to increase the probability up to 70%. Notably, the expectation value or constant CVaR completely fail. The right part of the figure gives another example of our approach performing better. This instance constitutes an example where smaller α's do not lead to better performance for constant CVaR[3]. We can see in the figure that while $\alpha = 0.1$ failed, $\alpha = 0.2$ was able to achieve a high-quality parameterized state. This is another indication of why our approach is more flexible.

4.5.1.2 CVaR$_{\alpha_t}$-QAOA

Solving the MaxCut problem using QAOA, with small-depth circuits, does not seem a very promising approach in any of the methods considered (constant CVaR or Ascending-CVaR). Regarding speed, all methods converged equally fast but in states with small overlap with the solution (with relatively small differences within different approaches). Having said that, as explained below, our method still gives improved performance.

While CVaR$_{\alpha_t}$-VQE optimization results in high overlap states, CVaR$_{\alpha_t}$-QAOA produces "flat" states, a behaviour also observed in [61]. These states have almost equal probability amplitudes to the majority of the computational basis states. For the MaxCut problem, as noted in [78], it seems that the states produced with QAOA with small p result in states with energy close to the (random) initialization point. The spread of the

[3]In most cases, small α gives better performance, but one cannot know apriori which is the suitable α in the constant CVaR case.

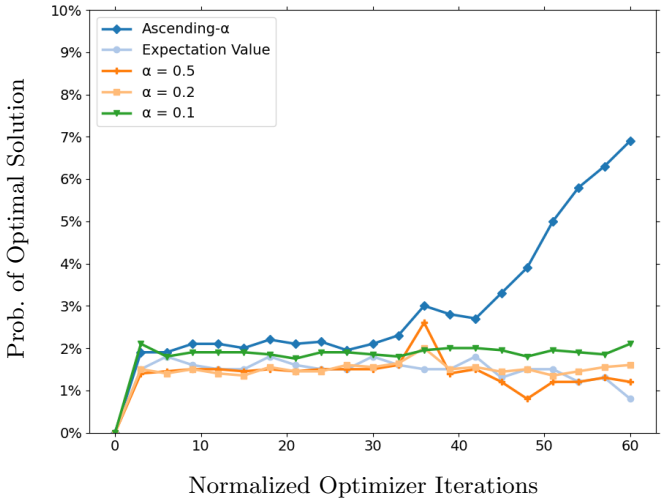

Figure 4.5: CVaR_{α_t}-QAOA optimization with linear ascending for a MaxCut instance of 17 qubits. The blue line (diamond marker), indicating the ascending optimization, results in more than a 100% increase in the overlap with the optimal solution in contrast to the expectation value or constant CVaR optimization.

energies does increase with p, possibly leading to a state close to the ground state, but in our analysis, we focused on small $p \leq 6$. Intuitively, the main reason why QAOA cannot achieve the same probability amplitudes as VQE in the same depth is due to having a smaller number of parameters as well as the architecture of the ansatz [65].

Note that the parameter space is filled with sub-optimal local minima. Constant CVaR objective functions with different confidence level α lead to different energy landscapes. This means that a local minimum for a confidence level α_1 does not, in general, correspond to a local minimum for a confidence level α_2 if $\alpha_1 \neq \alpha_2$. This is probably the reason that we get improved performance. For example, Figure 4.5 shows how Ascending-CVaR can avoid local minima. In this example, all constant CVaR achieve less than 3% overlap with the ground state, while the Ascending-CVaR gives 7%.

4.5.2 Number Partitioning

On *Number Partitioning* we tested instances with 17 to 20 integers, on both VQE and QAOA.

NP	Successful Instances					Average Overlap				
	α_t	0.1	0.2	0.5	1	α_t	0.1	0.2	0.5	1
N_1	87	85	66	16	2	54.17	11.50	16.56	7.94	0.99
N_2	80	69	29	11	0	48.33	10.24	7.56	5.88	0.4
N_3^*	95	58	24	9	0	56.85	8.24	5.84	3.45	0.16

Table 4.2: Results table for the *Number Partitioning* problem (VQE) for the three different sets N_1, N_2 and N_3^*, where the star at the last set indicates that we used the sigmoid ascending function.

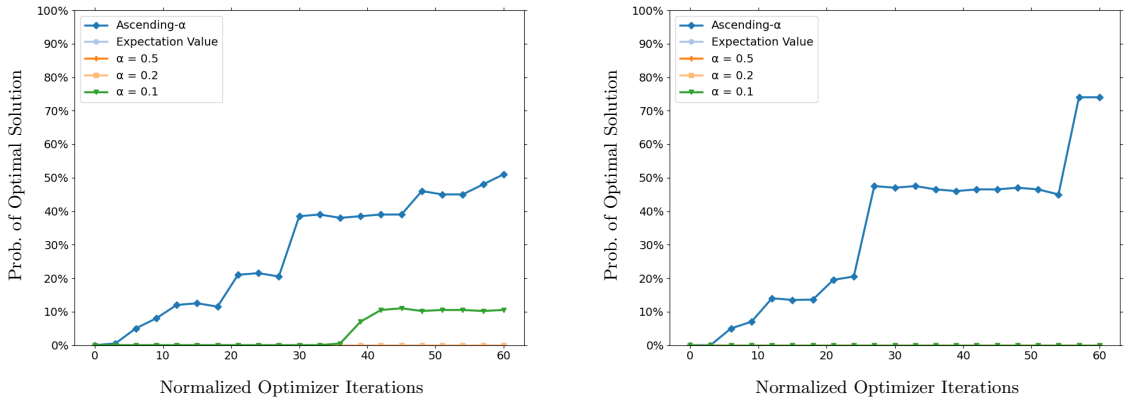

Figure 4.6: Probability of sampling the optimal solution for *Number Partitioning* instances with 17-20 integers uniformly drawn from the sets $N_1 = \{0, \ldots, 200\}$ (on the left) and $N_2 = \{0, \ldots, 500\}$ (on the right). The blue line (diamond marker), indicating *Ascending-CVaR* outperforms constant CVaR in terms of speed and overlap with the optimal solution.

4.5.2.1 CVaR$_{\alpha_t}$-VQE

On CVaR$_{\alpha_t}$-VQE, we tested 300 instances with 17 to 20 integers, sampled randomly from three sets; $N_1 = \{0, \ldots, 200\}$, $N_2 = \{0, \ldots, 500\}$ and $N_3 = \{0, \ldots, 750\}$. We highlight that the smaller the set from which the numbers are uniformly drawn, the easier for the optimizer to find the optimal solution. The results are summarised in Table 4.2.

For the first two sets, we used a linear ascending function with an ascending factor $\lambda = 0.03$. Further optimization of the parameter may lead to either faster convergence or more successful instances. Either way, the Ascending-CVaR method outperforms constant CVaR and the expectation value objective function on the aforementioned metrics (e.g. see typical performance on Figure 4.6).

For the last set, N_3, constant CVaR and the expectation value as objective functions

struggled to achieve even a small overlap with the optimal solution. Indicatively, at 40% of the cases, none of the constant CVaR objective functions could be successful Recall that successful in our convention means to achieve overlap of at least 10% with the optimal solution.. We found that by choosing a sigmoid ascending function, the optimizer is able to attain a high-quality parameterized state and succeed in the majority of instances (95%). The trade-off is that using the sigmoid ascending function, in contrast to linear ascending, comes with some cost of more circuit shots in order to achieve the same accuracy. Note also that the linear ascending function, while performing worse than the sigmoid, was still more successful than the constant CVaR objective functions.

4.5.2.2 CVaR$_{\alpha_t}$-QAOA

While CVaR$_{\alpha_t}$-VQE optimization efficiently achieved a high overlap state already within the first layer for instances drawn from the two sets N_1 and N_2, CVaR$_{\alpha_t}$-QAOA failed to achieve a high overlap on small depths. To address this issue without having to increase the depth of the ansatz, we chose to work on instances drawn from the smaller set $M = \{0, \ldots, 50\}$. For the Number Partitioning problem, the cost function's parameter space is highly dependent on the set from which we draw the numbers. The unitary transformation $e^{i\gamma H_C}$ is composed of $e^{i\gamma n_k n_l \sigma_z^k \sigma_z^l}$ terms where n_k, n_l correspond to the numbers on the k and l index respectively. The parameter γ is then restricted to $0 \leq \gamma < 2\pi/(n_j n_m)$ with n_j and n_m corresponding to the two smallest numbers of the set.

Our method succeeds in finding quantum states with higher overlap that are unreachable with constant CVaR optimization, possibly because it avoids the high amount of local minima. Indicatively, in Figure 4.7, we see an example where Ascending-CVaR achieves more than double overlap with the optimal solution than other methods but is still below the threshold of 10% required to classify this as a "successful run".

4.5.3 Portfolio Optimization

On *Portfolio Optimization* we tested instances with 16 to 20 assets, on both VQE and QAOA, with a budget drawn uniformly at random from the set $B = \{0, ...n\}$ where n is the number of assets and many different risk factors q.

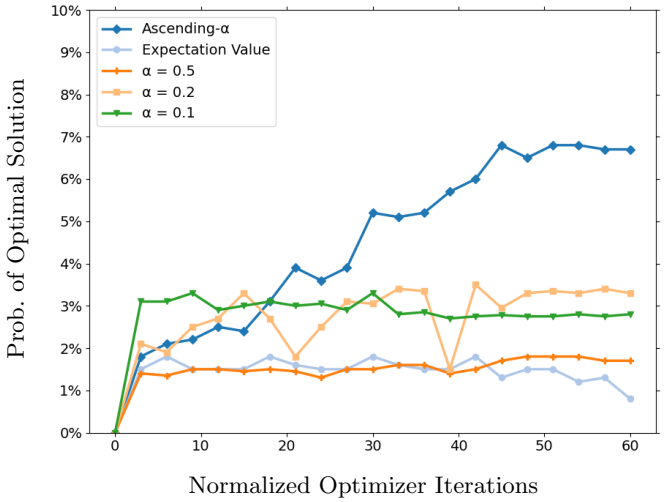

Figure 4.7: CVaR$_{\alpha_t}$-QAOA for an 18-integer instance Number Partitioning problem with $p = 4$. The blue line (diamond marker), indicating ascending-CVaR optimization, is able to achieve a 100% increase in the overlap with the optimal solution with respect to the other objective functions.

Portfolio Optimization	Successful Instances					Average Overlap				
	α_t	0.1	0.2	0.5	1	α_t	0.1	0.2	0.5	1
Random Portfolios	100	100	100	16	1	63.25	13.35	24.74	9.42	0.64

Table 4.3: Results table for the *Portfolio Optimization* problem (VQE) for 100 random Portfolios with 16 to 20 assets.

4.5.3.1 CVaR$_{\alpha_t}$-VQE

We used linear ascending with an ascending factor $\lambda = 0.045$, and the confidence level was initialized on $\alpha_0 = 0.01$. The results are summarized in Table 4.3. In Figure 4.8, we see the typical performance of two different instances where we plotted the probability of obtaining the optimal solution over the normalized number of optimizer iterations for the CVaR$_{\alpha_t}$-VQE.

We highlight the fact that Ascending-CVaR and constant CVaR with $\alpha = 0.1, 0.2$ succeed in achieving at least 10% overlap on all instances tested (see results on Table 4.3), while the expectation value ($\alpha = 1$) failed in almost all cases. Moreover, it is worth noting that our method offers a significant improvement in comparison with all the other approaches in the speed that this overlap was achieved (in terms of normalized optimizer iterations and circuit repetitions) and in the overall magnitude of the overlap achieved

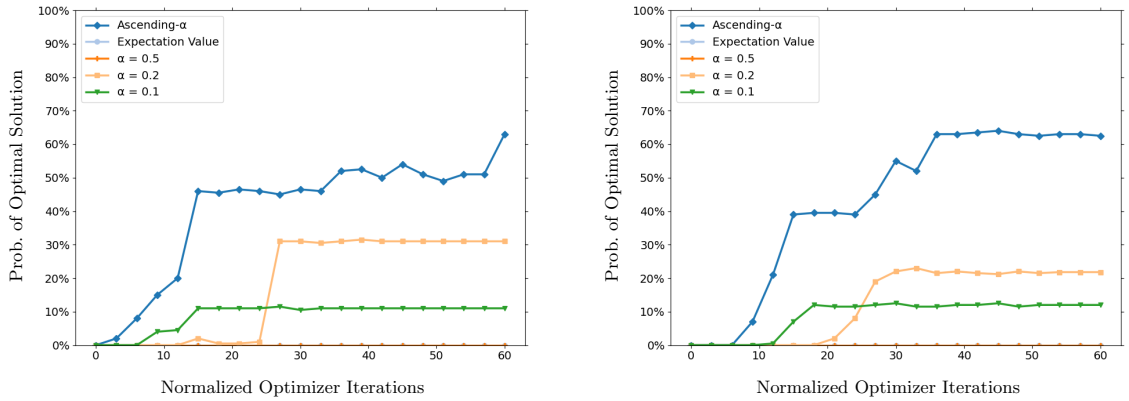

Figure 4.8: Portfolio Optimization problem for 18 and 20 asset instances with linear ascending and $\lambda = 0.045$. The blue line (diamond marker) indicates the CVaR$_{\alpha_t}$-VQE optimization, which already within the first 15 optimizer iterations has achieved over 40% overlap, compared to constant α where either fail ($\alpha = 0.5, 1$) or lead to slower and sub-optimal convergence ($\alpha = 0.1, 0.2$).

(see also Table 4.3).

4.5.3.2 CVaR$_{\alpha_t}$-QAOA

CVaR$_{\alpha_t}$-QAOA, similarly with [61], underperforms significantly in terms of overlap with the optimal state, compared to CVaR$_{\alpha_t}$-VQE. Specifically, keeping the depth as in previous parts and without increasing the shots each circuit is implemented, all methods fail to achieve overlap with the optimal solution well below 1%. There are several reasons for this failure, including the *Reachability Deficits* [140] and the large problem density [75]. This, however, goes beyond the focus of this chapter, which is to find a way to improve the performance of previously used objective functions. To illustrate the improvement, we could have used (a significantly) larger number of shots, where Ascending-CVaR would start showing better performance. This would make the comparison with other problems unfair (where in all cases, we used the same "normalized" number of shots), and it would still not present a practical way to solve the Portfolio Optimization problem (VQE is much better), so we omitted it.

4.6 Additional Experiments

In this section, we provide some additional experiments for the Ascending-CVaR method.

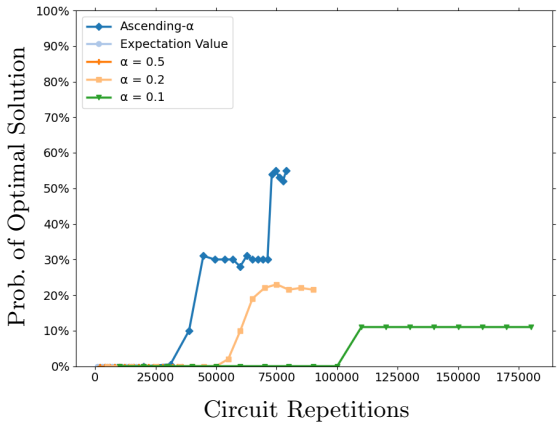

Figure 4.9: Probability of sampling an optimal solution over the circuit repetitions for a Number-Partitioning instance.

4.6.1 Circuit Repetitions

In this subsection, we demonstrate how our method outperforms the previously used objective functions in terms of real circuit repetitions and the quality of the output state. We set our "default" circuit repetitions to $K = 1000$, which we then scale up along the discretely increasing α using the expression K/α_t for each given time. While one may think that this would weaken our results, as illustrated below, it seems that in terms of circuit repetitions, our method converges to the chosen threshold of 10% faster than the best of constant CVaR or the expectation value approaches.

4.6.2 Numerical analysis of Ascending factor

In this subsection, we illustrate how the performance of our algorithm depends on the choice of the *ascending factor* λ. We numerically tested a large number of instances of sizes from 16 to 20 qubits and observed that the algorithm performed optimally for ascending factors drawn from the set $[0.025, 0.045]$. For this reason, as an example, we choose to draw the behavior of our algorithm for two random instances (one for Portfolio Optimization and one for Number Partitioning) for different choices of the hyperparameter λ.

The performance of our method is sensitive to the choice of λ. A small λ still converges to an optimal solution but requires a large number of iterations, compared to λ chosen from the set $[0.025, 0.045]$ which is able to attain a 10% within a small number of

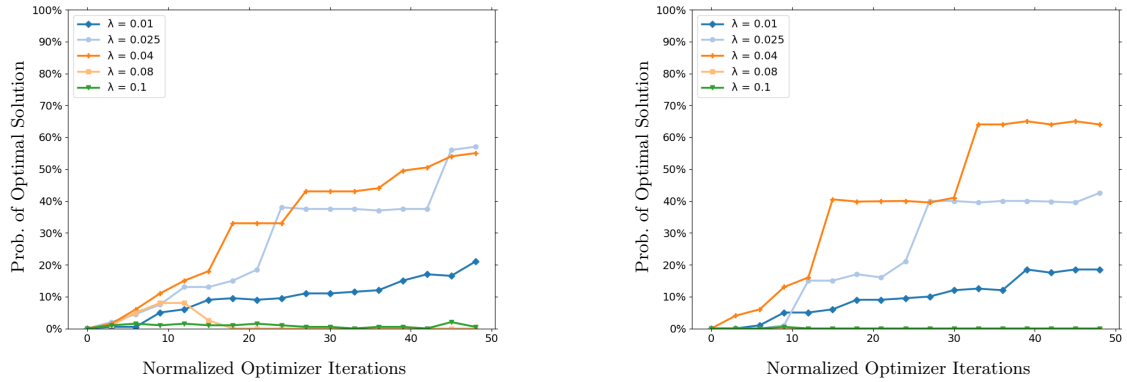

Figure 4.10: Performance of *Ascending-CVaR* algorithm with linear ascending for different choices of the ascending factor λ. The yellow (square marker) and green (down-pointing marker) lines, which refer to $\alpha = 0.08$ and $\alpha = 0.1$, respectively, are not able to reach a good approximation to the optimal solution. On the other hand, the orange (line marker) and light blue (circle marker) lines which correspond to $\alpha = 0.04$ and $\alpha = 0.025$ are both able to achieve an overlap larger than 50%. Finally, the dark blue line (diamond marker) is still able to reach a good approximation of the ground state, but it lacks in terms of speed of convergence and magnitude of overlap achieved. The graph on the left refers to a Number Partitioning problem while the graph on the right shows a Portfolio Optimization Problem.

iterations. On the other hand, choosing λ to be large (hoping for a faster convergence) fails to achieve even a minor overlap with the optimal solution. A careful tuning of λ is, therefore, necessary for the optimal performance of the algorithm given the size and class of the problem at hand.

