QUANTUM ALGORITHMS FOR SIMULATION, OPTIMIZATION, AND ESTIMATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Dhrumilkumar Patel

December 2025

QUANTUM ALGORITHMS FOR SIMULATION, OPTIMIZATION, AND ESTIMATION

Dhrumilkumar Patel, Ph.D.

Cornell University 2025

Quantum algorithms hold the promise of fundamentally changing the way we approach problems in physics, chemistry, computer science, and engineering. This thesis explores new provably efficient quantum algorithms for three fundamental directions in quantum computing: simulation, optimization, and estimation. Together, our results in each of these directions contribute to the broader effort of making quantum computation a practical tool for science and engineering. First, in the area of quantum simulation, we propose Wave Matrix Lindbladization, a set of algorithms that extend the philosophy of density matrix exponentiation to Lindbladian dynamics by working with program-state encodings of Lindblad operators. Specifically, when these operators are local, we show that these algorithms are provably efficient by analyzing their sample and gate complexities. Second, for optimization, we study quantum Boltzmann machines as a variational ansatz for ground-state energy estimation, proving efficient gradient estimation and convergence to approximate stationary points. We also propose hybrid quantum-classical algorithms for semidefinite programming, establish rigorous convergence guarantees, and validate their robustness to noise through numerical simulations for applications such as MaxCut. Finally, for estimation, we develop the tapered quantum phase estimation algorithm, a coherent phase estimation method that leverages taper/window functions from signal processing to reach optimal query complexity without relying on expensive coherent median techniques, and we provide error guarantees in both asymptotic and non-asymptotic regimes along with efficient taper state preparation.

BIOGRAPHICAL SKETCH

Dhrumil Patel completed his schooling at the H. B. Kapadia New High School in Ahmedabad, Gujarat, India. He earned a bachelor's degree in Computer Science from the International Institute of Information Technology, India, in 2019, and a master's degree in Computer Science from Louisiana State University, in 2022, under the mentorship of Prof. Rahul Shah, where he worked on algorithms for string problems. He is currently a Ph.D. candidate in Computer Science at Cornell University, advised by Prof. Mark M. Wilde, with research focused on quantum algorithms. His research interests center on finding new and efficient ways to solve problems, whether through classical or quantum algorithms.

I dedicate this dissertation to my beloved family,
the foundation of all my work.

ACKNOWLEDGEMENTS

My Ph.D. journey has been a challenging yet rewarding path, filled with unexpected turns, moments of struggle, and lessons that shaped both my work and myself. I could not have reached this point without the support, guidance, and encouragement of many people, to whom I am truly grateful.

Among them, I owe my sincerest thanks to my advisor, Prof. Mark M. Wilde, for sparking and nurturing my interest in quantum computing. I could not have asked for a better guide in exploring this fascinating field and for encouraging me to grow as a researcher. It has always been inspiring to see his dedication towards research. I still remember being introduced to quantum information theory in 2016 through his book. Moving from reading his book and papers to working with him directly has been an unexpected but remarkable step in my research journey, and I am truly thankful for it.

I would also like to thank Prof. Jayadev Acharya and Prof. Eilyan Bitar for serving on my committee. I am grateful for their insightful questions, constructive feedback, and the thoughtful discussions I had with them during this process.

Some of the most memorable parts of my research came through collaboration. The exchange of ideas and the variety of perspectives not only improved my work but also shaped the way I approach problems. During my long-term graduate assistantship at Los Alamos National Lab and beyond, I was fortunate to work closely with Yiğit Subaşı, Andrew T. Sornborger, Sam Slezak, my wonderful friend Shi Jie Samuel Tan, Kristian Temme, and Jiaqing Jiang. I truly enjoyed the many discussions we shared, and it was rewarding to see how naturally our research interests connected. These collaborations were not only intellectually stimulating but also a lot of fun. I also had the chance to collaborate during my internship at Phasecraft, where I am especially grateful to Steven Flammia, Raul García-Patrón, and Laura Clinton for a wonderful collaboration, as well

as to Adrian Chapman, Marcos Crichigno, and Andrew Projansky for many stimulating conversations.

Beyond these, I was fortunate to meet and work with many other collaborators who brought fresh ideas and lasting inspiration to my work, including Rahul Bandyopadhyay, Jingxuan Chen, Patrick Coles, Ziv Goldfeld, Byeongseon Go, Zoë Holmes, Daniel Koch, Hyukjoon Kwon, Nana Liu, Ivy Luo, Michele Minervini, Theshani Nuradha, Siheon Park, Saahil Patel, Aby Philip, Alex H. Rubin, Marina Radulaski, Soorya Rethinasamy, Sreejith Sreekumar, Aidan N. Sims, Kathie Wang, and Hanna Westerheim. Each has left a meaningful impact on my growth as a researcher, and I am thankful for their contributions.