Chapter 5

Adiabatic quantum computing with parameterized quantum circuits

5.1 Introduction

As we discussed in previous sections, conventional hybrid quantum/classical algorithms rely on a continuous feedback loop between the quantum computer that generates and measures non-classically-simulatable parameterized quantum states and the classical computer that updates the parameters towards the direction that minimizes the loss, using a classical optimization algorithm.

Despite the vast number of applications and research interest, the true performance of these algorithms and whether they can provide a valuable advantage over their pure classical counterparts is still an open question. Recapitulating, a major reason is that the emerging objective function landscapes of VQAs are filled with a high number of local minima [37, 38] and as the number of parameters increases and the ansatz families become more expressive, the classical optimization algorithms require exponentially many resources to navigate in these barren plateaux [42, 44, 45, 77].

To further understand the geometry and trainability of the underlying non-convex landscapes and subsequently understand the limitations of these algorithms, a new field called Quantum Landscape Theory (QLT) was introduced [21–28]. In QLT, the main objective is to understand how the loss function landscapes emerge but, at the same time, understand some of their properties. In this chapter, we aim to contribute to the field of QLT by quantifying how much the global minima of these objective functions are

shifted if we introduce a small perturbation in the initial Hamiltonian. This information, as we discuss later, has special importance as it can be used as the basis of a hybrid quantum/classical algorithm.

On the other end, conventional techniques for quantum computing with guaranteed performance, such as *Adiabatic Quantum Computing* (AQC) [141] (see Sec. 5.1.1) require depth (or else coherent evolution for large time interval) that is unreachable for current quantum devices. In AQC the system is initialized in the ground state of an easy-to-compute ground state and the Hamiltonian is "slowly" varied until it becomes the Hamiltonian of interest. However, the system must be varied sufficiently slowly so that it remains in the instantaneous ground state throughout the evolution. Then, at the final time t_f, the system will be found in the ground state of the desired Hamiltonian. The time taken to complete the evolution quantifies the cost/resources required for a given computation. What determines the minimum time that suffices for the problem to be solved (i.e. how to ensure adiabaticity) is the spectral gap [142]. The total evolution time t_f must scale as the inverse of the spectral gap, meaning that problems in which the gap becomes exponentially small [143] require exponentially large time and thus cannot be efficiently solved. Moreover, addressing the effect of noise in the adiabatic quantum evolution is also a complicated task.

In the past few years, the notions and ideas of AQC have tried to be incorporated into the NISQ literature [144–147]. In [145, 146] the authors incorporated certain ideas from AQC into the Variational Quantum Eigensolver. Specifically, they defined a (discrete) parameterized Hamiltonian similar to the one used in AQC, and they started from a Hamiltonian with a known ground state that belongs to a certain ansatz family of parametrized states. Then, they iteratively tried to minimize the expectation value of the (parameterized) Hamiltonian by using the output parameters at every step as the starting point/initialization for the parameters of the next. Then they argued that the final output would be close to the optimal angles and the whole procedure could be used as a warm-starting method [101]. Relevant to this work, [147] proposed a method to variationally simulate the adiabatic evolution, and [148] proposed to enhance the Quantum Approximate Optimization Algorithm (QAOA) with additional counterdiabatic driving terms which were shown to outperform the standard QAOA for the problems they investigated. Finally, [149] used recurrent neural networks to simulate an annealing framework and showed that on average their method outperforms simulated annealing on several spin-glass problems.

5.1.1 Adiabatic Quantum Computing

Adiabatic Quantum Computing (AQC) seeks to evolve a state under a time-dependent Hamiltonian $H(t)$. More specifically, a system of qubits is initialized into an easy-to-prepare ground state of a Hamiltonian H_0. Then, the system is allowed to interact through the Hamiltonian

$$H(t) = \left(1 - \frac{t}{t_f}\right) H_0 + \frac{t}{t_f} H_1, \ t \in [0, t_f]. \tag{5.1}$$

If the Hamiltonian is gapped and the evolution is "slow enough" so that the system of qubits always remains in the instantaneous ground state throughout the evolution, then at the final time t_f the system will be in the ground state of the desired Hamiltonian H_1. There are exact bounds to restrict the time t_f in order to ensure adiabaticity [150, 151] and is correlated to the spectral gap, i.e. the energy difference between the ground state and the first excited state. Specifically, exponentially (with the system size) small gaps require exponentially large evolution time the Hamiltonian (5.1).

The building block (and inspiration) of AQC is the *Adiabatic Theorem* which states that an evolving quantum system under a time-dependent Hamiltonian $H(t)$ will remain in the instantaneous ground state as long as the system never "receives" enough energy to make a transition to the instantaneous first excited state. The sufficient energy to make a transition is bounded by the spectral gap. As a result, all AQC-inspired algorithms must have a runtime that is dependent on the minimum spectral gap of the evolution. There are problems, such as solving linear systems of equations or search-engine problems, where bounding the spectral gap is possible. In [152], *Costa et al.* were able to show that a discrete version of AQC can achieve an asymptotically optimal scaling for solving linear systems while in [153] the authors were able to provide a polylogarithmic (to the system size) AQC algorithm for the PageRank problem.

As proven in [143], the total unitary evolution $U(t_f, 0) = e^{-i \int_0^{t_f} H(t) dt}$ required to interpolate between the two Hamiltonians can be approximated by $M = \mathcal{O}(\text{poly}(n) t_f)$ discrete unitary evolutions where n is the system size. Clearly, the large depth that is required to approximate the adiabatic evolution makes it intractable for near-term devices as the system size increases. Thus, it would be useful to examine whether we can use a trade-off between trainable parameters and circuit depth.

5.1.2 Notation

At this point, it is important to clarify our notation and highlight the different Hamiltonians used throughout this chapter.

- H_0: The initial Hamiltonian at time $t = 0$, for which we know the parameters that produce its ground state. For all our experiments, we choose $H_0 = -\sum_j \sigma_j^x$ with a ground state $|\psi_0\rangle = |+\rangle^{\otimes n}$.
- H_1: This is the target Hamiltonian that the user aims to find its ground state. The goal is to identify the parameters of the parameterized quantum circuit that generates the ground state of H_1.
- V: This Hamiltonian corresponds to the perturbation that we add in each iteration. In the case of Algorithm 4, we choose the perturbation Hamiltonian to be $V \equiv H_1 - H_0$.
- H_s: This is the starting Hamiltonian *at each step of the algorithm* (or else the unperturbed Hamiltonian). It is equivalent with H_0 at $t = 0$, but it could be any intermediate Hamiltonian for which we have found its ground state (in the previous step) using Theorem 5.1.
- H_λ: This is the perturbed Hamiltonian on all intermediate steps (or else the Hamiltonian that we seek its ground state on intermediate steps). That is, on every iteration, we start with the ground state of H_s and we use Theorem 5.1 to find the ground state of $H_\lambda = H_s + \lambda V$. H_λ is also equivalent to H_1 at time $t = t_f$ when the algorithm terminates.

5.2 Adiabatic Quantum Computing with Parameterized Quantum Circuits

In this section, we will present our two main results: A theorem that quantifies how small changes in the Hamiltonian affect the position of the global minima, and a hybrid quantum-classical algorithm, which we call AQC-PQC, that utilizes the aforementioned theorem and ideas from AQC to find the best approximation of the ground state of a Hamiltonian within a family of quantum states obtained using a parameterized quantum

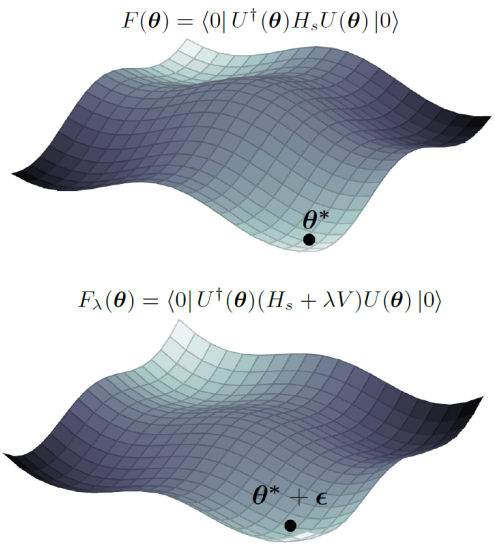

Figure 5.1: Variation of the loss function landscape for a small variation in the Hamiltonian. The global minimum shifts from the optimal point $\boldsymbol{\theta}^*$ of the Hamiltonian H_s to the optimal point $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$ of the Hamiltonian $H_s + \lambda V$.

circuit. A comparison with other approaches is given at the end of the section, while the proof of the theorem follows in Sec. 5.3.

Consider a Hamiltonian H_s, whose ground state, just like in AQC, is known. Let also a parameterized quantum circuit $U(\boldsymbol{\theta})$, prepared with parameters $\boldsymbol{\theta}^*$ that generates the ground state $|\psi(\boldsymbol{\theta}^*)\rangle$ of H_s. The first question that we want to answer is: "If the Hamiltonian H_s is deformed by a small amount λV ($H_\lambda = H_s + \lambda V$), what is the shift vector $\boldsymbol{\epsilon}$ that will translate the system from the initial ground state $|\psi(\boldsymbol{\theta}^*)\rangle$ of H_s to the ground state $|\psi(\boldsymbol{\theta}^* + \boldsymbol{\epsilon})\rangle$ of the slightly deformed Hamiltonian H_λ". The answer to this question is given in Theorem 5.1.

Theorem 5.1. *Consider a* parameterized quantum circuit *defined via the unitaries $U(\boldsymbol{\theta})$, and the corresponding states $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle$. We are given a Hamiltonian H_s and the angles $\boldsymbol{\theta}^*$ that minimize its energy, i.e. $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \langle\psi(\boldsymbol{\theta})| H_s |\psi(\boldsymbol{\theta})\rangle$. If we perturb the Hamiltonian H_s by a small amount λV with $\lambda \ll 1$, and $\|H_s\| \approx \|V\|$, then there exists a shift vector $\boldsymbol{\epsilon}$ such that, with high probability, the state $|\psi(\boldsymbol{\theta}^* + \boldsymbol{\epsilon})\rangle$ is the ground state of the perturbed Hamiltonian $H_\lambda = H_s + \lambda V$ and the shift vector is the solution of the*

following mathematical problem:

$$\begin{aligned} &min\ \|\boldsymbol{\epsilon}\| \\ &subject\ to:\ A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0, \\ &\boldsymbol{H}^{\lambda}\big|_{\boldsymbol{\theta}^*+\boldsymbol{\epsilon}} \succcurlyeq 0, \end{aligned} \tag{5.2}$$

where $\boldsymbol{H}^{\lambda}\big|_{\boldsymbol{\theta}^*+\boldsymbol{\epsilon}}$ *is the Hessian evaluated at the shifted point,* $\boldsymbol{Q} = \sum_i Q_i \hat{\boldsymbol{e}}_{\boldsymbol{i}}$ *is a vector and* A *is a matrix that are defined via their elements*

$$\begin{aligned} Q_i =& \lambda \frac{\partial}{\partial \theta_i} \left(\langle \psi(\boldsymbol{\theta}) | \, V \, | \psi(\boldsymbol{\theta}) \rangle \right) \Big|_{\boldsymbol{\theta}^*} \\ A_{ij} =& \frac{\partial^2}{\partial \theta_i \partial \theta_j} \left(\langle \psi(\boldsymbol{\theta}) | \, H_\lambda \, | \psi(\boldsymbol{\theta}) \rangle \right) \Big|_{\boldsymbol{\theta}^*}. \end{aligned}$$

Intuitively, we are looking for the smaller shift (first line of Eq. 5.2) that has vanishing gradient (second line) which at the same time is a minimum (rather than saddle point or maximum), as given by the constraint in the Hessian matrix (third line). An important point is to note that the elements A_{ij}, Q_j and the Hessian matrix can all be calculated using expectations and derivatives for the unperturbed state $|\psi(\boldsymbol{\theta}^*)\rangle$. This means that if we use this approach iteratively (see below), to compute the "new ground state", one needs to prepare a fixed number of quantum states in each step[1].

The motivation behind Theorem 5.1 is outlined below. As we discuss next, one could iteratively use the aforementioned theorem and construct an algorithm that allows the ground state preparation of a target Hamiltonian. Consider the task of finding the ground state of *a target Hamiltonian* H_1. The user would utilize a parameterized quantum circuit and initialize the parameters in the ground state of *a different but known Hamiltonian* H_0. If the Hamiltonian is deformed sufficiently slowly (by introducing small perturbations) and at the final time the Hamiltonian H_0 has transformed into the target Hamiltonian H_1, then by iteratively applying Theorem 5.1 (after each small deformation) the user would reach the target ground state (or an approximation of it). Note, however, that the known ground state (of the Hamiltonian H_s) at each iteration is the ground state of the slightly deformed Hamiltonian of the previous iteration. We can now come back to the aim of the chapter, to obtain a method that uses PQC to approximate AQC. Our approach can be summarized in Algorithm 4.

[1]The details depend on the PQC used, e.g. on whether parameter-shift rules are applicable or not.

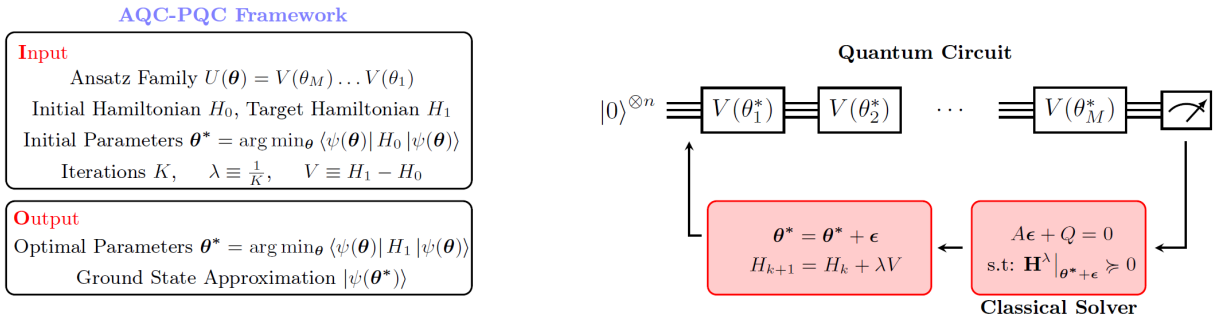

Figure 5.2: Adiabatic Quantum Computing with Parameterized Quantum Circuits. At every iteration, a series of observables are calculated for the instantaneous ground state. These observables form a linear system of equations whose solution corresponds to the shift vector $\boldsymbol{\epsilon}$. After the solution is found, the parameters $\boldsymbol{\theta}^*$ are shifted by $\boldsymbol{\epsilon}$ and the ground state is updated to $|\psi(\boldsymbol{\theta}^* + \boldsymbol{\epsilon})\rangle$. Then, a small perturbation is added to the Hamiltonian for the new observables to be calculated. Finally, a ground state approximation of H_1 is given at the output.

Algorithm 4: Adiabatic Quantum Computing with Parameterized Quantum Circuits

Input : Initial Hamiltonian H_0;
Target Hamiltonian H_1;
Ansatz family $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle$ with M parameters such that the ground state of H_0 and H_1 (or a good approximation of them) is contained within the ansatz;
$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \langle\psi(\boldsymbol{\theta})| H_0 |\psi(\boldsymbol{\theta})\rangle$;
Set of expectation values of observables: the Hessian $\mathbf{H}^\lambda$ and $\{Q_i, A_{ij}\}$ as given in Eq. (5.3);
Total steps K;
for $k = 1, 2, \dots, K$ **do**
  $H_k = (1 - \frac{k}{K})H_0 + \frac{k}{K}H_1$;
  Measure and estimate $\{Q_i, A_{ij}, \mathbf{H}^\lambda\}$ using a quantum processor;
  Use Eq 5.2 and a classical solver to calculate $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \dots, \epsilon_M)$;
  $\boldsymbol{\theta}^* = \boldsymbol{\theta}^* + \boldsymbol{\epsilon}$;
end
return $|\psi(\boldsymbol{\theta}^*)\rangle$

The algorithm can also be visualized in Figure 5.2. The main idea is the following. We start with discretizing AQC in a way similar to VAQC [145, 146], taking K steps.

We consider the (step-dependent) Hamiltonian

$$H_k = \left(1 - \frac{k}{K}\right) H_0 + \frac{k}{K} H_1. \tag{5.3}$$

Here the step subscript k has the role of the (discrete in our case) time. Let us set $\lambda := 1/K \ll 1$, and $V := (H_1 - H_0)$. We can rewrite the step-dependent Hamiltonian as

$$H_k = H_0 + k\lambda V. \tag{5.4}$$

We can easily see that $H_{k+1} - H_k = \lambda V$, and thus we can apply Theorem 5.1 for any consecutive pair of $\{H_k, H_{k+1}\}$. We start from H_s and initialize the algorithm with the known ground state that corresponds to the initial parameters $\boldsymbol{\theta}^*$. Then, for each step, we compute the shift vector $\boldsymbol{\epsilon}$ and add it to the parameters corresponding to the ground state of the previous step to obtain the ground state of the next step. For example, at the $k+1$ iteration, we can apply Theorem 5.1 using H_k as the known Hamiltonian H_s and $\lambda(H_1 - H_0)$ as the perturbation. Thus, by solving Eq. (5.2) we can calculate the shift vector $\boldsymbol{\epsilon}$ that will translate the system onto the ground state of the Hamiltonian H_{k+1}.

As we noted after our main theorem, to compute the shift vector we need (i) to estimate the expectation values of certain observables evaluated for the starting state using our quantum device and (ii) use a classical solver to solve Eq. 5.2. The shifted ground state is then used as the ground state for the next step. In the Kth (final) step, the Hamiltonian becomes H_1 and thus the ground state we recover is the desired ground state terminating the quantum/classical loop.

Comparisons. Our method differs significantly from both the traditional adiabatic evolution and VAQC proposed by [145, 146]. In AQC the adiabatic evolution, when run in a digital quantum computing device, is approximated by a series of Trotterized unitary evolutions. As a result, to simulate these unitaries, a circuit with a very large depth is needed which makes AQC inapplicable for near-term devices. In our approach, we take advantage of the fixed architecture of the parameterized quantum circuits, which supports quantum states that can be prepared with relatively high fidelity, while we still exploit the advantages and guarantees that adiabatic quantum computing offers (at least in the noiseless case). Similarly to variational algorithms, we delegate part of the computation to a classical processor to construct a hybrid algorithm. In AQC the (time-evolving) Hamiltonian needs to be physically implemented (or a Trotterized version of it). In our approach, we evaluate the energy corresponding to that Hamiltonian

by measuring the parameterized quantum states we produce, which allows us to easily consider more general Hamiltonians (for example ones that include terms of higher order than quadratic).

Our approach also differs significantly from variational quantum algorithms, offering several potential advantages. The main difference in our approach is that we do not perform energy minimization in the conventional way. As such, we do not have to rely on empirical choices of the hyperparameters for the success of our method as we do not iteratively follow the direction of the negative gradient. In our approach, the energy minimization is performed by finding the closest minimum, which is identified by solving the constrained linear system in Eq (5.2).

As we discussed, the classical part is a constrained linear solver (which can be performed very efficiently and with guarantees of finding the solution), while in variational approaches the user usually selects a first or second-order minimization method. Most of the limitations of VQAs come exactly from the optimization part and specifically due to two main bottlenecks. First of all, a random initialization of parameters may lead to either bad performance or *barren plateaux*. Secondly, the emerging landscapes of VQAs are filled with a vast amount of local minima or barren plateaux that make them untrainable. While there exist methods that try to overcome these limitations [51, 84, 85, 100] our algorithm offers a more robust strategy (see also for simple simulated experiments in Sec. 5.5).

Another key difference, and advantage of our approach, is that traditional variational quantum approaches require constant preparation of the quantum circuits. For each iteration, multiple quantum states need to be prepared (details depend on the classical optimizer used). As the classical optimization algorithm approaches the minimum, the number of shots and quantum state preparations increase significantly (as the gradients tend to zero). On top of that, we neither know in advance how many iterations would be required to reach convergence nor if the quantum state that we converge to is the correct ground state[2]. In contrast, in our strategy, we can mimic adiabatic quantum computing with only K steps, where K is the chosen number of discretization steps and is typically much smaller than the iterations a classical optimizer requires in VQAs. Therefore, we have a known number of quantum states that are to be prepared and

[2]It is a heuristic approach.

measured [3], and also have a guarantee that our method will find the solution if AQC can solve the problem efficiently and K is chosen suitably.

A final potential advantage compared to traditional variational approaches, is that quantum error mitigation methods may be more effective in our case. There are QEM methods (for example [154]) that are specifically applicable if the quantum state considered is close to the ground state. In our method, all quantum states used are close to the ground state of some time-dependent Hamiltonian, and thus these approaches should be more productive. A full analysis of these implications, as well as the questions stated in the above paragraph, are a subject for further research.

5.3 Parameterized Perturbation Theory

In this section, we will first prove Theorem 5.1, which reduces the problem of finding how a small perturbation of a Hamiltonian shifts the parameters that minimize the energy, to a constrained system of linear equations. We then give two ways to impose the constraint. Finally, for a special (but very widely used) class of families of parameterized quantum circuits, we give expressions of how to exactly evaluate the required derivatives and outline the method we follow for our simulated experiments in the next section.

Proof of Theorem 5.1. We first give the outline in four steps. To establish that we have found a minimum of a function we need to (i) check that the gradient of the function, evaluated at that point, vanishes (critical point) and that (ii) the second derivative (Hessian) corresponds to a positive semi-definite matrix (is a minimum). Our function is the (perturbed) energy $F_\lambda(\boldsymbol{\theta})$ corresponding to the Hamiltonian $H_\lambda = H_s + \lambda V$ of the parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$. The first step is to consider the gradient of a general shifted point $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$. For the new ground state, this gradient should vanish. However, since our aim is to find the shift vector $\boldsymbol{\epsilon}$, we cannot measure the energy of the shifted state until we know (or until we have a good guess for) the shift vector. The second step is to exploit the fact that the Hamiltonian is only slightly perturbed. We expect that the new ground state is close to the one we started, thus the shift vector $\boldsymbol{\epsilon}$ is small and we can expand the energy around the previous minimum using a Taylor expansion, while we are justified in keeping only the leading (linear) terms

[3] We require $\mathcal{O}\Big((1+\dim(\mathcal{N}_\kappa(A))M^2\Big)$ different quantum states for each step if we follow the approach used in Lemma 5.5

in $\boldsymbol{\epsilon}$. Now all the functions (and derivatives) required are evaluated for the previous (and known) value $\boldsymbol{\theta}^*$. The third step is to evaluate the derivatives either approximately using finite differences, or exactly if the PQC allows us to use the parameter shift rules (Eqs 2.4,2.5). This reduces the problem to a system of linear equations. This system (in general) has many solutions since some of the equations are not independent. If λ is sufficiently small, and for well-behaved Hamiltonians, the smallest shift vector that gives a minimum is the global minimum and thus the ground state. The final fourth step requires exactly this, to minimize over the possible solutions for the shift vector, while also confirming that the solution is indeed a minimum by checking the positivity of the Hessian.

Step 1. We consider the energy $F_\lambda(\boldsymbol{\theta})$ given by

$$F_\lambda(\boldsymbol{\theta}) = \langle 0| U^\dagger(\boldsymbol{\theta}) H_\lambda U(\boldsymbol{\theta}) |0\rangle . \tag{5.5}$$

We are interested in the value of the new ground state, i.e. the new minimum of the energy. We expect the new ground state, since the Hamiltonian is perturbed slightly by λV, to be close to the previous ground state, i.e. we search for some point $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$. For this point to be a minimum, the gradient of the energy should vanish,

$$\frac{\partial}{\partial \theta_i} F_\lambda(\boldsymbol{\theta}) \bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*+\boldsymbol{\epsilon}} = 0 \quad \forall \quad i. \tag{5.6}$$

Step 2. We can expand the energy around the old ground state, using a Taylor expansion, as

$$F_\lambda(\boldsymbol{\theta}^* + \boldsymbol{\epsilon}) = F_\lambda(\boldsymbol{\theta}^*) + \sum_{i=1}^{M} \epsilon_i \frac{\partial}{\partial \theta_i} F_\lambda(\boldsymbol{\theta}^*) + \mathcal{O}\left(\|\boldsymbol{\epsilon}\|^2\right). \tag{5.7}$$

For suitably small perturbation (i.e. sufficiently small λ), the shift vector $\boldsymbol{\epsilon}$ is also small and we can approximate accurately the energy by keeping up to the linear in $\boldsymbol{\epsilon}$ term of the Taylor expansion and plug it into Eq. (5.6) to impose a vanishing gradient,

$$\frac{\partial}{\partial \theta_i} F_\lambda(\boldsymbol{\theta}^*) + \frac{\partial}{\partial \theta_i} \left(\sum_{j=1}^{M} \epsilon_j \frac{\partial}{\partial \theta_j} F_\lambda(\boldsymbol{\theta}^*) \right) = 0 \quad \forall \quad i. \tag{5.8}$$

By noting that

$$F_\lambda(\boldsymbol{\theta}^*) = \langle \psi(\boldsymbol{\theta}^*)| H_s |\psi(\boldsymbol{\theta}^*)\rangle + \lambda \langle \psi(\boldsymbol{\theta}^*)| V |\psi(\boldsymbol{\theta}^*)\rangle$$

and that

$$\frac{\partial}{\partial\theta_i}\langle\psi(\boldsymbol{\theta}^*)|\, H_s\, |\psi(\boldsymbol{\theta}^*)\rangle = 0 \quad \forall \quad i,$$

and because $\boldsymbol{\theta}^*$ is the ground state of H_s, we have

$$\frac{\partial}{\partial\theta_i}F_\lambda(\boldsymbol{\theta}^*) = \lambda\frac{\partial}{\partial\theta_i}\langle\psi(\boldsymbol{\theta}^*)|\, V\, |\psi(\boldsymbol{\theta}^*)\rangle := Q_i. \tag{5.9}$$

Similarly, we can define

$$A_{ij} := \frac{\partial^2}{\partial\theta_i\partial\theta_j}F_\lambda(\boldsymbol{\theta}^*). \tag{5.10}$$

Eq. (5.8) becomes

$$Q_i + \sum_j A_{ij}\epsilon_j = 0 \quad \forall \quad i, \tag{5.11}$$

the system of equations in Eq. (5.2) in Theorem 5.1.

Step 3. Both the Q_i and the A_{ij} involve derivatives of expectation values evaluated at the known point $\boldsymbol{\theta}^*$. As we have seen in Sec. 2.1, one can evaluate such derivatives by computing the expectation values at a number of points (one or two per dimension of the parameter space) related to $\boldsymbol{\theta}^*$. In the general case, those points need to be very close to $\boldsymbol{\theta}^*$, and the result is an approximation (finite differences). In the special case that the generators have a certain specific form (see later) one can evaluate exact derivatives using the parameter-shift rule (Eqs (2.4), (2.5)). The exact choice depends on the PQC that one considers, suggesting that PQC that admit parameter-shift rules would perform more accurately in our method.

Step 4. To determine the (small) shift in the parameter space that the ground state moved because of a small perturbation, we need to find (i) the turning point closest to the old ground state (vanishing determinant) that also (ii) is actually a minimum. To ensure the former we need to take the $\boldsymbol{\epsilon}$ with the minimum norm, while for the latter we need to ensure that the Hessian $H^\lambda_{jk} = \frac{\partial^2}{\partial\theta_j\partial\theta_k}F_\lambda(\boldsymbol{\theta})$ evaluated at the new point $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$ is a positive semi-definite matrix[4]. The shift vector is therefore the solution to the problem

$$\begin{gathered}\min \|\boldsymbol{\epsilon}\| \\ \text{subject to: } A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0, \\ \mathbf{H}^\lambda\big|_{\boldsymbol{\theta}^*+\boldsymbol{\epsilon}} \succcurlyeq 0.\end{gathered} \tag{5.12}$$

[4]In [155], the authors numerically analyzed the objective function landscapes that appear in the highly parameterized vector spaces of variational quantum algorithms. They noted that in most cases, the Hessian matrix at the local and global minima is positive semi-definite, with the zero eigenvalue being highly degenerate.

□

To solve the constrained problem in Eq. (5.2), both a classical and a quantum device is needed. Ideally, we would like to first measure some expectation values, and then, with these values as input, use a (fully) classical solver to find the shift vector. However, the constraint involves checking the (semi) positivity of a matrix (Hessian) which is evaluated for the new ground state $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$. Since the shift vector is not known (yet), one would think that the constraint cannot even be defined unless one has a candidate shift vector. While this approach is possible (see *remark* later), we can also exploit, once more, the fact that the shift vector is small for small perturbations.

Lemma 5.1. *The mathematical problem of Eq. (5.2) can be solved using expectation values of observables and their derivatives evaluated at the known point $\boldsymbol{\theta}^*$. Specifically, the Hessian at $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$ can be approximated using this expression*

$$\boldsymbol{H}^{\lambda}_{jk}\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*+\boldsymbol{\epsilon}} = \boldsymbol{H}^{\lambda}_{jk}\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} + \sum_{i=1}^{M} \epsilon_i \frac{\partial}{\partial \theta_i} \boldsymbol{H}^{\lambda}_{jk}\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \tag{5.13}$$

Proof. The linear set of equations is already defined at $\boldsymbol{\theta}^*$ while the Hessian is obtained by Taylor-expanding and keeping up to linear terms. □

Here we should note that to compute the Hessian in the way described, we require up to third derivatives of the ground state energy. Since each derivative requires us to evaluate the state in at least one different point per dimension of the parameter space, computing third derivatives would require $O(M^3)$ quantum states.

Remark. An alternative approach would be to use a method that has more cycles of classical-quantum subroutines. Specifically, one could first evaluate expectation values at $\boldsymbol{\theta}^*$ and solve the simpler (unconstrained) system

$$\begin{aligned} &\min \|\boldsymbol{\epsilon}\| \\ &\text{subject to: } A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0. \end{aligned} \tag{5.14}$$

Then, using the trial shift-vector $\tilde{\boldsymbol{\epsilon}}$, prepare the new set of states (of $O(M^2)$) to check if the Hessian at the point $\boldsymbol{\theta}^* + \tilde{\boldsymbol{\epsilon}}$ is positive semi-definite. If it is, one outputs $\boldsymbol{\epsilon} = \tilde{\boldsymbol{\epsilon}}$. If it is not, one goes back and finds a new candidate shift-vector and prepares a new

set of quantum states to check the Hessian at that point. The process terminates when a suitable solution is found. While this approach may require $O(M^2)$ preparations of quantum states, it has some disadvantages that led us to focus on Lemma 5.1: (i) It is not clear after how many rounds we are guaranteed (or likely) to find a suitable solution; (ii) We need to go back and forth between the classical and quantum processors; (iii) Optimized classical solvers for the constrained problem cannot be used, and a naive, trial-and-error method imposing the constraint is followed.

As a final point, we note that for most PQCs, for example, those that have Pauli rotations as parameterized quantum gates, one can use parameter-shift rules to evaluate the derivatives exactly.

Lemma 5.2. *Consider the statement in Theorem 5.1, where the parameterized quantum circuit is defined via unitaries that have generators g_j with two distinct eigenvalues $\pm r$. Then Eq. (5.2) can be evaluated exactly using*

$$\begin{aligned} Q_i &= \frac{\lambda}{2}\left(V\left(\boldsymbol{\theta}^* + \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}}\right) - V\left(\boldsymbol{\theta}^* - \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}})\right)\right) \\ A_{ij} &= \frac{1}{4}\left(F_\lambda\left(\boldsymbol{\theta}^* + \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}} + \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\right) - F_\lambda\left(\boldsymbol{\theta}^* - \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}} + \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\right)\right. \\ &\left. -F_\lambda\left(\boldsymbol{\theta}^* + \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}} - \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\right) + F_\lambda\left(\boldsymbol{\theta}^* - \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{i}} - \frac{\pi}{2}\hat{\boldsymbol{e}}_{\boldsymbol{j}}\right)\right) \end{aligned} \tag{5.15}$$

and similarly, the Hessian in Eq. (5.13) can be evaluated by taking third derivatives with the parameter-shift rule.

Proof. This follows directly using the parameter-shift rules given in Eqns (2.4), (2.5). □

5.4 Solving the constrained linear system

In this section, we will decompose the main mathematical problem incorporated in AQC-PQC and focus on all of its subsequent parts separately. As it is clear, the hardness in our approach comes in solving the main mathematical problem in Eq. (5.2):

$$\begin{gathered} \min \|\boldsymbol{\epsilon}\| \\ \text{subject to: } A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0, \\ \mathbf{H}^\lambda\big|_{\boldsymbol{\theta}^*+\boldsymbol{\epsilon}} \succcurlyeq 0, \end{gathered}$$

using the least number of classical and quantum resources.

Recall that we are searching for the minimum vector $\boldsymbol{\epsilon} \in \mathbb{R}^M$ that has zero gradient and is a (global) minimum. The norm of a vector $\|\boldsymbol{\epsilon}\|$ is a convex function. The same holds for the linear equation constraint, as the affine map $A\boldsymbol{\epsilon}+\boldsymbol{Q}$ is also a convex function. The main bottleneck in our problem is that the Hessian at the point $(\boldsymbol{\theta^*} + \boldsymbol{\epsilon})$ is not (in general) an affine map in $\boldsymbol{\epsilon}$, making it a non-convex problem. However, as we will see, this is not the case when $\boldsymbol{\epsilon}$ is small.

5.4.1 Equality Constraint

We start with the equality constraints:

$$A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0 \tag{5.16}$$

with $A \in \mathcal{S}^M$ ($\mathcal{S}^M = \{X|\ X^T = X\}$ is the space of symmetric matrices of dimension M) and $\boldsymbol{Q} \in \mathbb{R}^M$. The first step is to eliminate all *equality constraints*, as in most cases this linear system of equations is overdetermined. Consider any $\boldsymbol{\epsilon_0}$ that is a solution of Eq. (5.16), i.e. $A\boldsymbol{\epsilon_0} + \boldsymbol{Q} = 0$. It will be wise, for reasons that will become clear later, to choose $\boldsymbol{\epsilon}_0$ to be the smallest vector that satisfies the linear equation, i.e. $\boldsymbol{\epsilon_0}$ is the solution of the convex problem:

$$\begin{gathered}\min \|\boldsymbol{\epsilon}\| \\ \text{subject to: } A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0\end{gathered}$$

Such a vector is unique. Let $\mathcal{F}$ be the feasible set of solutions of Eq. (5.16):

$$\begin{gathered}\mathcal{F} = \{\boldsymbol{\epsilon}\big|A\boldsymbol{\epsilon} + \boldsymbol{Q} = 0\} \implies \mathcal{F} = \{\boldsymbol{\epsilon}|A(\boldsymbol{\epsilon} - \boldsymbol{\epsilon_0}) = 0\} \implies \\ \mathcal{F} = \{\boldsymbol{u} + \boldsymbol{\epsilon_0}|A\boldsymbol{u} = 0\}\end{gathered} \tag{5.17}$$

where in the last line we defined $\boldsymbol{u} \equiv \boldsymbol{\epsilon} - \boldsymbol{\epsilon_0}$. As a result, all $\boldsymbol{\epsilon} = \boldsymbol{u} + \boldsymbol{\epsilon_0}$, with $\boldsymbol{u} \in \mathcal{N}(A)$ (where $\mathcal{N}(A)$ is the null space of A) correspond to solutions of the linear system of equations. Since Eq. (5.16) is a linear equation (whose solution provides us with the critical points in the energy landscape) that is approximately equal to 0, it is more appropriate to define the notion of an κ-approximate null space (denoted as $\mathcal{N}_\kappa(A)$), i.e $A\boldsymbol{u} \approx 0$.