I am especially grateful to Rahul Shah and Avni Rambhia for being like family to me during my Ph.D. Their support helped me through some rough patches in my personal life, and I will always value their presence in this journey. I also thank Rahul for sparking my interest in theoretical computer science, especially algorithms and data structures, and for teaching me, through his constant challenges, how to approach problems and think of new ideas. I could always count on him, even for a midnight call.

I am deeply grateful to my friends and peers, including Shalin, Nidhi, Muneera, Sanjay, Charles, Adithya, Ramya, Harshini, Vishal, Aby, Hemant Mishra, Hemanth (Ven), Aliza, Michele, Aidan, Theshani, Kaiyuan, Soorya, and Hami for their constant support and understanding. I am grateful for all the friendships I had along the way. Each friendship gave me a different perspective on life. I will always cherish the late-night philosophical talks, random research discussions, the delicious Indian food we shared, and the laughter from watching random comedy movie scenes together.

Finally, I would like to thank my family, without whom none of this would have been possible. I am forever grateful to my mother, Rajeshwari Patel, for being my source of endless love and emotional strength, and to my father, Jagdish Patel, whose steady support has been the backbone of my life. They gave me the confidence to take risks, the

courage to face challenges, and the freedom to pursue my dreams. I also want to thank my younger brother, Poojan, for always standing by my side. Sharing the same sense of humor made even the toughest days lighter, and his presence has been a constant reminder that I was never alone in this journey.

The most beautiful part of my Ph.D. journey has been meeting the love of my life, my wife, Simran. Her patience, encouragement, and unwavering belief in me have carried me through difficult times during my Ph.D. and made the good moments even more meaningful. Words cannot truly capture how much her support has meant to me.

I would also like to thank my in-laws, Bharti and Rakesh Dhingra, for welcoming me as part of their family. I will always treasure the endless tea-time chats with my mother-in-law and the lively conversations with my father-in-law about politics and badminton. I am especially grateful to my brother-in-law, Abhi, for his incredible support during my wedding preparations in India and for the wonderful gesture of adjusting his own wedding date to accommodate my Ph.D. defense date. And speaking of his wedding, I am also thankful to my soon-to-be sister-in-law, Soundarya, another newcomer to the family like me, for making me feel so welcomed, and for her amazing cake videos that I can only watch but not taste, which always leave me with late-night cravings.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

WAVE MATRIX LINDBLADIZATION I: QUANTUM PROGRAMS FOR SIMULATING MARKOVIAN DYNAMICS[1]

1.1 Abstract

Density Matrix Exponentiation is a technique for simulating Hamiltonian dynamics when the Hamiltonian to be simulated is available as a quantum state. In this chapter, we present a natural analogue to this technique, for simulating Markovian dynamics governed by the well known Lindblad master equation. For this purpose, we first propose an input model in which a Lindblad operator L is encoded into a quantum state ψ. Then, given access to n copies of the state ψ, the task is to simulate the corresponding Markovian dynamics for time t. We propose a quantum algorithm for this task, called Wave Matrix Lindbladization, and we also investigate its sample complexity. In particular, we show that when L is local, our algorithm approximates the target dynamics to within $O(\varepsilon)$ approximation error using only $n = O(t^2/\varepsilon)$ copies of ψ.

1.2 Introduction

1.2.1 Background

By the early 1980s, it was evident that simulating the behavior of complex quantum systems using a classical computer is computationally expensive, since it involves keeping

[1]This chapter is based verbatim on the work [PW23], with modifications to incorporate local Lindblad operators. An updated version will be posted to arXiv soon.

track of an exponentially large number of quantum state amplitudes. In order to overcome this difficulty, Feynman proposed the idea of simulating quantum systems using a computational device that is itself quantum mechanical [Fey82]. Originally, it was merely a conjecture; however, it gave birth to the field of quantum simulation [Llo96, GAN14], which is currently one of the most anticipated applications of a quantum computer. The essential idea behind quantum simulation involves using a quantum computer to perform the simulation, which can then allow for a detailed investigation of the quantum system being simulated.

Hamiltonian simulation is a particular kind of quantum simulation that involves simulating the behavior of a closed quantum system. This is an extensively investigated area, and many quantum algorithms have been developed to date to solve this specific problem [Llo96, BCC+13, BCG14, LMR14, BCC+14, BCK15, LC17, KLL+17].