Definition 5.1. (κ-Approximate null space). *Let $A \in \mathcal{S}^M$ and let $\kappa > 0$. Consider the set of eigenvectors $\{\boldsymbol{v_1}, \ldots, \boldsymbol{v_l}\}$ of A such that $\|\boldsymbol{v_i}\| = 1$ and $\|A\boldsymbol{v_i}\| \leq \kappa$. The κ-approximate null space $\mathcal{N}_\kappa(A)$ is defined as $\mathcal{N}_\kappa(A) := span(\boldsymbol{v_1}, \ldots, \boldsymbol{v_l})$.*

We will now proceed and explain how one can construct an approximate null space. As a first step, we apply a singular value decomposition (SVD) of the matrix A. By doing so, the matrix A can be decomposed as:

$$A = U\Sigma V^T \tag{5.18}$$

where U, V are orthogonal matrices and Σ is a diagonal matrix with singular values of A as its entries. Let $\text{diag}(\Sigma) = (\sigma_1(A) > \sigma_2(A) > \ldots > \sigma_M(A) > 0)$ be the singular values of A. Since A is symmetric, $U = V$ and their columns are the eigenvectors of A. Additionally, the singular values σ_i are the absolute values of the eigenvalues of A, i.e. $\sigma_i = |\lambda_i|$. If $\boldsymbol{v_i}$ are the eigenvectors of A then from Eq. (5.18), we can write A as:

$$A = \sum_{i=1}^{M} \sigma_i \boldsymbol{v_i} \boldsymbol{v_i}^T \tag{5.19}$$

As our next step, we can apply a low-rank approximation of the matrix A. Specifically, from Eq. (5.19) we can keep all terms up to the k-th term. Let $A_k = \sum_{i \leq k} \sigma_i \boldsymbol{v_i} \boldsymbol{v_i}^T$ be the k-rank approximation of A. The error in the approximation is given in the Frobenius norm as:

$$\|A - A_k\|_F = \sqrt{\sum_{k+1}^{m} \sigma_i^2} \tag{5.20}$$

and is the optimal k-rank approximation according to *Eckart-Young-Mirsky Theorem*. As a result, the above analysis provides a recipe for how to define the κ-approximate null space $\mathcal{N}_\kappa(A)$. That is, one can set a threshold $\kappa > 0$ with $\kappa \approx 0$ so that all singular values smaller than κ are neglected. The basis of $\mathcal{N}_\kappa(A)$ is then clearly $(\boldsymbol{v_{k+1}}, \ldots, \boldsymbol{v_M}) = \text{span}(\mathcal{N}_\kappa(A))$ with $\dim(\mathcal{N}_\kappa(A)) = M - k$. Consider now the most general vector $\boldsymbol{\mu} \in \mathcal{N}_\kappa(A)$

$$\boldsymbol{\mu} = c_{k+1}\boldsymbol{v_{k+1}} + \ldots + c_M \boldsymbol{v_M} = \sum_{i=k+1}^{M} c_i \boldsymbol{v_i} \tag{5.21}$$

with $c_i \in \mathbb{R}$. As a result, the main mathematical problem (5.2) has thus been reformulated to:

$$\begin{aligned} &\min_{\boldsymbol{\mu}} ||\boldsymbol{\epsilon_0} + \boldsymbol{\mu}|| \\ &\text{subject to: } \mathbf{H}^{\lambda}\big|_{\boldsymbol{\theta^*}+\boldsymbol{\epsilon_0}+\boldsymbol{\mu}} \succcurlyeq 0 \\ &\boldsymbol{\mu} \in \mathcal{N}_{\kappa}(A) \end{aligned} \tag{5.22}$$

Corollary 5.1. *Solving the main mathematical problem in Eq.* (5.2) *reduces the classical search to only* $\dim(\mathcal{N}_{\kappa}(A)) < M$ *parameters.*

As a result, we can reduce the overall hardness required in gradient approaches, where we usually have to optimize all the parameters of the parameterized quantum circuit. Relevant approaches where only a subset of the total parameters are varied have been previously utilized in [60, 156]. It is worth noting that in problems that we examine in section Sec 5.5, the dimension of the κ-approximate null space is usually much smaller than the total number of parameters M.

5.4.2 Positive-semidefinite constraint

Since we eliminated the equality constraint in Eq. (5.2), we will proceed to satisfy the second constraint, which corresponds to the Hessian. Specifically, our goal is to make the Hessian matrix positive-semidefinite ($\mathbf{H}^{\lambda}|_{\boldsymbol{\theta^*}+\boldsymbol{\epsilon}} \succcurlyeq 0$), as the solution of Eq. (5.2) must translate the system to the ground state of the perturbed Hamiltonian which is a (global) minimum. Making the Hessian matrix positive-semidefinite is equivalent to making its smallest eigenvalue greater or equal to zero.

Consider a matrix $X \in \mathcal{S}^M$ where $\mathcal{S}^M = \{X |\ X^T = X\}$ is the space of symmetric matrices. Clearly, any Hessian matrix belongs in $\mathcal{S}^M$.

Definition 5.2. (Minimum eigenvalue function). The function $f : \mathcal{S}^M \to \mathbb{R}$ that inputs a symmetric matrix X and outputs its minimum eigenvalue is defined as:

$$f(X) = \inf_{v}\{\boldsymbol{v}^T X \boldsymbol{v} \mid \|\boldsymbol{v}\| = 1\} \tag{5.23}$$

The minimum eigenvalue function has the following important property, highlighted in Lemma 5.3.

Lemma 5.3. *The function $f : \mathcal{S}^M \to \mathbb{R}$ that inputs a symmetric matrix X and outputs its minimum eigenvalue is concave. If $X_1, X_2 \in \mathcal{S}^M$ and $0 \leq \theta \leq 1$, then f satisfies Jensen's inequality:*

$$f[\theta X_1 + (1-\theta)X_2] \geq \theta f(X_1) + (1-\theta) f(X_2) \tag{5.24}$$

Proof. The proof follows immediately, as f is the pointwise infimum of a family of linear functions (i.e. $\boldsymbol{v}^T X \boldsymbol{v}$). For more details, see [124]. □

Definition 5.3. We define the composite function $h = f \circ \mathbf{H} : \mathbb{R}^M \to S^M \to \mathbb{R}$:

$$h(\boldsymbol{\epsilon}) = f(\mathbf{H}^\lambda|_{\boldsymbol{\theta}^* + \boldsymbol{\epsilon}}) \tag{5.25}$$

It is clear that the function h is not concave since the Hessian operator is not in general affine in $\boldsymbol{\epsilon}$. However, from our previous analysis, we have assumed that for small perturbations λ, the shift vector that translates the system onto the new ground state is also small. This allows us to define the affine approximation of the Hessian.

Definition 5.4. The affine approximation $\tilde{\mathbf{H}}$ of $\mathbf{H}$ is defined as:

$$\tilde{\mathbf{H}} = \mathbf{H}^\lambda|_{\boldsymbol{\theta}^*} + \sum_{k=1}^{M} \epsilon_k D_k|_{\boldsymbol{\theta}^*} \tag{5.26}$$

where the matrix D_k is defined as $D_k \equiv \frac{\partial \mathbf{H}^\lambda}{\partial \theta_k}$.

One can quantify the error of the affine approximation of the Hessian from the full matrix using techniques described in [157].

Lemma 5.4. *The function $h = f \circ \tilde{\boldsymbol{H}}$ defined as:*

$$h(\boldsymbol{\epsilon}) = f\Big(\boldsymbol{H}^\lambda|_{\boldsymbol{\theta}^*} + \sum_{k=1}^{M} \epsilon_k D_k|_{\boldsymbol{\theta}^*}\Big) \tag{5.27}$$

is concave in $\mathbb{R}^M$.

Proof. The proof follows similarly to Lemma 5.3, as h is the pointwise minimum of a family of affine functions. □

Our analysis has allowed us to express the problem as a semidefinite program. It is straightforward to see that if we used the κ-approximate null space defined in Definition 5.2 and use the fact that $\boldsymbol{\epsilon} = \boldsymbol{\epsilon_0} + \boldsymbol{\mu}$, where $\boldsymbol{\mu} \in \mathcal{N}_\kappa(A)$ then Eq. (5.2) can be reduced to its final form:

$$\begin{aligned} &\min_{\boldsymbol{\mu}} ||\boldsymbol{\epsilon_0} + \boldsymbol{\mu}|| \\ &\text{subject to: } f\Big(\mathbf{H}^\lambda + \boldsymbol{\nabla}_{\boldsymbol{\mu}}\mathbf{H}^\lambda\Big|_{\boldsymbol{\theta}^* + \boldsymbol{\epsilon_0}}\Big) \geq 0 \\ &\boldsymbol{\mu} \in \mathcal{N}_\kappa(A) \end{aligned} \tag{5.28}$$

where $\boldsymbol{\nabla}_{\boldsymbol{\mu}} = \boldsymbol{\mu}^T\boldsymbol{\nabla}$ is the directional derivative pointing in the κ-approximate null space. As a result, in Lemma 5.5, we can conclude what are the essential quantum resources to formulate and solve the mathematical problem defined in Eq. (5.2).

Lemma 5.5. *(Quantum Resources). The total number of quantum resources required at every step of the AQC-PQC algorithm scales as:*

$$\mathcal{O}\Big((1 + \dim(\mathcal{N}_\kappa(A))M^2\Big)$$

where A is the Hessian of the perturbed Hamiltonian at the ground state of the unperturbed Hamiltonian.

Proof. The proof follows immediately from our previous analysis. □

Finally, once the problem has been formulated, classical algorithms, such as interior point methods [124] or supergradient ascent methods [158], can be utilized to return the solution.

5.5 Simulated Experiments

We will first give an overview of the technical details of the experiments and we will also introduce the different classes of problems that we examined. Then, we will provide an analysis of our method for different choices of discretization steps.

5.5.1 Technical Details

For our experiments, we used both *Qiskit Statevector* and QuEST simulators which allow exact *noiseless* calculation of the expectation values.

For the parameterized family of gates, we chose a *hardware-efficient ansatz* consisting of a layer of R_y rotations for each qubit, followed by a series of controlled-Z operations applied in a nearest-neighbor fashion, and then finally another layer of R_y rotations (see Figure 2.2). In order to evaluate the efficiency of our algorithm we used Eq. (5.14). Then, to test the condition that the Hessian at the point $\boldsymbol{\theta}^* + \boldsymbol{\epsilon}$ is positive semidefinite, we prepared the state $|\psi(\boldsymbol{\theta}^* + \boldsymbol{\epsilon})\rangle$ and calculated the Hessian using Eq. (2.5). We chose to evaluate our method in MaxCut and Number Partitioning (see Sec. 2.3) and on the Transverse-Field Ising Chain model (see Sec. 2.4).

5.5.2 Results

5.5.2.1 Classical Optimization Problems

The two methods (VQE and AQC-PQC) were tested on the MaxCut and the Number Partitioning problems (details can be found in Sec. 2.3). For these problems, we chose to compare the methods on instance classes that we consider hard. Both of these problems have an intrinsic $\mathbb{Z}_2$ symmetry, and so we chose instances with only two optimal solutions (one solution can be acquired from the other by flipping all qubits). Specifically, for the MaxCut problem, we sampled 100 *random weighted graphs* of sizes 8 to 12, while for the Number Partitioning problem, we sampled 100 instances of the same size as MaxCut with integers drawn from the interval $[0, 50]$. The results are illustrated in Figure 5.3. Additionally, we can see the overlap returned by both algorithms in Tables 5.1,5.2.

Overall, we can see that AQC-PQC is able to outperform VQE in all instances, achieving overlap even five times larger in MaxCut (see Table 5.1) and ten times larger in Number Partitioning (see Table 5.2). Moreover, as seen in Figure 5.3, the output states returned by AQC-PQC are significantly closer (in terms of energy) to the ground state compared to VQE. This is to be expected as the non-convexity of the cost landscape results in the classical optimization part of VQE being stuck in a local minimum. As pointed out in [37, 38], the cost landscapes in shallow-depth VQAs, such as those utilized in this chapter, are filled with a vast amount of local minima, which makes them untrainable. One potential way out of this is to overparametrize the ansatz family [24], provided that the ansatz family can be overparametrized with a polynomial number of parameters. However, for NISQ devices, the large depth would result in noisy calculations due to the large number of errors and low coherence times.

On the other hand, AQC-PQC provides a more robust strategy to navigate the

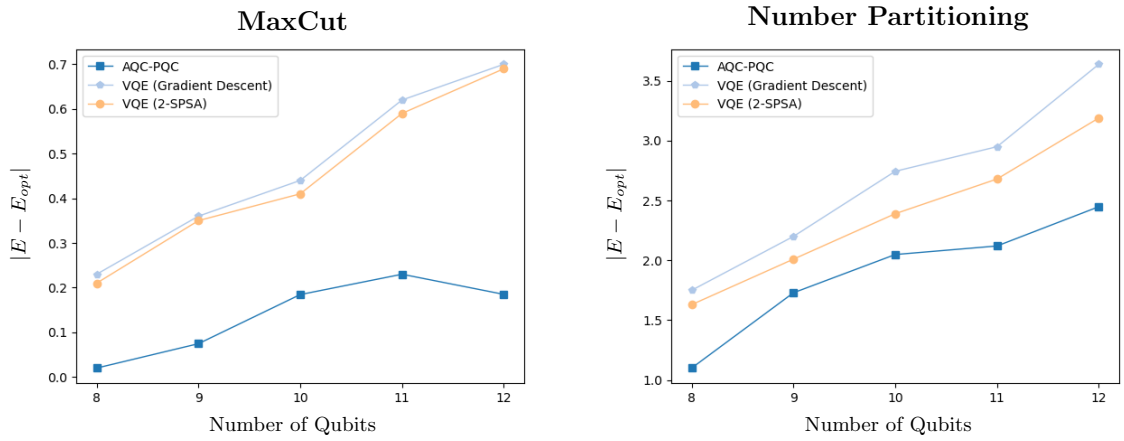

Figure 5.3: Performance of AQC-PQC algorithm compared to VQE (with 2-SPSA and Gradient Descent optimizers) for the MaxCut problem (left) and the Number Partitioning problem (right). The AQC-PQC algorithm with the dark blue line (square markers) is able to outperform both 2-SPSA and Gradient Descent in terms of the quality of the output solution.

MaxCut	Optimal Solution Overlap (%)					
	7 Qubits	**8 Qubits**	**9 Qubits**	**10 Qubits**	**11 Qubits**	**12 Qubits**
AQC-PQC	82.7	74.3	93.1	50	28.1	56.6
VQE	62.3	54.7	60.8	39.2	22.1	11.1

Table 5.1: Probability of sampling the optimal solution for the MaxCut problem for instances of size 7-12. AQC-PQC was able to outperform VQE on all instance sizes, achieving a larger overlap with the optimal solution.

(time-evolving) landscape. Provided that the number of steps is chosen accordingly and the ansatz family is expressive enough, the latter algorithm will always achieve a large overlap with the optimal solution.

However, it is important to stress that the expressiveness of the ansatz plays a significant role in the performance of the algorithm. We have observed that within the limits of very large steps, if the ansatz family is not sufficiently expressive, AQC-PQC will converge into a suboptimal solution. The reason is that for the intermediate ground states, circuits of large depth are required in order to remain close to the instantaneous ground state. In Sec. 5.6, we provide a case study in which we investigate the performance of AQC-PQC for different ansatz families.

Number Partitioning	Optimal Solution Overlap (%)					
	7 Qubits	**8 Qubits**	**9 Qubits**	**10 Qubits**	**11 Qubits**	**12 Qubits**
AQC-PQC	37.5	21.9	24.7	12.6	5	4.6
VQE	28.5	6.2	6.4	1.2	0.8	0.4

Table 5.2: Probability of sampling the optimal solution for the Number Partitioning problem for instances of size 7-12. AQC-PQC was able to achieve a significantly larger overlap than VQE for all instance sizes.

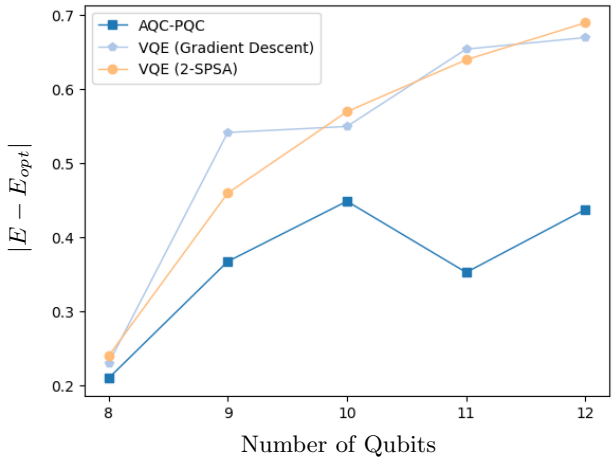

Figure 5.4: Performance of AQC-PQC algorithm compared to VQE (with 2-SPSA and Gradient Descent optimizers) for the Transverse-Field Ising Chain model.

5.5.2.2 Transverse-Field Ising Chain

Details for the TFI Chain problem can be found in Sec. 2.4. We compared the two methods on instances of sizes 8 to 12 qubits. We evaluated the performance of the two algorithms on the TFI Chain for 100 random instances with the couplings (J_k, h) drawn uniformly at random from the uniform distribution. The results of the TFI Chain model are illustrated in Figure 5.4. Overall, we observe that AQC-PQC can return approximations of the ground state of H_{TFI} that are closer compared to those returned by VQE.

5.6 Ansatz Expressiveness

It is very important to understand how crucial it is to have a parameterized family of gates that is sufficiently expressive. To be exact, the ansatz family should be expressive enough so that in the vicinity of small energy gaps, the energy returned by AQC-PQC

is close to the ground state. In Figure 5.5 we can visualize how close are the ground state energies returned by AQC-PQC at every step of the algorithm for different ansatz families.

Specifically, Figure 5.5 illustrates a 3-regular graph of size 6 for the MaxCut problem. The system was initialized at the ground state of $H_0 = -\sum_i \sigma_i^x$ (with ground state $|+\rangle^{\otimes 6}$) and the Hamiltonian was discretized into 30 steps:

$$H_k = \left(1 - \frac{k}{30}\right) H_0 + \frac{k}{30} H_{\text{MC}} \tag{5.29}$$

where $k \in [30]$ and H_{MC} is the MaxCut Hamiltonian defined in Eq. (2.17). As a parameterized family of gates, we used the ansatz on the left of Figure 2.2 with 0 (no-entanglement gates), 1,2, and 3 layers of entanglement gates. Despite the fact that all these parameterized families contain both the initial and final ground states, they differ in the reachability of the intermediate ground states. We can see in Figure 5.5 how the most expressive ansatz family (of 3 layers) is able to always remain close to the true ground state energy compared to the ansatz family, which uses no entanglement. Although the latter achieves a non-zero overlap with the optimal solution, the final output energy is far from the optimal. As a result, for harder instances where the returned energy is larger than the first excited energy, there is a high probability that the resulting overlap will tend to zero.

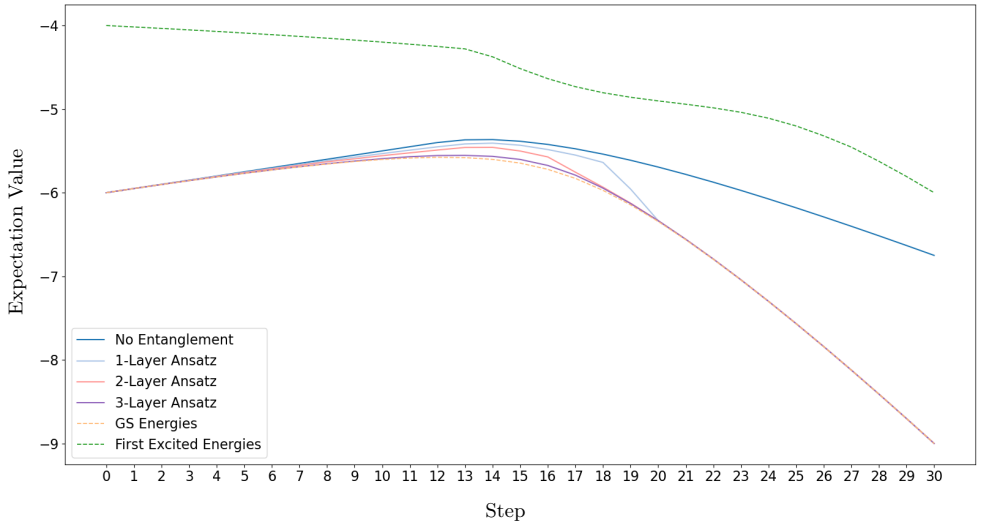

Figure 5.5: AQC-PQC performance for three different parameterized family of gates for the MaxCut problem. The 3-Layer Ansatz (purple line) always remains close to the true ground state energy, returning a quantum state with an overlap equal to one at the end of the algorithm. On the other hand, the ansatz family that uses no entanglement (and thus is the least expressive) cannot remain close to the ground state (blue line), and so at the final step, it returns a solution that is far from the optimal.

Chapter 6

Conclusion and future directions

In this thesis, we aimed to advance the practicality of hybrid quantum/classical algorithms. Our goal was to solve the three main bottlenecks that prohibit the usefulness of variational quantum algorithms. As we previously discussed these are:

1. Distribution of quantum and classical resources.
2. Local minima and bad initialization.
3. Expressivity and barren plateaux.

The narrative we have presented was composed of three main phases. First, in Chapter 3, we aimed to solve the first bottleneck. We showed how information-theoretic methods outperform naive optimization methods in Variational Quantum Algorithms (VQAs) [11]. We introduced a novel optimization algorithm, called *random natural gradient* where we showed that it is quadratically faster (in terms of quantum state preparations) than the well-established (and well-studied) quantum natural gradient [30]; a method that uses the computationally expensive quantum Fisher information matrix (QFIM) and is related to the quantum imaginary-time evolution. Our proposed method is able to adapt the information of the local geometry of the parameterized quantum states in the optimization process and overcome existing barriers of VQAs. To achieve this, the parameterized quantum state is first rotated at random by a random unitary and then measured in the computational basis. Then, the measurement outcomes are used to calculate the (random) classical Fisher information matrix, which is used as a precondition in the optimization process. In the same chapter, we provided two estimators of the QFIM. The first requires

a collection of random measurements, where the random measurement is performed by first applying a random unitary that is sampled from a 2-design distribution and then measured on the computational basis while the latter is constructed as an average classical Fisher information matrix. Furthermore, we show how the second estimator is connected to RNG. Finally, we introduce a stochastic-coordinate approximation of the QFIM where a reduced QFIM is constructed by randomly sampling a set of parameters in the quantum circuits.

This part of the work opens up a number of different ways for future research. At first, it would be fruitful to develop strategies to identify good subsets of unitaries $\nu \in U(2^n)$ (with small depth requirements) so that sampling from these unitaries would result in CFIMs close to the QFIM. This opens up a new research direction, as the way of choosing the appropriate basis may also be viewed as a quantum machine learning problem. On top of that, it is necessary to research and test the practicality limit of already known quantum algorithms when certain computationally expensive quantities (such as the QFIM) are estimated (up to some error) by computationally cheaper objects. Techniques such as our proposed algorithm showcase their practicality as quantum-inspired classical approximation methods, on top of the advantages that were previously mentioned. A user can classically prepare and store ansatz states, and by performing K measurements on the stored state, they can approximate the QFIM and, eventually, identify the direction that follows imaginary-time evolution. Additionally, another interesting research direction is to quantify the sample complexity K of the two estimators. Different choices for K may be needed to achieve a desired accuracy ϵ (error from the QFIM) for different parameterized quantum states. Thus, it is essential to understand how the geometry of the underlying quantum states is connected to the sampling requirements of our algorithm. Finally, a very interesting research direction is to examine connections between the random natural gradient and the BFGS algorithm [159].

Next, in Chapter 4, we aimed to solve the second bottleneck. We showed how developing different objective functions that depend on the problem class leads to significantly improved performance, overcoming one of the fundamental bottlenecks of VQAs. As we showed, for classical combinatorial optimization problems where the optimal solution is a computational basis state, using an averaged and increasing α-tail of the energy outcomes avoids local minima and is able to return solutions with very large (and close to unity) overlaps with the optimal solution, overcoming barriers of the previously introduced objective function of [61]; a method that we named *ascending-*

CVaR. Our idea was to use an objective function that dynamically changes during the optimization process. Specifically, the number of samples that the optimizer sees during the optimization increases with time, essentially becoming the expectation value at the end of the process. This, intuitively, avoids getting stuck at local minima since the energy landscape for different α's differs, apart from the global minimum.

This work not only offers a generic method to improve the performance of VQAs for combinatorial optimization problems, but it also suggests a new direction of research where dynamic objective functions can be used to boost the performance in terms of quality and speed of near-term quantum algorithms. An immediate follow-up to the proposal suggested here is to generalize our approach. Concretely, our method introduces two extra degrees of freedom. The hyperparameter λ and the function according to which the parameter α increases. It is worth exploring a more systematic rule on how to fix these degrees of freedom according to the problem considered and the features of the specific instance. Considering other dynamic objective functions is another direction that is worth pursuing.

Finally, in Chapter 5, we aimed to solve the third bottleneck. We introduced a novel hybrid algorithm that hops between instantaneous ground states by iteratively solving a constrained linear system of equations. On top of that, our proposal does not initialize the parameterized quantum architecture on a random configuration (with the problems inherited on that, such as flat landscapes and local minima near the initialization point) but rather on the configuration that generates a ground state of a known Hamiltonian. Our algorithm draws a connection to a well-studied quantum computing framework, namely *adiabatic quantum computing* (AQC) [141]. We show that by slowly varying the Hamiltonian from an initial Hamiltonian H_0 with a known ground state to a target Hamiltonian (corresponding to the unknown target ground state), we are able to hop between ground states and by solving this constrained linear system to reach the ground state of interest.

Our final work also opens up at least three distinct future directions. First of all, the time required for AQC is related to the number of steps K. While we provided evidence for the practicality of our method, an important task is to analyze K extensively (both theoretically and using extensive numerical simulations). There are cases (ansatz families) where the first excited state may not necessarily correspond to a minimum and so the algorithm can acquire a speedup. Secondly, testing our method for different problems

and much larger instances is the other obvious direction. Thirdly, identifying the optimal algorithm to solve the inherited constrained linear system problem in our approach is the final research direction. Optimizing in terms of both 1) number of state preparations and 2) calls in the classical computer is a necessary step to make the approach practical.

We hope that the work presented in this thesis has made it even more apparent that the study and improvement of hybrid quantum/classical algorithms can lead to this technology moving into the practical regime, where it can offer a speedup over its classical counterparts.

Appendix A

A.1 Derivation of Classical Fisher Information Matrix

We start with the KL-divergence for two probability distributions $\boldsymbol{p}(\boldsymbol{\theta})$, $\boldsymbol{p}(\boldsymbol{\theta}+\boldsymbol{\epsilon})$:

$$\begin{aligned}\mathrm{KL}(\boldsymbol{p}(\boldsymbol{\theta})||\boldsymbol{p}(\boldsymbol{\theta}+\boldsymbol{\epsilon})) = \sum_l p_l(\boldsymbol{\theta}) \log \frac{p_l(\boldsymbol{\theta})}{p_l(\boldsymbol{\theta}+\boldsymbol{\epsilon})} = \\ \sum_l p_l(\boldsymbol{\theta}) \log p_l(\boldsymbol{\theta}) - \sum_l p_l(\boldsymbol{\theta}) \log p_l(\boldsymbol{\theta}+\boldsymbol{\epsilon})\end{aligned} \tag{A.1}$$

The elements of the CFIM are defined as the second-order derivatives of the KL-divergence. Specifically, an element $[\mathcal{F}_C]_{ij}$ is given by:

$$\begin{aligned}[\mathcal{F}_C]_{ij} = -\frac{\partial^2}{\partial\epsilon_i\partial\epsilon_j}\sum_l p_l(\boldsymbol{\theta}) \log p_l(\boldsymbol{\theta}+\boldsymbol{\epsilon})\bigg|_{\boldsymbol{\epsilon}=0} = -\sum_l p_l(\boldsymbol{\theta}) \frac{\partial^2}{\partial\epsilon_i\partial\epsilon_j} \log p_l(\boldsymbol{\theta}+\boldsymbol{\epsilon})\bigg|_{\boldsymbol{\epsilon}=0} = \\ -\mathbb{E}\left\{\frac{\partial^2}{\partial\epsilon_i\partial\epsilon_j} \log p_l(\boldsymbol{\theta}+\boldsymbol{\epsilon})\bigg|_{\boldsymbol{\epsilon}=0}\right\} = -\mathbb{E}\left\{\frac{\partial^2}{\partial\theta_i\partial\theta_j} \log p_l(\boldsymbol{\theta})\right\}\end{aligned} \tag{A.2}$$

Since:

$$-\frac{\partial^2}{\partial\theta_i\partial\theta_j} \log p_l(\boldsymbol{\theta}) = -\frac{1}{p_l(\boldsymbol{\theta})}\frac{\partial^2 p_l(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} + \frac{1}{p_l^2(\boldsymbol{\theta})}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_j} \tag{A.3}$$

Substituting Eq. (A.3) in Eq. (A.2) we get:

$$\begin{aligned}[\mathcal{F}_C]_{ij} = \mathbb{E}\left\{-\frac{1}{p_l(\boldsymbol{\theta})}\frac{\partial^2 p_l(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} + \frac{1}{p_l^2(\boldsymbol{\theta})}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_j}\right\} = \\ \sum_l p_l(\boldsymbol{\theta})\left[\frac{-1}{p_l(\boldsymbol{\theta})}\frac{\partial^2 p_l(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} + \frac{1}{p_l^2(\boldsymbol{\theta})}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_j}\right] \\ \implies [\mathcal{F}_C]_{ij} = \sum_l \frac{1}{p_l(\boldsymbol{\theta})}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_l(\boldsymbol{\theta})}{\partial\theta_j}\end{aligned} \tag{A.4}$$

where we used the fact that

$$\sum_l \frac{\partial^2 p_l(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} = \frac{\partial^2}{\partial\theta_i\partial\theta_j}\sum_l p_l(\boldsymbol{\theta}) = 0 \tag{A.5}$$

As you can see, although the Fisher information is the matrix corresponding to second-order derivatives, after our analysis, it can be written as a product of first-order derivatives. This has an immediate advantage in the number of resources needed. It requires only the calculation of the first-order derivatives in the quantum computer. Then, the classical computer post-processes the m first-order derivatives and stores the classical fisher information using $\mathcal{O}_C(m^2)$ classical memory.

A.2 Additional Experiments

In this section, we provide some extra experiments, comparing our two proposed methods (see *Random Natural Gradient* in Sec. 3.2 and *Stochastic-Coordinate Quantum Natural Gradient* in Sec. 3.6) on the MaxCut problem (see Sec. 3.7). For our experiments, we employed the PQC visualized on the right side of Figure 2.2 with 3 layers. We sampled random 8 and 10-qubit unweighted 3-regular graph instances.

A.3 Random Measurements and QFIM

Randomized measurements constitute a powerful tool that has been exploited for several different applications throughout the quantum computing literature [54, 55, 59, 160, 161]. We begin by recalling a few standard notions on random operators acting on our space of parameterized qubits, referring to [134, 162] for a comprehensive introduction.

Definition A.1. (Haar Measure) [134]. The Haar measure on the unitary group $U(d)$ is the unique probability measure μ_H [163] that is both *left* and *right* invariant over the group $U(d)$, i.e. for all integrable function $f : U(d) \to \mathcal{L}(\mathbb{C}^d)$ and for all $V \in U(d)$ we have:

$$\int_{U(d)} f(U)d\mu_H(U) = \int_{U(d)} f(UV)d\mu_H(U) = \int_{U(d)} f(VU)d\mu_H(U) \tag{A.6}$$

In this thesis, we will denote the integral of a function $f(U)$ over the Haar measure as the expected value of $f(U)$ with respect to the probability measure μ_H, denoted as $\mathbb{E}_{U\sim\mu_H}[f(U)]$:

$$\mathbb{E}_{U\sim\mu_H}[f(U)] := \int_{U(d)} f(U)d\mu_H(U) \tag{A.7}$$

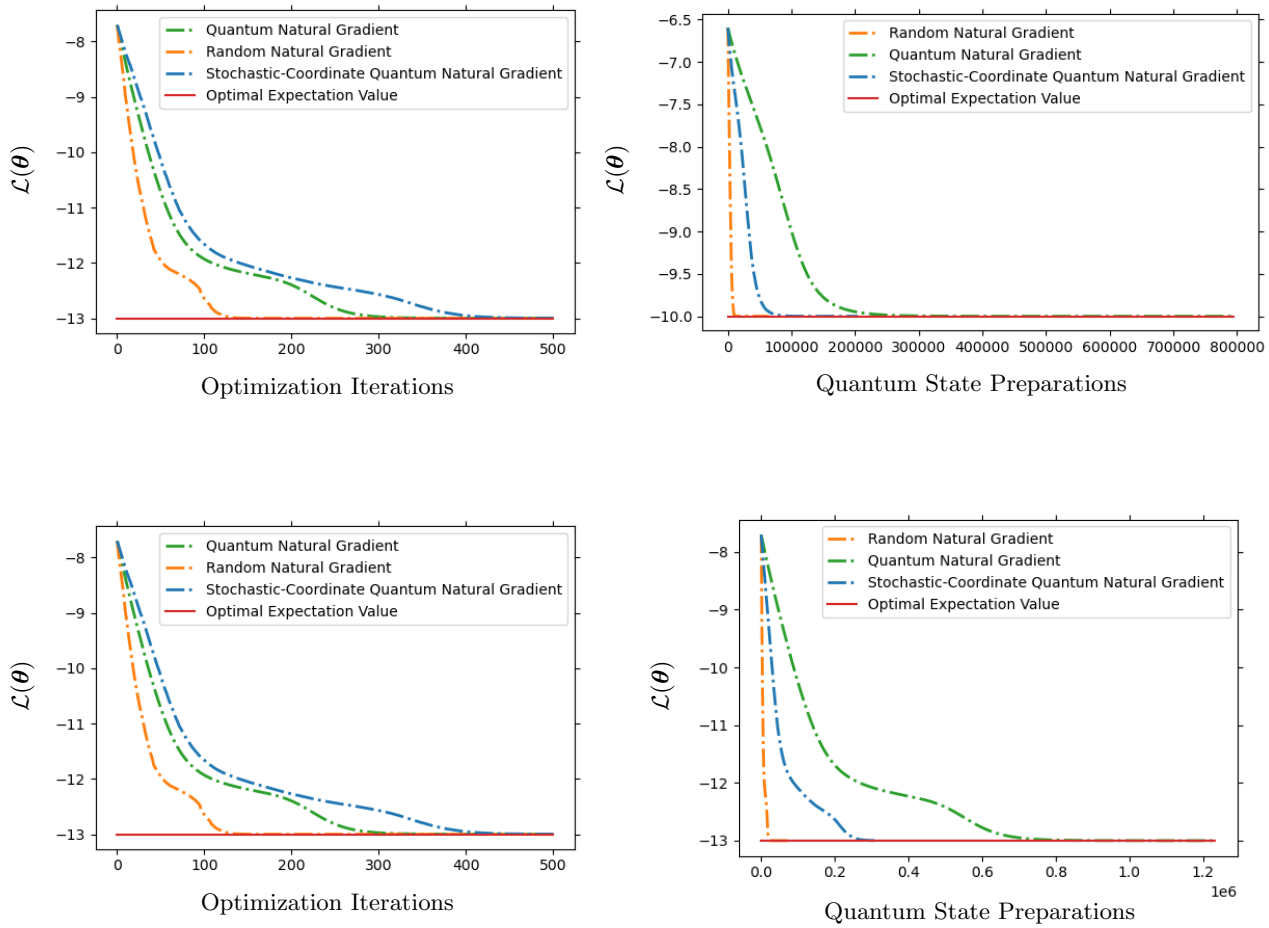

Figure A.1: Comparison of QNG, RNG and SC-QNG on MaxCut instances corresponding to 3-regular graphs of 8 qubits (up left and up right figures) and 10 qubits (bottom left and bottom right).