While Hamiltonian simulation is a well studied problem, it is limited to only closed quantum systems. In many real-world scenarios, quantum systems are subject to the influence of an environment, leading to more complex dynamics that are better described by open system models. Moreover, if the dynamics of an open system are Markovian in nature (i.e., its quantum state at time $t+\Delta$ only depends on the quantum state at time t and is independent of states before time t), then such dynamics are well captured by the Lindblad master equation. The general form of this equation was delineated independently by Göran Lindblad [Lin76] and by Gorini, Kossakowski, and Sudarshan [GKS08], and so, this equation is also known as the Gorini–Kossakowski–Sudarshan–Lindblad equation. The significance of this master equation cannot be overstated. It is crucial in understanding the behavior of a wide range of quantum systems and scenarios [BP02, Wei21], including condensed matter [Pro11, MTAB12, OLG12], quantum chemistry [Nit06, MK08], quantum optics [PK98, GZ04], entanglement preparation [KBD+08, KRS11, RRS16], ther-

mal state preparation [KB14], quantum state engineering [VWC09], and the effects of noise on quantum computers [MPGC13].

In this chapter, we consider the problem of simulating the Lindbladian evolution of a finite-dimensional quantum system in an initial state ρ for time t. This evolution is governed by the following Lindblad master equation:

$$\frac{\partial \rho}{\partial t} = \mathcal{L}(\rho) := -i[H, \rho] + \sum_{k=1}^{K} L_k \rho L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho\right\}, \tag{1.1}$$

where H is a Hermitian operator representing the system's Hamiltonian, and the operators $\{L_k\}_{k=1}^K$ are called Lindblad operators, which are not necessarily Hermitian and in fact have no constraints on them. In addition, the superoperator $\mathcal{L}$ is known as a Lindbladian. The notation $\{A, B\}$ above refers to the anti-commutator of operators A and B, i.e., $\{A, B\} = AB + BA$. By simulating the aforementioned evolution for time t, we mean implementing its corresponding quantum channel $e^{\mathcal{L}t}$, which is the solution of (1.1), where

$$e^{\mathcal{L}t}(\rho) = \sum_{k=0}^{\infty} \frac{\mathcal{L}^k(\rho) t^k}{k!}, \tag{1.2}$$

and $\mathcal{L}^k$ denotes k sequential applications of the Lindbladian $\mathcal{L}$. For small t, note that $e^{\mathcal{L}t}(\rho) = \rho + \mathcal{L}(\rho)t + O(t^2)$, and we make use of this expansion in what follows.

Throughout this chapter, we focus on a simple case in which the Lindblad master equation consists of only a single Lindblad operator L. For clarity, we rewrite the Lindbladian corresponding to this simple case:

$$\mathcal{L}(\rho) = L\rho L^\dagger - \frac{1}{2}\left\{L^\dagger L, \rho\right\}. \tag{1.3}$$

We use this basic scenario as a starting point, as it is easier to grasp the intuition behind the techniques we introduce here. Furthermore, one can easily extend this case to simulate more complex Lindbladian evolutions with multiple Lindblad operators, by using

Proposition 2 of [CL16]. Specifically, this proposition states that, given efficient implementations of polynomially many Lindbladians $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m$, one can efficiently implement their linear combination $\sum_{i=1}^{m} \mathcal{L}_i$.

More recently, there has been growing interest in developing efficient quantum algorithms for simulating the dynamics of open quantum systems, as given by (1.1) [CL16, CW17, KSMM22, SHMS+22, SBH+23] (see [MOTT22] for a review). These works are primarily based on the assumption that some succinct representation of the Lindblad operators or black-box access to them are provided beforehand. For example, a list of non-zero coefficients when writing these operators as a linear combination of Paulis is one such succinct representation [CW17].

In this chapter, we approach the above problem from a different angle. We assume that the Lindblad operator L is encoded in a pure quantum state $|\psi\rangle$, and we have access to multiple copies of this state. That is, we suppose that L is encoded in $|\psi\rangle$ in the following manner:

$$|\psi\rangle := (L \otimes I)|\Gamma\rangle, \tag{1.4}$$

where $|\Gamma\rangle := \sum_j |j\rangle|j\rangle$ is a maximally entangled vector. This way of encoding lies at the heart of our quantum algorithm, and as far as we are aware, it is the first time that such an encoding scheme has been proposed. We refer to such a state as a program state, as it can be programmed to encode any square linear operator according to the problem at hand. The only constraint on the operator L, encoded as above, is that $\|L\|_2 = 1$, where $\|A\|_2 := \sqrt{\mathrm{Tr}[A^\dagger A]}$ is the Schatten-2 norm of a matrix A (also known as the Hilbert–Schmidt norm). This constraint on L arises from the fact that $|\psi\rangle$ is a quantum state. That being said, for encoding a Lindblad operator L' with an arbitrary norm and corresponding Lindbladian $\mathcal{L}'$, we can suppose that its normalized version, i.e., $L'/\|L'\|_2$, is encoded in

a quantum state. Then, for simulating its corresponding quantum channel $e^{\mathcal{L}'t}$, we simulate the channel $e^{\mathcal{L}'t'/\|L'\|_2^2}$ for time $t' = \|L'\|_2^2 t$, so that $e^{\mathcal{L}'t} = e^{\mathcal{L}'t'/\|L'\|_2^2}$. This is evident from (1.3). Thus, without loss of generality, we can assume that the Lindblad operator L is normalized, i.e., $\|L\|_2 = 1$, and we do so throughout this chapter.