A quantity that will play a very important role in our analysis is the k-moment operator, with $k \in \mathbb{N}$ (or else the k-fold twirl).

Definition A.2. (k-moment operator). The k-moment operator, with respect to the probability measure μ_H, is defined as $\mathcal{M}_k : \mathcal{L}((\mathbb{C}^d)^{\otimes k}) \to \mathcal{L}((\mathbb{C}^d)^{\otimes k})$:

$$\mathcal{M}^{(k)}_{\mu_H}(O) := \mathbb{E}_{U\sim\mu_H}[U^{\otimes k} O U^{\dagger\otimes k}] \tag{A.8}$$

for all operators $O \in \mathcal{L}((\mathbb{C}^d)^{\otimes k})$.

As it turns out, there are tools that will allow us to calculate the k-moment operators. Specifically, the moment operator defined in Definition A.2 is the orthogonal projector onto the commutant $Comm(U(d), k)$. The commutant is defined below.

Definition A.3. (Commutant). Given $S \subseteq \mathcal{L}(\mathbb{C}^d)$, we define its k-order commutant as:

$$Comm(S,k) := \{A \in \mathcal{L}((\mathbb{C}^d)^{\otimes k}) : [A, B^{\otimes k}] = 0 \ \ \forall B \in S\} \tag{A.9}$$

As it can be easily seen, a set of operators that commute with every unitary $U^{\otimes k}$ are the permutation operators. These are defined as:

Definition A.4. (Permutation operators). Given $\pi \in S_k$ an element of the symmetric group S_k, we define the permutation matrix $V_d(\pi)$ to be the unitary matrix that satisfies:

$$V_d(\pi) |\psi_1\rangle \otimes \ldots \otimes |\psi_k\rangle = \left|\psi_{\pi^{-1}(1)}\right\rangle \otimes \ldots \otimes \left|\psi_{\pi^{-1}(k)}\right\rangle \tag{A.10}$$

for all $|\psi_1\rangle, \ldots, |\psi_k\rangle \in \mathbb{C}^d$

A well-celebrated result is the Schur-Weyl duality [162], that states that the image of the k-moment operator is spanned by the permutation operators. As such, we can calculate the first and second moments of an operator $O \in \mathcal{L}(\mathbb{C}^d)$ as:

$$\mathbb{E}_{U\sim\mu_H}[UOU^\dagger] = \frac{\mathrm{Tr}(O)}{d} I \tag{A.11}$$

$$\mathbb{E}_{U\sim\mu_H}[U^{\otimes 2}OU^{\dagger\otimes 2}] = \frac{\mathrm{Tr}(O) - d^{-1}\,\mathrm{Tr}(\mathbb{S}O)}{d^2-1}\mathbb{I} + \frac{\mathrm{Tr}(\mathbb{S}O) - d^{-1}\,\mathrm{Tr}(O)}{d^2-1}\mathbb{S} \tag{A.12}$$

where $I, \mathbb{I}$ correspond to the identity operators on $\mathbb{C}^d$ and $(\mathbb{C}^d)^{\otimes 2}$ respectively and $\mathbb{S}$ is the SWAP operator defined as:

$$\mathbb{S}(|\psi_1\rangle \otimes |\psi_2\rangle) = |\psi_2\rangle \otimes |\psi_1\rangle \tag{A.13}$$

Our starting point originates from [55, 161], where the authors showed that the fidelity between two quantum states ρ_1, ρ_2 can be calculated using the following Theorem.

Theorem A.1. *(Fidelity of two quantum states [55]) Consider two quantum states ρ_1 and ρ_2 on n qubits in Hilbert space $\mathcal{H}$ of dimension $\mathcal{D} = 2^n$. For global random unitaries U, the overlap between the quantum states is given by:*

$$\mathrm{Tr}[\rho_1\rho_2] = 2^n \sum_{\boldsymbol{s},\boldsymbol{s'}} (-2^n)^{-D_G[\boldsymbol{s},\boldsymbol{s'}]} \langle \boldsymbol{s}| \langle \boldsymbol{s'}| \mathcal{M}^{(2)}_{\mu_H}(\rho_1 \otimes \rho_2) |\boldsymbol{s}\rangle |\boldsymbol{s'}\rangle \tag{A.14}$$

where the global Hamming distance D_G is defined as:

$$D_G[\boldsymbol{s}, \boldsymbol{s'}] = \begin{cases} 0 \text{ if } \boldsymbol{s} = \boldsymbol{s'} \\ 1 \text{ if } \boldsymbol{s} \neq \boldsymbol{s'} \end{cases} \tag{A.15}$$

and $\mathcal{M}^{(k)}_{\mu_H}(\cdot) := \mathbb{E}_{U\sim\mu_H}[U^{\otimes k}(\cdot)U^{\dagger\otimes k}]$ is the k-moment operator.

In our case, we work with *parameterized quantum states* consisting of a total of m parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$. Specifically, let $\rho_1 := \rho(\boldsymbol{\theta})$ and $\rho_2 := \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})$ and let also ρ_1, ρ_2 be *pure*. As such, ρ_1 and ρ_2 can be written as:

$$\begin{aligned} \rho_1 = \rho(\boldsymbol{\theta}) = |\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})| \\ \rho_2 = \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon}) = |\psi(\boldsymbol{\theta} + \boldsymbol{\epsilon})\rangle\langle\psi(\boldsymbol{\theta} + \boldsymbol{\epsilon})| \end{aligned} \tag{A.16}$$

We can express the elements of the moment operator in the computation basis as:

$$\begin{aligned} \langle \boldsymbol{s}, \boldsymbol{s'}| \mathcal{M}^{(2)}_{\mu_H}(\rho(\boldsymbol{\theta}) \otimes \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon}))\, |\boldsymbol{s}, \boldsymbol{s'}\rangle = \int_{U(d)} d\mu_H(U)\, \langle \boldsymbol{s}|\, \langle \boldsymbol{s'}|\, U^{\otimes 2} \rho(\boldsymbol{\theta}) \otimes \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon}) U^{\dagger \otimes 2}\, |\boldsymbol{s}\rangle\, |\boldsymbol{s'}\rangle \\ \int_{U(d)} d\mu_H(U)\, \langle \boldsymbol{s}|\, U\rho(\boldsymbol{\theta})U^\dagger\, |\boldsymbol{s}\rangle\, \langle \boldsymbol{s'}|\, U\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})U^\dagger\, |\boldsymbol{s'}\rangle = \int_{U(d)} d\mu_H(U)\, \mathrm{Tr}\big[\rho(\boldsymbol{\theta})U^\dagger \Pi_{\boldsymbol{s}} U\big]\, \mathrm{Tr}\big[\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})U^\dagger \Pi_{\boldsymbol{s'}} U\big] \\ = \int_{U(d)} d\mu_H(U)\, \mathrm{Tr}\big[\rho(\boldsymbol{\theta}) \otimes \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon}) U^{\dagger \otimes 2} (\Pi_{\boldsymbol{s}} \otimes \Pi_{\boldsymbol{s'}}) U^{\otimes 2}\big] = \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] \end{aligned} \tag{A.17}$$

where $p^U_{\boldsymbol{s}} = \mathrm{Tr}\big[\rho(\boldsymbol{\theta})U^\dagger \Pi_{\boldsymbol{s}} U\big]$. As such we can rewrite Eq. (A.14) as:

$$\begin{aligned} \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})] &= 2^n \sum_{\boldsymbol{s}, \boldsymbol{s'}} (-2^n)^{-D_G[\boldsymbol{s}, \boldsymbol{s'}]}\, \mathrm{Tr}\Big[\rho(\boldsymbol{\theta}) \otimes \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \mathcal{M}^{(2)}_{\mu_H}(\Pi_{\boldsymbol{s}} \otimes \Pi_{\boldsymbol{s'}})\Big] \\ &= 2^n \sum_{\boldsymbol{s}, \boldsymbol{s'}} (-2^n)^{-D_G[\boldsymbol{s}, \boldsymbol{s'}]} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] \end{aligned} \tag{A.18}$$

By expanding the sum, the fidelity between the two states can be written as:

$$\begin{aligned} \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})] = 2^n \sum_{\boldsymbol{s}, \boldsymbol{s'}} (-2^n)^{-D_G[\boldsymbol{s}, \boldsymbol{s'}]} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] \\ = 2^n \sum_{\boldsymbol{s}} (-2^n)^{-D_G[\boldsymbol{s}, \boldsymbol{s}]} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] + 2^n \sum_{\substack{\boldsymbol{s}, \boldsymbol{s'} \\ \boldsymbol{s} \neq \boldsymbol{s'}}} (-2^n)^{-D_G[\boldsymbol{s}, \boldsymbol{s'}]} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] \\ = 2^n \sum_{\boldsymbol{s}} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] - \sum_{\substack{\boldsymbol{s}, \boldsymbol{s'} \\ \boldsymbol{s} \neq \boldsymbol{s'}}} \mathbb{E}_{U \sim \mu_H}\big[p^U_{\boldsymbol{s}}(\boldsymbol{\theta}) p^U_{\boldsymbol{s'}}(\boldsymbol{\theta} + \boldsymbol{\epsilon})\big] \end{aligned} \tag{A.19}$$

where we used Eq. (A.15). Consider now the infidelity between two quantum states as defined as:

$$d_F\Big(\rho(\boldsymbol{\theta}), \rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})\Big) = 1 - \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})] \tag{A.20}$$

If we Taylor expand the infidelity around $\boldsymbol{\epsilon} = \mathbf{0}$, then we have:

$$\begin{aligned} 1 - \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})] = 1 - \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta})] - \sum_{i=1}^{m} \frac{\partial}{\partial \epsilon_i} \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})]\Big|_{\boldsymbol{\epsilon}=0} \epsilon_i \\ -\frac{1}{2} \sum_{i,j=1}^{m} \frac{\partial}{\partial \epsilon_i \partial \epsilon_j} \mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta} + \boldsymbol{\epsilon})]\Big|_{\boldsymbol{\epsilon}=0} \epsilon_i \epsilon_j + \mathcal{O}(\|\boldsymbol{\epsilon}\|_1^3) \end{aligned}$$

where the partial derivatives at $\boldsymbol{\epsilon} = \mathbf{0}$ are zero, since $\mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta}+\boldsymbol{\epsilon})]$ is maximized at $\boldsymbol{\epsilon} = \mathbf{0}$. As such, the infidelity, can be expressed as:

$$d_F\Big(\rho(\boldsymbol{\theta}), \rho(\boldsymbol{\theta}+\boldsymbol{\epsilon})\Big) = \frac{1}{4}\boldsymbol{\epsilon}^T \mathcal{F}_Q(\boldsymbol{\theta})\boldsymbol{\epsilon} + \mathcal{O}(\|\boldsymbol{\epsilon}\|_1^3) \tag{A.21}$$

where for small $\boldsymbol{\epsilon}$, the higher-order terms can be neglected. The matrix $\mathcal{F}_Q(\boldsymbol{\theta})$ is the *quantum Fisher information matrix*, defined as:

$$\mathcal{F}_Q(\boldsymbol{\theta}) = -2\boldsymbol{\nabla}^2 \,\mathrm{Tr}[\rho(\boldsymbol{\theta})\rho(\boldsymbol{\theta}+\boldsymbol{\epsilon})]\Big|_{\boldsymbol{\epsilon}=0} \tag{A.22}$$

Thus, a matrix element $[F_Q(\boldsymbol{\theta})]_{ij}$ can be written as:

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = -2^{n+1}\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}\Big|_{\boldsymbol{\epsilon}=0}\right] + 2\sum_{\substack{\boldsymbol{s},\boldsymbol{s}'\\ \boldsymbol{s}\neq\boldsymbol{s}'}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}\Big|_{\boldsymbol{\epsilon}=0}\right] \tag{A.23}$$

If we focus at the second term of Eq (A.23), we notice that:

$$\begin{aligned}
\sum_{\substack{\boldsymbol{s},\boldsymbol{s}'\\ \boldsymbol{s}\neq\boldsymbol{s}'}} p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j} &= \sum_{\boldsymbol{s}} p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\sum_{\boldsymbol{s}'\neq\boldsymbol{s}}\frac{\partial^2 p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}\\
\sum_{\boldsymbol{s}} p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\left(\sum_{\boldsymbol{s}'}\frac{\partial^2 p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j} - \frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}\right) &= -\sum_{\boldsymbol{s}} p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}
\end{aligned} \tag{A.24}$$

where we used the fact that:

$$\sum_{\boldsymbol{s}'}\frac{\partial^2 p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j} = \frac{\partial^2}{\partial\epsilon_i\partial\epsilon_j}\sum_{\boldsymbol{s}'} p_{\boldsymbol{s}'}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon}) = 0 \tag{A.25}$$

Thus, the QFIM elements can be expressed as:

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = -(2^{n+1}+2)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})}{\partial\epsilon_i\partial\epsilon_j}\Big|_{\boldsymbol{\epsilon}=0}\right] = -(2^{n+1}+2)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta})\frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right] \tag{A.26}$$

The calcuation of the quantum Fisher information using Eq. (3.8) is impractical. The reason is that it requires the calculation of the Hessian of the outcome probabilities, which in general requires $O(m^2)$ quantum states to estimate it. However, we can calculate

that:

$$\begin{aligned}\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right] &= \sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}U^\dagger\Pi_{\boldsymbol{s}}U\right]\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}U^\dagger\Pi_{\boldsymbol{s}}U\right]\right]\\ &= \sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\otimes\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}U^{\dagger\otimes 2}\Pi_{\boldsymbol{s}}^{\otimes 2}U^{\otimes 2}\right]\right]\\ &= \frac{1}{2^{2n}-1}\sum_{\boldsymbol{s}}\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\otimes\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\left[\left(1-\frac{1}{2^n}\right)\mathbb{I}+\left(1-\frac{1}{2^n}\right)\mathbb{S}\right]\right]\\ &= \frac{2^n-1}{2^{3n}-2^n}\sum_{\boldsymbol{s}}\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\right]\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]+\frac{2^n-1}{2^{3n}-2^n}\sum_{\boldsymbol{s}}\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]\\ &= \frac{2^n-1}{2^{3n}-2^n}\sum_{\boldsymbol{s}}\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]=\frac{1}{2^n+1}\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]\end{aligned} \tag{A.27}$$

where in the second line we used Eq. (A.12) and we also used the fact that $\operatorname{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\right] = \frac{\partial}{\partial\theta_i}\operatorname{Tr}[\rho(\boldsymbol{\theta})] = 0$ and $\operatorname{Tr}[A\otimes B\mathbb{S}] = \operatorname{Tr}[AB]$. Now, we can use the fact that:

$$\frac{\partial^2\rho^2(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} = 2\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}+2\rho(\boldsymbol{\theta})\frac{\partial^2\rho(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} \tag{A.28}$$

Thus, substituting the above equation in Eq. (A.27) we get:

$$\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right] = \frac{1}{2(2^n+1)}\operatorname{Tr}\left[\frac{\partial^2\rho^2(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right]-\frac{1}{(2^n+1)}\operatorname{Tr}\left[\rho(\boldsymbol{\theta})\frac{\partial^2\rho(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right] = \frac{[\mathcal{F}_Q]_{ij}}{2(2^n+1)} \tag{A.29}$$

where we used the fact that $\operatorname{Tr}\left[\frac{\partial^2\rho(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right] = \frac{\partial^2}{\partial\theta_i\partial\theta_j}\operatorname{Tr}[\rho(\boldsymbol{\theta})] = 0$. As such, we were able to prove that the matrix elements of the quantum Fisher information matrix can be written as product of first-order derivatives:

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = 2(2^n+1)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right] \tag{A.30}$$

As a result, we proved that the quantum Fisher information matrix can be approximated as the average (over the Haar distribution) of a quantity that requires $\mathcal{O}(m)$ quantum states and $O(m^2)$ classical memory to store the matrix. As we see in our numerical experiments in Fig. 3.3 we can achieve a very good approximation of the quantum Fisher information with a small number of repetitions (usually much less than m repetitions).

Definition A.5. *(Unitary k-design) Let ν be a probability distribution defined over a set of unitaries $S \subseteq U(d)$. The distribution ν is unitary k-design if and only if:*

$$\mathbb{E}_{V\sim\nu}[V^{\otimes k}OV^{\dagger\otimes k}] = \mathbb{E}_{U\sim\mu_H}[U^{\otimes k}OU^{\dagger\otimes k}] \tag{A.31}$$

for all $O \in \mathcal{L}((\mathbb{C}^d)^{\otimes k})$.

In general, generating Haar random unitaries on a quantum computer is a computationally exhaustive task, since most unitary operators require a number of gates that scale exponentially to the number of qubits [134]. On the other hand, k-designs are distributions that match the Haar moments up to the k-order (see Definition 3.2). The advantage is that k-designs can be generated efficiently. As a result, we provide the following Corollary.

Corollary A.1. *If U is sampled from a 2-design ν, then the matrix elements of the quantum Fisher information matrix can be calculated as:*

$$[\mathcal{F}_Q(\boldsymbol{\theta})]_{ij} = 2(2^n+1)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\nu}\left[\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_j}\right] \tag{A.32}$$

Proof. We can express the quantity:

$$\mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_j}\right]$$

as:

$$\begin{aligned}\mathbb{E}_{U\sim\mu_H}\left[\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial p^U_{\boldsymbol{s}}(\boldsymbol{\theta})}{\partial\theta_j}\right] &= \mathbb{E}_{U\sim\mu_H}\left[\mathrm{Tr}\left[U\Pi_{\boldsymbol{s}}U^\dagger\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\right]\mathrm{Tr}\left[U\Pi_{\boldsymbol{s}}U^\dagger\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]\right] \\ &= \mathrm{Tr}\left[\mathbb{E}_{U\sim\mu_H}[U^{\otimes 2}\Pi_{\boldsymbol{s}}^{\otimes 2}U^{\dagger\otimes 2}]\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\otimes\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]\end{aligned} \tag{A.33}$$

Thus, using Definition 3.2 for the unitary k-designs (for $O = \Pi_s^{\otimes 2}$) we can conclude that if U comes from a 2-design then the proposition holds. □

As a result, the quantum Fisher information can be estimated by sampling unitaries that come from a k-design with $k \geq 2$ (since any k-design is also a 2-design if $k \geq 2$). An example of such ensembles is the n-qubit Clifford group $Cl(n)$ which forms a 3-design. The Clifford group is defined as:

$$Cl(n) := \{U \in U(2^n) |\ UPU^\dagger \in \mathcal{P}_n \text{ for all } P \in \mathcal{P}_n\} \tag{A.34}$$

where $\mathcal{P}_n$ is the Pauli group. Elements from the n-qubit Clifford group can be generated by a circuit with at most $\mathcal{O}(n^2/\log n)$ elementary gates [135].