We refer to this newly introduced method of Lindbladian simulation as *Wave Matrix Lindbladization*. The reasoning behind this terminology is that L is known as a wave matrix [MZB23], and we are "lindbladizing" it, i.e., transforming it from a wave matrix into a Lindblad operator. Essentially, we ask: given one copy of an unknown quantum state ρ and n copies of the program state $\psi := |\psi\rangle\langle\psi|$, can we approximately implement the quantum channel $e^{\mathcal{L}t}$ up to an approximation error ε? That is, can we realize the following transformation?

$$\rho \otimes \underbrace{\psi \otimes \cdots \otimes \psi}_{n \text{ times}} \overset{\varepsilon}{\approx\!\longrightarrow} e^{\mathcal{L}t}(\rho). \tag{1.5}$$

It is worth noting that Wave Matrix Lindbladization can be seen as a natural analogue to *Density Matrix Exponentiation* [LMR14]. Density Matrix Exponentiation is a well known protocol for Hamiltonian simulation, and it is also used in the context of quantum machine learning [BWP+16]. The task here is to implement a unitary $e^{-i\rho t}$ given multiple copies of an unknown quantum state ρ, so that the quantum state ρ serves as a Hamiltonian in this case. Similarly, in Wave Matrix Lindbladization, given multiple copies of an unknown quantum state ψ encoding an operator L, the task is to "lindbladize" this operator, i.e., implement the transformation given by (1.5).

1.2.2 Summary of Main Results

In this chapter, we propose a quantum algorithm that implements the quantum channel $e^{\mathcal{L}t}$ with some desired accuracy ε, where $0 < \varepsilon < 1$. We then investigate the sample complexity used by our algorithm (see Section 1.3.1). By sample complexity, we mean the number of copies of the program state ψ used to achieve the above task. Furthermore, we extend the single Lindblad operator case to the case in which the Lindblad master equation also consists of a Hamiltonian term (see Section 1.3.2), and we propose a quantum algorithm for this case as well.

Key Idea — Our quantum algorithm primarily involves two steps, which we repeat for each copy of the program state ψ. Suppose that ρ is in register 1 and the program state ψ is in registers 2 and 3. In the first step, we evolve ρ and ψ according to the following Lindbladian $\mathcal{M}$ for a short duration of time $\Delta := t/n$:

$$\mathcal{M}(\rho \otimes \psi) := M(\rho \otimes \psi)M^\dagger - \frac{1}{2}\left\{M^\dagger M, \rho \otimes \psi\right\}, \tag{1.6}$$

where M is a Lindblad operator defined as

$$M := \frac{1}{\sqrt{d}} (I_1 \otimes |\Gamma\rangle\langle\Gamma|_{23}) (\text{SWAP}_{12} \otimes I_3). \tag{1.7}$$

Here, SWAP_{12} is the swap operation that swaps the states in registers 1 and 2, and $|\Gamma\rangle$ is the maximally entangled vector. We explicitly define them both later in (1.15) and (1.16), respectively. Note that by evolving according to (1.6) for time Δ, this leads to the application of the quantum channel $e^{\mathcal{M}\Delta}$. Furthermore, the second step of our algorithm involves tracing out the program state ψ. To see this algorithm pictorially, please refer to Figure 1.1.

Formally, we can write the above two steps as

$$\text{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho \otimes \psi)\right] = \rho + \text{Tr}_{23}[\mathcal{M}(\rho \otimes \psi)]\,\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2), \tag{1.8}$$

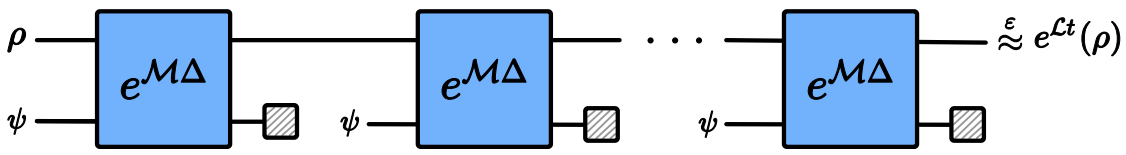


Figure 1.1: Our quantum algorithm, repeated $n = O(d^2t^2/\varepsilon)$ times, for approximating the target quantum channel, i.e., $e^{\mathcal{L}t}$, with an approximation error of ε. Each small, hatched square represents the trace-out operation. At the very left, the state ρ is in register 1, and the program state ψ is in registers 2 and 3, represented as a single line at this instance and thereafter for simplicity. If the Lindblad operator L is local, acting nontrivially on only a constant number of qubits, the same algorithm requires only $n = O(t^2/\varepsilon)$ repetitions, independent of the global system size d.