At the same time, the classical Fisher information matrix (when the parameterized quantum state is rotated by a global random unitary U and then measured in the computational basis) can be expressed as:

$$[\mathcal{F}_C^U(\boldsymbol{\theta})]_{ij} = -\sum_{\boldsymbol{s}} p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) \frac{\partial^2 \ln[p_{\boldsymbol{s}}^U(\boldsymbol{\theta})]}{\partial\theta_i \partial\theta_j} = \sum_{\boldsymbol{s}} \frac{1}{p_{\boldsymbol{s}}^U(\boldsymbol{\theta})} \frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i} \frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j} \tag{A.35}$$

since:

$$-\frac{\partial^2}{\partial\theta_i\partial\theta_j} \ln p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) = -\frac{\partial}{\partial\theta_i}\left[\frac{1}{p_{\boldsymbol{s}}^U(\boldsymbol{\theta})} \frac{p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j}\right] = -\frac{1}{p_{\boldsymbol{s}}^U(\boldsymbol{\theta})} \frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} + \frac{1}{(p_{\boldsymbol{s}}^U(\boldsymbol{\theta})^2)} \frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_i} \frac{\partial p_{\boldsymbol{s}}^U(\boldsymbol{\theta})}{\partial\theta_j} \tag{A.36}$$

and we used again Eq. (A.25).

Conjecture: *The average classical Fisher information matrix, when the parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$ is rotated by a random unitary U and then measured in the computational basis approximates the quantum Fisher information matrix as:*

$$\mathbb{E}_{U\sim\mu_H}[\mathcal{F}_C^U(\boldsymbol{\theta})] = \frac{1}{2}\mathcal{F}_Q(\boldsymbol{\theta}) \tag{A.37}$$

One would have to show that:

$$\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) \frac{\partial^2 \ln[p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})]}{\partial\epsilon_i\partial\epsilon_j}\bigg|_{\boldsymbol{\epsilon}=0}\right] = -(2^n+1)\sum_{\boldsymbol{s}} \mathbb{E}_{U\sim\mu_H}\left[p_{\boldsymbol{s}}^U(\boldsymbol{\theta}) \frac{\partial^2 p_{\boldsymbol{s}}^U(\boldsymbol{\theta}+\boldsymbol{\epsilon})]}{\partial\epsilon_i\partial\epsilon_j}\bigg|_{\boldsymbol{\epsilon}=0}\right] \tag{A.38}$$

where $p_s^U(\boldsymbol{\theta}) = \mathrm{Tr}\big[\rho(\boldsymbol{\theta})U^\dagger \Pi_{\boldsymbol{s}} U\big]$. Proving the above conjecture is a very challenging task. The main reason is that it requires the calculation of a Haar integral over the unitary group $U(2^n)$ where the unitaries rise in a non-linear way. Equivalently, it cannot be written as a k-moment of an operator for which ways to calculate the integrals are known (e.g. see (A.12)). The proof of this conjecture is left for future work.

A.4 Proof of Lemma 3.2

Let the quantum Fisher information matrix $\mathcal{F}_Q$ and its estimator $\tilde{\mathcal{F}}_Q$ be non-singular with their eigenvalues satisfying:

$$\lambda_1(\mathcal{F}_Q) \geq \lambda_2(\mathcal{F}_Q) \geq \ldots \geq \lambda_m(\mathcal{F}_Q) > 0$$
$$\lambda_1(\tilde{\mathcal{F}}_Q) \geq \lambda_2(\tilde{\mathcal{F}}_Q) \geq \ldots \geq \lambda_m(\tilde{\mathcal{F}}_Q) > 0$$

Consider the two different linear systems in Eqs. (3.25), (3.56) with their corresponding solutions $\dot{\boldsymbol{\theta}}_Q$ and $\dot{\tilde{\boldsymbol{\theta}}}_Q$ respectively. We have that:

$$\begin{aligned}\left\|\dot{\boldsymbol{\theta}}_Q - \dot{\tilde{\boldsymbol{\theta}}}_Q\right\| &= \left\|(\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1})\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_\tau\right\| \\ \leq \left\|\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}\right\| \|\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_\tau\| &\leq \left\|\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}\right\| \|\mathcal{F}_Q\| \left\|\dot{\boldsymbol{\theta}}_Q\right\|\end{aligned}$$

where we used the fact that $\|\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_\tau\| \leq \|\mathcal{F}_Q\| \left\|\dot{\boldsymbol{\theta}}_Q\right\|$. As such, the relative error is upper bounded as:

$$\frac{\left\|\dot{\boldsymbol{\theta}}_Q - \dot{\tilde{\boldsymbol{\theta}}}_Q\right\|}{\left\|\dot{\boldsymbol{\theta}}_Q\right\|} \leq \left\|\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}\right\| \|\mathcal{F}_Q\| \tag{A.39}$$

If we consider the case where estimator differs from the quantum Fisher information matrix by a small matrix Δ (with $\|\Delta\| \leq \epsilon$), then we have:

$$\begin{aligned}\mathcal{F}_Q = \tilde{\mathcal{F}}_Q + \Delta \implies \\ \tilde{\mathcal{F}}_Q^{-1} = \mathcal{F}_Q^{-1} + \tilde{\mathcal{F}}_Q^{-1}(\mathcal{F}_Q - \tilde{\mathcal{F}}_Q)\mathcal{F}_Q^{-1}\end{aligned}$$

where:

$$\begin{aligned}\left\|\tilde{\mathcal{F}}_Q^{-1}(\mathcal{F}_Q - \tilde{\mathcal{F}}_Q)\tilde{\mathcal{F}}_Q^{-1}\right\| &\leq \left\|\tilde{\mathcal{F}}_Q^{-1}\right\| \left\|\mathcal{F}_Q - \tilde{\mathcal{F}}_Q\right\| \|\mathcal{F}_Q^{-1}\| \\ &\leq \frac{\epsilon}{\lambda_{\min}(\tilde{\mathcal{F}}_Q)\lambda_{\min}(\tilde{\mathcal{F}}_Q)} = \frac{\epsilon}{\lambda_m(\tilde{\mathcal{F}}_Q)\lambda_m(\mathcal{F}_Q)}\end{aligned}$$

If we assume that the matrix $\tilde{\mathcal{F}}_Q^{-1}(\mathcal{F}_Q - \tilde{\mathcal{F}}_Q)\mathcal{F}_Q^{-1}$ is also small then we can use the dual Weyl's inequality and have that:

$$\begin{aligned}\lambda_1(\tilde{\mathcal{F}}_Q^{-1}) \geq \lambda_m(\mathcal{F}_Q^{-1}) + \lambda_1(\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}) \implies \\ \lambda_1(\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}) \leq \lambda_1(\tilde{\mathcal{F}}_Q^{-1}) - \lambda_m(\mathcal{F}_Q^{-1}) \implies \\ \lambda_1(\tilde{\mathcal{F}}_Q^{-1} - \mathcal{F}_Q^{-1}) \leq \frac{1}{\lambda_m(\tilde{\mathcal{F}}_Q)} - \frac{1}{\lambda_1(\mathcal{F}_Q)}\end{aligned} \tag{A.40}$$

In that case, putting everything back in Eq. (A.39), the relative error can be upper bounded as:

$$\frac{\left\|\dot{\boldsymbol{\theta}}_Q - \dot{\tilde{\boldsymbol{\theta}}}_Q\right\|}{\left\|\dot{\boldsymbol{\theta}}_Q\right\|} \leq \frac{\lambda_1(\mathcal{F}_Q)}{\lambda_m(\tilde{\mathcal{F}}_Q)} - 1 \tag{A.41}$$

Bibliography

[1] Richard P Feynman.
Simulating physics with computers.
In *Feynman and computation*, pages 133–153. cRc Press, 2018.

[2] John Preskill.
Quantum computing in the NISQ era and beyond.
Quantum, 2:79, 2018.

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al.
Quantum supremacy using a programmable superconducting processor.
Nature, 574(7779):505–510, 2019.

[4] Rajeev Acharya, Laleh Aghababaie-Beni, Igor Aleiner, Trond I Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Nikita Astrakhantsev, Juan Atalaya, et al.
Quantum error correction below the surface code threshold.
arXiv preprint arXiv:2408.13687, 2024.

[5] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al.
Logical quantum processor based on reconfigurable atom arrays.
Nature, 626(7997):58–65, 2024.

[6] Daniel Stilck França and Raul Garcia-Patron.
Limitations of optimization algorithms on noisy quantum devices.
Nature Physics, 17(11):1221–1227, 2021.

[7] Peter W Shor.
Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.
SIAM review, 41(2):303–332, 1999.

[8] Lov K Grover.
A fast quantum mechanical algorithm for database search.
In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[9] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd.
Quantum algorithm for linear systems of equations.
Physical review letters, 103(15):150502, 2009.

[10] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al.
Noisy intermediate-scale quantum algorithms.
Reviews of Modern Physics, 94(1):015004, 2022.

[11] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al.
Variational quantum algorithms.
Nature Reviews Physics, 3(9):625–644, 2021.

[12] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan.
Variational ansatz-based quantum simulation of imaginary time evolution.
npj Quantum Information, 5(1):75, 2019.

[13] Ioannis Kolotouros, David Joseph, and Anand Kumar Narayanan.
Accelerating quantum imaginary-time evolution with random measurements.
arXiv preprint arXiv:2407.03123, 2024.

[14] Panagiotis Kl Barkoutsos, Jerome F Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J Egger, Matthias Troyer, Antonio Mezzacapo, et al.

Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions.
Physical Review A, 98(2):022322, 2018.

[15] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T Sornborger, and Patrick J Coles.
Quantum-assisted quantum compiling.
Quantum, 3:140, 2019.

[16] Zhimin He, Lvzhou Li, Shenggen Zheng, Yongyao Li, and Haozhen Situ.
Variational quantum compiling with double Q-learning.
New Journal of Physics, 23(3):033002, 2021.

[17] Julien Gacon, Jannes Nys, Riccardo Rossi, Stefan Woerner, and Giuseppe Carleo.
Variational quantum time evolution without the quantum geometric tensor.
Physical Review Research, 6(1):013143, 2024.

[18] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch.
Hardware-efficient variational quantum algorithms for time evolution.
Physical Review Research, 3(3):033083, 2021.

[19] Boniface Yogendran, Daniel Charlton, Miriam Beddig, Ioannis Kolotouros, and Petros Wallden.
Big data applications on small quantum computers.
arXiv preprint arXiv:2402.01529, 2024.

[20] Teague Tomesh, Pranav Gokhale, Eric R Anschuetz, and Frederic T Chong.
Coreset clustering on small quantum computers.
Electronics, 10(14):1690, 2021.

[21] Andrew Arrasmith, Zoë Holmes, Marco Cerezo, and Patrick J Coles.
Equivalence of quantum barren plateaus to cost concentration and narrow gorges.
Quantum Science and Technology, 7(4):045015, 2022.

[22] Juneseo Lee, Alicia B Magann, Herschel A Rabitz, and Christian Arenz.
Progress toward favorable landscapes in quantum combinatorial optimization.
Physical Review A, 104(3):032401, 2021.

[23] Joonho Kim and Yaron Oz.
Quantum energy landscape and circuit optimization.

Physical Review A, 106(5):052424, 2022.

[24] Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J Coles, and Marco Cerezo.
Theory of overparametrization in quantum neural networks.
Nature Computational Science, 3(6):542–551, 2023.

[25] Bobak Toussi Kiani, Seth Lloyd, and Reevu Maity.
Learning unitaries by gradient descent.
arXiv preprint arXiv:2001.11897, 2020.

[26] Eric R Anschuetz.
Critical points in quantum generative models.
arXiv preprint arXiv:2109.06957, 2021.

[27] Bálint Koczor and Simon C Benjamin.
Quantum analytic descent.
Physical Review Research, 4(2):023017, 2022.

[28] Samson Wang, Piotr Czarnik, Andrew Arrasmith, Marco Cerezo, Lukasz Cincio, and Patrick J Coles.
Can error mitigation improve trainability of noisy variational quantum algorithms?
Quantum, 8:1287, 2024.

[29] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J Coles, and Marco Cerezo.
Diagnosing barren plateaus with tools from quantum optimal control.
Quantum, 6:824, 2022.

[30] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo.
Quantum natural gradient.
Quantum, 4:269, 2020.

[31] David Wierichs, Christian Gogolin, and Michael Kastoryano.
Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer.
Physical Review Research, 2(4):043246, 2020.

[32] Johannes Jakob Meyer.
Fisher information in noisy intermediate-scale quantum applications.

Quantum, 5:539, 2021.

[33] Jing Liu, Haidong Yuan, Xiao-Ming Lu, and Xiaoguang Wang.
Quantum Fisher information matrix and multiparameter estimation.
Journal of Physics A: Mathematical and Theoretical, 53(2):023001, 2020.

[34] Tobias Haug, Kishor Bharti, and MS Kim.
Capacity and quantum geometry of parametrized quantum circuits.
PRX Quantum, 2(4):040309, 2021.

[35] Tobias Haug and MS Kim.
Generalization with quantum geometry for learning unitaries.
arXiv preprint arXiv:2303.13462, 2023.

[36] Tobias Haug and MS Kim.
Natural parametrized quantum circuit.
Physical Review A, 106(5):052611, 2022.

[37] Eric R Anschuetz and Bobak T Kiani.
Quantum variational algorithms are swamped with traps.
Nature Communications, 13(1):7760, 2022.

[38] Xuchen You and Xiaodi Wu.
Exponentially many local minima in quantum neural networks.
In *International Conference on Machine Learning*, pages 12144–12155. PMLR, 2021.

[39] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik.
Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms.
Advanced Quantum Technologies, 2(12):1900070, 2019.

[40] Kouhei Nakaji and Naoki Yamamoto.
Expressibility of the alternating layered ansatz for quantum computation.
Quantum, 5:434, 2021.

[41] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner.
The power of quantum neural networks.
Nature Computational Science, 1(6):403–409, 2021.

[42] Michael Ragone, Bojko N Bakalov, Frédéric Sauvage, Alexander F Kemper, Carlos Ortiz Marrero, Martin Larocca, and M Cerezo.
A unified theory of barren plateaus for deep parametrized quantum circuits.
arXiv preprint arXiv:2309.09342, 2023.

[43] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J Coles, Lukasz Cincio, Jarrod R McClean, Zoë Holmes, and M Cerezo.
A review of barren plateaus in variational quantum computing.
arXiv preprint arXiv:2405.00781, 2024.

[44] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles.
Noise-induced barren plateaus in variational quantum algorithms.
Nature communications, 12(1):6961, 2021.

[45] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles.
Cost function dependent barren plateaus in shallow parametrized quantum circuits.
Nature communications, 12(1):1791, 2021.

[46] Kaining Zhang, Min-Hsiu Hsieh, Liu Liu, and Dacheng Tao.
Toward trainability of quantum neural networks.
arXiv preprint arXiv:2011.06258, 2020.

[47] Tyler Volkoff and Patrick J Coles.
Large gradients via correlation in random parameterized quantum circuits.
Quantum Science and Technology, 6(2):025008, 2021.

[48] Arthur Pesah, Marco Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J Coles.
Absence of barren plateaus in quantum convolutional neural networks.
Physical Review X, 11(4):041011, 2021.

[49] Léo Monbroussou, Jonas Landman, Alex B Grilo, Romain Kukla, and Elham Kashefi.
Trainability and expressivity of hamming-weight preserving quantum circuits for machine learning.
arXiv preprint arXiv:2309.15547, 2023.

[50] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti.

An initialization strategy for addressing barren plateaus in parametrized quantum circuits.
Quantum, 3:214, 2019.

[51] Chae-Yeun Park and Nathan Killoran.
Hamiltonian variational ansatz without barren plateaus.
Quantum, 8:1239, 2024.

[52] Javier Rivera-Dean, Patrick Huembeli, Antonio Acín, and Joseph Bowles.
Avoiding local minima in variational quantum algorithms with neural networks.
arXiv preprint arXiv:2104.02955, 2021.

[53] Ruslan Shaydulin, Ilya Safro, and Jeffrey Larson.
Multistart methods for quantum approximate optimization.
In *2019 IEEE high performance extreme computing conference (HPEC)*, pages 1–8. IEEE, 2019.

[54] Andreas Elben, Steven T Flammia, Hsin-Yuan Huang, Richard Kueng, John Preskill, Benoît Vermersch, and Peter Zoller.
The randomized measurement toolbox.
Nature Reviews Physics, 5(1):9–24, 2023.

[55] Andreas Elben, Benoît Vermersch, Christian F Roos, and Peter Zoller.
Statistical correlations between locally randomized measurements: A toolbox for probing entanglement in many-body quantum states.
Physical Review A, 99(5):052323, 2019.

[56] Andreas Elben, Benoît Vermersch, Marcello Dalmonte, J Ignacio Cirac, and Peter Zoller.
Rényi entropies from random quenches in atomic Hubbard and spin models.
Physical review letters, 120(5):050406, 2018.

[57] Andreas Elben, Richard Kueng, Hsin-Yuan Robert Huang, Rick van Bijnen, Christian Kokail, Marcello Dalmonte, Pasquale Calabrese, Barbara Kraus, John Preskill, Peter Zoller, et al.
Mixed-state entanglement from local randomized measurements.
Physical Review Letters, 125(20):200501, 2020.

[58] Andreas Elben, Benoît Vermersch, Rick van Bijnen, Christian Kokail, Tiff Brydges, Christine Maier, Manoj K Joshi, Rainer Blatt, Christian F Roos, and Peter Zoller.
Cross-platform verification of intermediate scale quantum devices.
Physical review letters, 124(1):010504, 2020.