where the notation $\mathrm{Tr}_{23}[\cdot]$ is used to denote the action of tracing out registers 2 and 3, and $\|\cdot\|_\diamond$ denotes the diamond norm. The above equality follows from a Taylor series expansion, as discussed just after (1.2). Now, the critical step here is to prove that

$$\mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \psi)] = \mathcal{L}(\rho). \tag{1.9}$$

We prove this equality in detail in Section 1.3.1 and Appendix 1.A. The equality in (1.9) is important because it implies the following:

$$\mathrm{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho \otimes \psi)\right] = \rho + \mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \psi)]\,\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \tag{1.10}$$

$$= \rho + \mathcal{L}(\rho)\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \tag{1.11}$$

$$= e^{\mathcal{L}\Delta}(\rho) + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2). \tag{1.12}$$

It is clear that $\|\mathcal{M}\|_\diamond \leq 2d$ from the definition of $\mathcal{M}$ in (1.6) (we show this explicitly in the error analysis in Appendix 1.B). Now, if we repeat the two steps $n = O(d^2t^2/\varepsilon)$ times, then we can approximate the target channel, i.e., $e^{\mathcal{L}t}$, with an approximation error of $O(\varepsilon)$. We prove this statement in Theorem 1. This theorem in turn is a consequence of the key

lemma of this chapter (Lemma 7 in Appendix 1.A), which establishes the equality in (1.9), as well as the error analysis in Appendix 1.B.

When the Lindblad operator L is local. It is important to note that in most practical settings, the Lindblad operator L acts only on a constant number of qubits. Under this locality assumption, the effective dimension parameter d in (1.7) and in the error term above is replaced by the local subsystem dimension $d_A = O(1)$, where A is the subsystem on which L acts nontrivially. Consequently, the d^2 factor in the sample complexity disappears, and the algorithm requires only $n = O(t^2/\varepsilon)$ repetitions. Throughout this chapter, we first present results in the general form and highlight the local case explicitly where relevant.

1.2.3 Notation

We use the notation $\mathcal{H}_S$ to denote a d-dimensional Hilbert space associated with a quantum system S. We denote the set of quantum states acting on $\mathcal{H}_S$ by $\mathcal{D}(\mathcal{H}_S)$. Let $\mathrm{Tr}[X]$ denote the trace of a matrix X, i.e., the sum of its diagonal elements. Also, let $X^\dagger$ denote the Hermitian conjugate (or adjoint) of the matrix X. The Schatten p-norm of a matrix X is defined for $p \in [1, \infty)$ as follows:

$$\|X\|_p := \left(\mathrm{Tr}\left[\left(X^\dagger X\right)^{\frac{p}{2}}\right]\right)^{\frac{1}{p}}. \tag{1.13}$$

For the purpose of this chapter, we use Schatten norms with $p = 1$ (also called trace norm), $p = 2$ (Hilbert–Schmidt norm), and $p = \infty$ (operator norm). Furthermore, let $[X, Y] := XY - YX$ and $\{X, Y\} := XY + YX$ denote the commutator and anti-commutator of the operators X and Y, respectively.

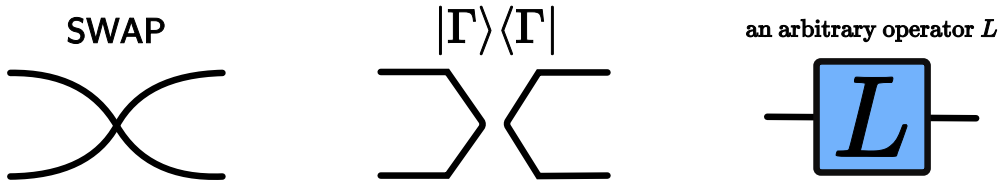


Figure 1.2: Tensor-network diagrams of operators SWAP, $|\Gamma\rangle\langle\Gamma|$, and L.

The diamond distance between two quantum channels $\mathcal{N}$ and $\mathcal{M}$ is defined as follows [Kit97]:

$$\|\mathcal{N} - \mathcal{M}\|_\diamond := \sup_{\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)} \|(\mathcal{I}_R \otimes \mathcal{N})(\rho) - (\mathcal{I}_R \otimes \mathcal{M})(\rho)\|_1 , \tag{1.14}$$

where R is a reference system and $\mathcal{I}_R$ is the identity channel acting on the system R. An important point to note here is that, in the above definition, the dimension of R is arbitrarily large. However, it is known that it suffices to perform the optimization over pure bipartite states with the dimension of R equal to the dimension of S. Furthermore, the quantity in the objective function of the above optimization is the trace distance, defined as $\|\rho - \sigma\|_1$ for two quantum states $\rho, \sigma \in \mathcal{D}(\mathcal{H}_S)$. In what follows, we employ the normalized diamond distance $\frac{1}{2}\|\mathcal{N} - \mathcal{M}\|_\diamond$ to measure approximation error—the normalization factor of $\frac{1}{2}$ guarantees that $\frac{1}{2}\|\mathcal{N} - \mathcal{M}\|_\diamond \in [0, 1]$ for quantum channels $\mathcal{N}$ and $\mathcal{M}$.

For ρ a quantum state in $\mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$, we denote the partial trace over the Hilbert space $\mathcal{H}_R$ by $\mathrm{Tr}_R[\rho]$. We also sometimes use a different notation for partial trace; i.e., given a multi-partite state ρ, we use the notation $\mathrm{Tr}_k[\rho]$ to denote the action of tracing out the k^{th} party. Furthermore, we define the maximally entangled vector in $\mathcal{H}_R \otimes \mathcal{H}_S$ as

$$|\Gamma\rangle_{RS} := \sum_i |i\rangle_R |i\rangle_S . \tag{1.15}$$

We also define the unitary swap operation in the following way:

$$\text{SWAP} := \sum_{i,j} |i\rangle\langle j| \otimes |j\rangle\langle i|. \tag{1.16}$$

In this chapter, we make extensive use of tensor-network diagrams. Figure 1.2 depicts tensor-network diagrams for some basic operators defined above, such as SWAP and $|\Gamma\rangle\langle\Gamma|$. For more background on tensor-network diagrams, please refer to [BB17]. Throughout this chapter, we sometimes suppress system labels for ease of notation; however, they will be clear from the context.

1.3 Quantum Algorithms for Simulating Markovian Dynamics

1.3.1 Single-Operator Case With No Hamiltonian Term

In this section, we provide a detailed analysis of our quantum algorithm for simulating the quantum channel $e^{\mathcal{L}t}$, in the case that the Lindbladian has only one Lindblad operator L, as in (1.3). The algorithm simulates the channel $e^{\mathcal{L}t}$ up to error ε in normalized diamond distance, using n copies of the program state ψ that encodes L (recall (1.4) here). Since we are interested in implementing the aforementioned channel with some desired accuracy in diamond distance, we assume that the channel input state is a joint quantum state of two systems rather than just one. Therefore, let $\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$ be an unknown quantum state given as input over the joint system RS, where the system R is a reference system. Furthermore, let the k^{th} copy of the program state ψ be a quantum state of the joint system $P_k Q_k$.

Algorithm 1 — Set $n \in \mathbb{N}$, with the particular choice specified later. Set $k = 1$. Given

the k^{th} copy of ψ, i.e., $\psi_{P_k Q_k}$, perform the following two steps:

1. Evolve the joint quantum state $\rho_{RS} \otimes \psi_{P_k Q_k}$ according to the dynamics realized by the following Lindbladian $\mathcal{M}$, for some small duration of time $\Delta = t/n$:

$$\mathcal{M}(\rho_{RS} \otimes \psi_{P_k Q_k}) := M(\rho_{RS} \otimes \psi_{P_k Q_k}) M^\dagger - \frac{1}{2}\left\{M^\dagger M, \rho_{RS} \otimes \psi_{P_k Q_k}\right\}. \tag{1.17}$$

   In the above, the Lindblad operator M acts on the joint system $RS P_k Q_k$, and we define it as

$$M := \frac{1}{\sqrt{d}} \left(I_{RS} \otimes |\Gamma\rangle\langle\Gamma|_{P_k Q_k}\right) \left(I_R \otimes \text{SWAP}_{S P_k} \otimes I_{Q_k}\right). \tag{1.18}$$

   (Note that we have redefined M as compared to (1.7), in order to include the trivial action on the reference system R.)

2. Trace out the systems $P_k Q_k$.

We repeat the above procedure using each copy of ψ, i.e., for all k ranging from 1 to n.

The following theorem states that the above algorithm uses $n = O(d^2 t^2/\varepsilon)$ copies of ψ to simulate the Lindbladian evolution of ρ_{RS}, given by (1.2)–(1.3), for time t, such that the final state is ε-close in normalized trace distance to the ideal target state $\left(\mathcal{I}_R \otimes e^{\mathcal{L}t}\right)(\rho_{RS})$, for an arbitrary input state ρ_{RS}.

Theorem 1 *Given access to n copies of the program state $\psi \in \mathcal{D}(\mathcal{H}_P \otimes \mathcal{H}_Q)$, which encodes the Lindblad operator L as in (1.4), there exists a quantum algorithm $\mathcal{A}$ such that the following error bound holds:*

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{A}\right\|_\diamond \leq \varepsilon, \tag{1.19}$$

with only $n = O(d^2 t^2/\varepsilon)$ copies of ψ, where d is the dimension of the system of interest, which is the system S in our case. In other words, $\mathcal{A}$ uses only $n = O(d^2 t^2/\varepsilon)$ copies of ψ to approximate the channel $e^{\mathcal{L}t}$ up to ε error in normalized diamond distance.