[59] Hsin-Yuan Huang, Richard Kueng, and John Preskill.
Predicting many properties of a quantum system from very few measurements.
Nature Physics, 16(10):1050–1057, 2020.

[60] Ioannis Kolotouros and Petros Wallden.
Random natural gradient.
Quantum, 8:1503, 2024.

[61] Panagiotis Kl Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner.
Improving variational quantum optimization using CVaR.
Quantum, 4:256, 2020.

[62] Ioannis Kolotouros and Petros Wallden.
Evolving objective function for improved variational quantum optimization.
Physical Review Research, 4(2):023225, 2022.

[63] Ioannis Kolotouros, Ioannis Petrongonas, Miloš Prokop, and Petros Wallden.
Simulating adiabatic quantum computing with parameterized quantum circuits.
Quantum Science and Technology, 10(1):015003, 2024.

[64] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta.
Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets.
Nature, 549(7671):242–246, 2017.

[65] Stuart Hadfield, Zhihui Wang, Bryan O'gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas.
From the quantum approximate optimization algorithm to a quantum alternating operator ansatz.
Algorithms, 12(2):34, 2019.

[66] Francesco A Evangelista, Garnet Kin Chan, and Gustavo E Scuseria.
Exact parameterization of fermionic wave functions via unitary coupled cluster theory.
The Journal of chemical physics, 151(24), 2019.

[67] Andrea Mari, Thomas R Bromley, and Nathan Killoran.
Estimating the gradient and higher-order derivatives on quantum hardware.
Physical Review A, 103(1):012405, 2020.

[68] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran.
Evaluating analytic gradients on quantum hardware.
Physical Review A, 99(3):032331, 2019.

[69] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin.
General parameter-shift rules for quantum gradients.
Quantum, 6:677, 2022.

[70] Leonardo Banchi and Gavin E Crooks.
Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule.
Quantum, 5:386, 2021.

[71] Li Li, Minjie Fan, Marc Coram, Patrick Riley, and Stefan Leichenauer.
Quantum optimization with a novel Gibbs objective function and ansatz architecture search.
Physical Review Research, 2(2):023074, 2020.

[72] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien.
A variational eigenvalue solver on a photonic quantum processor.
Nature communications, 5(1):4213, 2014.

[73] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen.
Exploring entanglement and optimization within the Hamiltonian variational ansatz.
PRX quantum, 1(2):020319, 2020.

[74] Alexandre Choquette, Agustin Di Paolo, Panagiotis Kl Barkoutsos, David Sénéchal, Ivano Tavernelli, and Alexandre Blais.

Quantum-optimal-control-inspired ansatz for variational quantum algorithms.
Physical Review Research, 3(2):023092, 2021.

[75] Giacomo Nannicini.
Performance of hybrid quantum-classical variational heuristics for combinatorial optimization.
Physical Review E, 99(1):013304, 2019.

[76] Joonho Lee, William J Huggins, Martin Head-Gordon, and K Birgitta Whaley.
Generalized unitary coupled cluster wave functions for quantum computation.
Journal of chemical theory and computation, 15(1):311–324, 2018.

[77] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven.
Barren plateaus in quantum neural network training landscapes.
Nature communications, 9(1):4812, 2018.

[78] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann.
A quantum approximate optimization algorithm.
arXiv preprint arXiv:1411.4028, 2014.

[79] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G Rieffel.
Quantum approximate optimization algorithm for MaxCut: A fermionic view.
Physical Review A, 97(2):022304, 2018.

[80] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin.
Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices.
Physical Review X, 10(2):021067, 2020.

[81] Asier Ozaeta, Wim van Dam, and Peter L McMahon.
Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems.
Quantum Science and Technology, 7(4):045036, 2022.

[82] Michael Streif and Martin Leib.
Forbidden subspaces for level-1 quantum approximate optimization algorithm and instantaneous quantum polynomial circuits.
Physical Review A, 102(4):042416, 2020.

[83] Sami Boulebnane and Ashley Montanaro.
Solving boolean satisfiability problems with the quantum approximate optimization algorithm.
arXiv preprint arXiv:2208.06909, 2022.

[84] Stefan H. Sack and Maksym Serbyn.
Quantum annealing initialization of the quantum approximate optimization algorithm.
Quantum, 5:491, July 2021.

[85] Nishant Jain, Brian Coyle, Elham Kashefi, and Niraj Kumar.
Graph neural network initialisation of quantum approximate optimisation.
Quantum, 6:861, 2022.

[86] Christos H Papadimitriou and Kenneth Steiglitz.
Combinatorial optimization: algorithms and complexity.
Courier Corporation, 2013.

[87] Andrew Lucas.
Ising formulations of many NP problems.
Frontiers in physics, 2:5, 2014.

[88] Gavin E Crooks.
Performance of the quantum approximate optimization algorithm on the maximum cut problem.
arXiv preprint arXiv:1811.08419, 2018.

[89] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang.
Hybrid quantum-classical algorithms for approximate graph coloring.
Quantum, 6:678, 2022.

[90] Johan Håstad.
Some optimal inapproximability results.
Journal of the ACM (JACM), 48(4):798–859, 2001.

[91] Michel X Goemans and David P Williamson.
Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.
Journal of the ACM (JACM), 42(6):1115–1145, 1995.

[92] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang.
Obstacles to variational quantum optimization from symmetry protection.
Physical review letters, 125(26):260505, 2020.

[93] Charles Moussa, Henri Calandra, and Vedran Dunjko.
To quantum or not to quantum: towards algorithm selection in near-term quantum optimization.
Quantum Science and Technology, 5(4):044009, 2020.

[94] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou.
The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size.
Quantum, 6:759, 2022.

[95] Stephan Mertens.
Number partitioning.
Computational Complexity and Statistical Physics, page 125, 2006.

[96] Richard E Korf.
Multi-way number partitioning.
In *IJCAI*, volume 9, pages 538–543, 2009.

[97] Roman Orus, Samuel Mugel, and Enrique Lizaso.
Quantum computing for finance: Overview and prospects.
Reviews in Physics, 4:100028, 2019.

[98] Davide Venturelli and Alexei Kondratyev.
Reverse quantum annealing approach to portfolio optimization problems.
Quantum Machine Intelligence, 1(1):17–30, 2019.

[99] Hans Kellerer, Renata Mansini, and M Grazia Speranza.
Selecting portfolios with fixed costs and minimum transaction lots.
Annals of Operations Research, 99(1):287–304, 2000.

[100] Daniel J Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain.
Quantum computing for Finance: state of the art and future prospects.
IEEE Transactions on Quantum Engineering, 2020.

[101] Daniel J Egger, Jakub Mareček, and Stefan Woerner.
Warm-starting quantum optimization.
Quantum, 5:479, 2021.

[102] Jeffrey Cohen, Alex Khan, and Clark Alexander.
Portfolio optimization of 40 stocks using the DWave quantum annealer.
arXiv preprint arXiv:2007.01430, 2020.

[103] N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang.
Quantum walk-based portfolio optimisation.
Quantum, 5:513, July 2021.

[104] Samuel Mugel, Carlos Kuchkovsky, Escolástico Sánchez, Samuel Fernández-Lorenzo, Jorge Luis-Hita, Enrique Lizaso, and Román Orús.
Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks.
Physical Review Research, 4(1):013006, 2022.

[105] Wen Wei Ho and Timothy H Hsieh.
Efficient variational simulation of non-trivial quantum states.
SciPost Physics, 6(3):029, 2019.

[106] Joris Kattemölle and Jasper Van Wezel.
Variational quantum eigensolver for the Heisenberg antiferromagnet on the kagome lattice.
Physical Review B, 106(21):214429, 2022.

[107] Manpreet Singh Jattana, Fengping Jin, Hans De Raedt, and Kristel Michielsen.
Assessment of the variational quantum eigensolver: application to the Heisenberg model.
Frontiers in Physics, 10:907160, 2022.

[108] Abhinav Anand, Matthias Degroote, and Alán Aspuru-Guzik.
Natural evolutionary strategies for variational quantum computation.
Machine Learning: Science and Technology, 2(4):045012, 2021.

[109] Tianchen Zhao, Giuseppe Carleo, James Stokes, and Shravan Veerapaneni.
Natural evolution strategies and variational Monte Carlo.
Machine Learning: Science and Technology, 2(2):02LT01, 2020.

[110] Amira Abbas, Robbie King, Hsin-Yuan Huang, William J Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod R McClean.
On quantum backpropagation, information reuse, and cheating measurement collapse.
arXiv preprint arXiv:2305.13362, 2023.

[111] Joseph Bowles, David Wierichs, and Chae-Yeun Park.
Backpropagation scaling in parameterised quantum circuits.
arXiv preprint arXiv:2306.14962, 2023.

[112] Shun-Ichi Amari.
Natural gradient works efficiently in learning.
Neural computation, 10(2):251–276, 1998.

[113] Bálint Koczor and Simon C Benjamin.
Quantum natural gradient generalized to noisy and nonunitary circuits.
Physical Review A, 106(6):062416, 2022.

[114] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandao, and Garnet Kin-Lic Chan.
Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution.
Nature Physics, 16(2):205–210, 2020.

[115] Stefan H Sack, Raimel A Medina, Alexios A Michailidis, Richard Kueng, and Maksym Serbyn.
Avoiding barren plateaus using classical shadows.
PRX Quantum, 3(2):020365, 2022.

[116] Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J O'Riordan, Thomas R Bromley, Juan Miguel Arrazola, and Nathan Killoran.
Fast quantum circuit cutting with randomized measurements.
Quantum, 7:934, 2023.

[117] Aram W Harrow, Ashley Montanaro, and Anthony J Short.
Limitations on quantum dimensionality reduction.
International Journal of Quantum Information, 13(04):1440001, 2015.

[118] Stephen J Wright.

Coordinate descent algorithms.
Mathematical programming, 151(1):3–34, 2015.

[119] Yu Nesterov.
Efficiency of coordinate descent methods on huge-scale optimization problems.
SIAM Journal on Optimization, 22(2):341–362, 2012.

[120] Paul Tseng.
Convergence of a block coordinate descent method for nondifferentiable minimization.
Journal of optimization theory and applications, 109:475–494, 2001.

[121] David JC MacKay.
Information theory, inference and learning algorithms.
Cambridge university press, 2003.

[122] Elena Alexandra Morozova and Nikolai Nikolaevich Chentsov.
Markov invariant geometry on manifolds of states.
Journal of Soviet Mathematics, 56:2648–2669, 1991.

[123] Dénes Petz.
Monotone metrics on matrix spaces.
Linear algebra and its applications, 244:81–96, 1996.

[124] Stephen P Boyd and Lieven Vandenberghe.
Convex optimization.
Cambridge university press, 2004.

[125] Xiaoyang Wang, Xu Feng, Tobias Hartung, Karl Jansen, and Paolo Stornati.
Critical behavior of the Ising model by preparing the thermal state on a quantum computer.
Physical Review A, 108(2):022612, 2023.

[126] Francesco Turro.
Quantum imaginary time propagation algorithm for preparing thermal states.
arXiv preprint arXiv:2306.16580, 2023.

[127] Niladri Gomes, Anirban Mukherjee, Feng Zhang, Thomas Iadecola, Cai-Zhuang Wang, Kai-Ming Ho, Peter P Orth, and Yong-Xin Yao.

Adaptive variational quantum imaginary time evolution approach for ground state preparation.
Advanced Quantum Technologies, 4(12):2100114, 2021.

[128] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin.
Theory of variational quantum simulation.
Quantum, 3:191, 2019.

[129] Diego García-Martín, Martin Larocca, and M Cerezo.
Effects of noise on the overparametrization of quantum neural networks.
arXiv preprint arXiv:2302.05059, 2023.

[130] Naoki Yamamoto.
On the natural gradient for variational quantum eigensolver.
arXiv preprint arXiv:1909.05074, 2019.

[131] Neal Parikh, Stephen Boyd, et al.
Proximal algorithms.
Foundations and trends® in Optimization, 1(3):127–239, 2014.

[132] Samuel L Braunstein and Carlton M Caves.
Statistical distance and the geometry of quantum states.
Physical Review Letters, 72(22):3439, 1994.

[133] Luca Pezzè, Mario A Ciampini, Nicolò Spagnolo, Peter C Humphreys, Animesh Datta, Ian A Walmsley, Marco Barbieri, Fabio Sciarrino, and Augusto Smerzi.
Optimal measurements for simultaneous quantum estimation of multiple phases.
Physical review letters, 119(13):130504, 2017.

[134] Antonio Anna Mele.
Introduction to Haar measure tools in quantum information: A beginner's tutorial.
Quantum, 8:1340, 2024.

[135] Scott Aaronson and Daniel Gottesman.
Improved simulation of stabilizer circuits.
Physical Review A, 70(5):052328, 2004.

[136] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner.
Simultaneous perturbation stochastic approximation of the quantum Fisher information.

Quantum, 5:567, 2021.

[137] Jorge J Moré and Stefan M Wild.
Benchmarking derivative-free optimization algorithms.
SIAM Journal on Optimization, 20(1):172–191, 2009.

[138] Michael JD Powell.
A direct search optimization method that models the objective and constraint functions by linear interpolation.
In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

[139] Aric Hagberg, Pieter Swart, and Daniel S Chult.
Exploring network structure, dynamics, and function using NetworkX.
Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[140] Vishwanathan Akshay, H Philathong, Igor Zacharov, and J Biamonte.
Reachability deficits in quantum approximate optimization of graph problems.
Quantum, 5:532, 2021.

[141] Tameem Albash and Daniel A Lidar.
Adiabatic quantum computation.
Reviews of Modern Physics, 90(1):015002, 2018.

[142] Mohammad HS Amin.
Consistency of the adiabatic theorem.
Physical review letters, 102(22):220401, 2009.

[143] Wim Van Dam, Michele Mosca, and Umesh Vazirani.
How powerful is adiabatic quantum computation?
In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 279–287. IEEE, 2001.

[144] Conor Mc Keever and Michael Lubasch.
Towards adiabatic quantum computing using compressed quantum circuits.
PRX Quantum, 5(2):020362, 2024.

[145] A Garcia-Saez and JI Latorre.
Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers.

arXiv preprint arXiv:1806.02287, 2018.

[146] Stuart M Harwood, Dimitar Trenev, Spencer T Stober, Panagiotis Barkoutsos, Tanvi P Gujarati, Sarah Mostame, and Donny Greenberg.
Improving the variational quantum eigensolver using variational adiabatic quantum computing.
ACM Transactions on Quantum Computing, 3(1):1–20, 2022.

[147] Ming-Cheng Chen, Ming Gong, Xiaosi Xu, Xiao Yuan, Jian-Wen Wang, Can Wang, Chong Ying, Jin Lin, Yu Xu, Yulin Wu, et al.
Demonstration of adiabatic variational quantum computing with a superconducting quantum coprocessor.
Physical Review Letters, 125(18):180501, 2020.

[148] Pranav Chandarana, Narendra N Hegade, Koushik Paul, Francisco Albarrán-Arriagada, Enrique Solano, Adolfo del Campo, and Xi Chen.
Digitized-counterdiabatic quantum approximate optimization algorithm.
Physical Review Research, 4(1):013141, 2022.

[149] Mohamed Hibat-Allah, Estelle M Inack, Roeland Wiersema, Roger G Melko, and Juan Carrasquilla.
Variational neural annealing.
Nature Machine Intelligence, 3(11):952–961, 2021.

[150] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler.
Bounds for the adiabatic approximation with applications to quantum computation.
Journal of Mathematical Physics, 48(10):102111, 2007.

[151] Alexander Elgart and George A Hagedorn.
A note on the switching adiabatic theorem.
Journal of Mathematical Physics, 53(10):102202, 2012.

[152] Pedro CS Costa, Dong An, Yuval R Sanders, Yuan Su, Ryan Babbush, and Dominic W Berry.
Optimal scaling quantum linear-systems solver via discrete adiabatic theorem.
PRX Quantum, 3(4):040303, 2022.

[153] Silvano Garnerone, Paolo Zanardi, and Daniel A Lidar.
Adiabatic quantum algorithm for search engine ranking.
Physical review letters, 108(23):230506, 2012.

[154] Elizabeth R Bennewitz, Florian Hopfmueller, Bohdan Kulchytskyy, Juan Carrasquilla, and Pooya Ronagh.
Neural error mitigation of near-term quantum simulations.
Nature Machine Intelligence, 4(7):618–624, 2022.

[155] Patrick Huembeli and Alexandre Dauphin.
Characterizing the loss landscape of variational quantum circuits.
Quantum Science and Technology, 6(2):025011, 2021.

[156] Zhiyan Ding, Taehee Ko, Jiahao Yao, Lin Lin, and Xiantao Li.
Random coordinate descent: a simple alternative for optimizing parameterized quantum circuits.
arXiv preprint arXiv:2311.00088, 2023.

[157] Gerald B Folland.
Higher-order derivatives and Taylor's formula in several variables.
Preprint, pages 1–4, 2005.

[158] Stephen Boyd, Lin Xiao, and Almir Mutapcic.
Subgradient methods.
lecture notes of EE392o, Stanford University, Autumn Quarter, 2004(01), 2003.

[159] Roger Fletcher.
Practical methods of optimization.
John Wiley & Sons, 2000.

[160] Simone Notarnicola, Andreas Elben, Thierry Lahaye, Antoine Browaeys, Simone Montangero, and Benoît Vermersch.
A randomized measurement toolbox for an interacting Rydberg-atom quantum simulator.
New Journal of Physics, 25(10):103006, 2023.

[161] Tiff Brydges, Andreas Elben, Petar Jurcevic, Benoît Vermersch, Christine Maier, Ben P Lanyon, Peter Zoller, Rainer Blatt, and Christian F Roos.
Probing Rényi entanglement entropy via randomized measurements.
Science, 364(6437):260–263, 2019.

[162] Daniel A Roberts and Beni Yoshida.
Chaos and complexity by design.
Journal of High Energy Physics, 2017(4):1–64, 2017.

[163] Barry Simon.
Representations of finite and compact groups.
Number 10. American Mathematical Soc., 1996.