Proof. In what follows, we provide a brief sketch of the proof. For a more detailed version of the proof, please refer to Appendix 1.B. For ease of notation and simplicity, we refrain from writing the system labels, and we also assume that the input state ρ does not have the reference system R for the purpose of this proof sketch.

Let us begin by expanding the target state $e^{\mathcal{L}t}(\rho)$ using the following Taylor series expansion, as in (1.2), at the initial time of $t = 0$:

$$e^{\mathcal{L}t}(\rho) = \rho + \mathcal{L}(\rho)t + \frac{1}{2}(\mathcal{L} \circ \mathcal{L})(\rho)t^2 + \ldots. \tag{1.20}$$

In the first step of Algorithm 1, we simulate the Lindbladian evolution of $\rho \otimes \psi$, using the Lindbladian $\mathcal{M}$ in (1.17) for some small duration of time Δ, and then trace out ψ. The output state obtained after this step is

$$\mathrm{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho \otimes \psi)\right] = \rho + \mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \psi)]\,\Delta + O(\|\mathcal{M}\|_{\diamond}^2\,\Delta^2), \tag{1.21}$$

where ρ is in register 1 and ψ in registers 2 and 3, and we have again used the expansion in (1.2). Then writing out the second term on the right-hand side of the above equation and using the definition in (1.17), we find that

$$\mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \psi)] = \mathrm{Tr}_{23}\left[M(\rho \otimes \psi)M^\dagger\right] - \frac{1}{2}\,\mathrm{Tr}_{23}\left[M^\dagger M\,(\rho \otimes \psi)\right] - \frac{1}{2}\,\mathrm{Tr}_{23}\left[(\rho \otimes \psi)\,M^\dagger M\right]. \tag{1.22}$$

We then invoke Lemma 7 in Appendix 1.A to simplify each term on the right-hand side of the above equation. As a result of this, we obtain the following equalities:

$$\mathrm{Tr}_{23}\left[M(\rho \otimes \psi)M^\dagger\right] = L\rho L^\dagger, \tag{1.23}$$

$$\mathrm{Tr}_{23}\left[M^\dagger M\,(\rho \otimes \psi)\right] = L^\dagger L\rho, \tag{1.24}$$

$$\mathrm{Tr}_{23}\left[(\rho \otimes \psi)\,M^\dagger M\right] = \rho L^\dagger L. \tag{1.25}$$

For a graphical representation of the above simplifications, please refer to the tensor-network diagrams provided in Figures 1.3, 1.5, and 1.6.

Using the above equations along with (1.3), we rewrite (1.21) as

$$\mathrm{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho\otimes\psi)\right] = \rho + \mathcal{L}(\rho)\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \tag{1.26}$$

$$= e^{\mathcal{L}\Delta}(\rho) + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2). \tag{1.27}$$

Substituting $\Delta = t/n, \|\mathcal{M}\|_\diamond \leq 2d$, and repeating Algorithm 1 for $n = O(d^2t^2/\varepsilon)$ times produces a quantum state that is $O(\varepsilon)$-close to the ideal target state $e^{\mathcal{L}t}(\rho)$ in normalized trace distance. For a detailed error analysis of this claim, in terms of the normalized diamond distance, please refer to Appendix 1.B. ■

Remark 2 *We note here that the simulation is unchanged if we employ the following definition of M, instead of that given in (1.18):*

$$M \coloneqq (I_{RS}\otimes|\varphi\rangle\!\langle\Gamma|_{P_kQ_k})\,(I_R\otimes \mathrm{SWAP}_{SP_k}\otimes I_{Q_k})\,. \tag{1.28}$$

where $|\varphi\rangle$ is an arbitrary bipartite state vector. As such, the choice of $|\varphi\rangle$ in (1.18) amounts to the maximally entangled state $\frac{1}{\sqrt{d}}|\Gamma\rangle$. The claim here can be checked by examining Figures 1.3, 1.4, 1.5, and 1.6, as well as the proof in Appendix 1.A.

When Lindblad Operator is Local

As mentioned before in the Introduction section, in most practical settings, the Lindblad operator L acts only on a constant number of qubits. Let the system S decompose as $S = A\bar{A}$, where A is the constant-size subsystem on which L acts nontrivially and $\bar{A}$ is the subsystem of the remaining qubits. We write

$$L \equiv L_A \otimes I_{\bar{A}}, \qquad d_A \coloneqq \dim(\mathcal{H}_A). \tag{1.29}$$

Now, because the Lindblad operator is local, we perform the fixed Lindbladian evolution (i.e., Step 1 of Algorithm 1) using a localized Lindblad operator that acts only on the subsystem A and the program registers. Let us denote the program registers for the k^{th} iteration of Algorithm 1 by $P_k^{(A)}$ and $Q_k^{(A)}$, and define the maximally entangled vector by

$$|\Gamma_A\rangle := \sum_{j=1}^{d_A} |j\rangle \otimes |j\rangle. \tag{1.30}$$

The localized Lindblad operator is

$$M_A := \frac{1}{\sqrt{d_A}} \Big(I_{RS} \otimes |\Gamma_A\rangle\langle\Gamma_A|_{P_k^{(A)} Q_k^{(A)}} \Big) \Big(I_{R\bar{A}} \otimes \mathrm{SWAP}_{AP_k^{(A)}} \otimes I_{Q_k^{(A)}} \Big), \tag{1.31}$$

and the corresponding fixed Lindbladian is

$$\mathcal{M}_A(X) := M_A X M_A^\dagger - \frac{1}{2} \left\{ M_A^\dagger M_A,\, X \right\}. \tag{1.32}$$

As in the global L case, one step of duration $\Delta = t/n$ evolves $\rho \otimes \psi$ by $e^{\mathcal{M}_A \Delta}$ and then we trace out the program registers $P_k^{(A)} Q_k^{(A)}$.

Theorem 3 (Local Lindblad operator, no Hamiltonian) *Suppose L acts on a constant-size subsystem A so that $L = L_A \otimes I_{\bar{A}}$ with $d_A = \dim(\mathcal{H}_A) = O(1)$ and $\bar{A}$ is the subsystem of the remaining qubits. Given access to n copies of a program state $\psi \in \mathcal{D}(\mathcal{H}_{P^{(A)}} \otimes \mathcal{H}_{Q^{(A)}})$ that encodes L_A as in (1.4) (with d replaced by d_A), there exists a quantum algorithm $\mathcal{A}_{\text{loc}}$ such that*

$$\frac{1}{2} \left\| e^{\mathcal{L}t} - \mathcal{A}_{\text{loc}} \right\|_\diamond \le \varepsilon, \tag{1.33}$$

using only

$$n \;=\; O\!\left(d_A^2 \, \frac{t^2}{\varepsilon} \right) \;=\; O\!\left(\frac{t^2}{\varepsilon} \right) \tag{1.34}$$

copies of ψ, where the hidden constant depends only on d_A (and thus is independent of the global system size).

Proof. The proof is identical in structure to Theorem 1, with all occurrences of the system S restricted to the subsystem A and with d replaced by d_A. Concretely, one can just repeat the analysis using M_A from (1.31) in place of M; the key contractions (established via Lemma 7) now yield

$$\mathrm{Tr}_{P_k^{(A)} Q_k^{(A)}} \left[M_A(\rho \otimes \psi) M_A^\dagger \right] = (L_A \otimes I_{\bar{A}})\, \rho\, (L_A^\dagger \otimes I_{\bar{A}}), \tag{1.35}$$

$$\mathrm{Tr}_{P_k^{(A)} Q_k^{(A)}} \left[M_A^\dagger M_A(\rho \otimes \psi) \right] = (L_A^\dagger L_A \otimes I_{\bar{A}})\, \rho, \tag{1.36}$$

$$\mathrm{Tr}_{P_k^{(A)} Q_k^{(A)}} \left[(\rho \otimes \psi) M_A^\dagger M_A \right] = \rho\, (L_A^\dagger L_A \otimes I_{\bar{A}}), \tag{1.37}$$

so that $\mathrm{Tr}_{P_k^{(A)} Q_k^{(A)}}[\mathcal{M}_A(\rho \otimes \psi)] = \mathcal{L}(\rho)$ with the same $\mathcal{L}$ as in (1.3). The per-step truncation error is $O(\|\mathcal{M}_A\|_\diamond^2\, \Delta^2)$; with $\|\mathcal{M}_A\|_\diamond \le 2 d_A$, this is $O\left(d_A^2 \Delta^2\right)$. Choosing $n = O(d_A^2 t^2/\varepsilon)$ makes the total error $O(\varepsilon)$. Since $d_A = O(1)$, this simplifies to $n = O(t^2/\varepsilon)$. ∎

Remark 4 *All tensor-network diagrams in Figures 1.3–1.6 continue to apply verbatim after relabeling the system wire as A; wires corresponding to $\bar{A}$ and R simply pass through as identities.*

1.3.2 Single Lindblad Operator Case with Hamiltonian Term

In this section, we consider the case of simulating a Lindbladian evolution, where the Lindbladian consists of a single Lindblad operator L and a Hamiltonian H. To be more precise, in this section, we are interested in simulating the Lindbladian dynamics of a quantum state ρ for time t according to the following Lindbladian:

$$\mathcal{L}(\rho) := -i[H, \rho] + L \rho L^\dagger - \frac{1}{2} \left\{ L^\dagger L, \rho \right\}. \tag{1.38}$$

To begin with, we will look at how to encode the operators L and H into quantum states. We suppose that L is encoded into a pure quantum state ψ in the same way that was

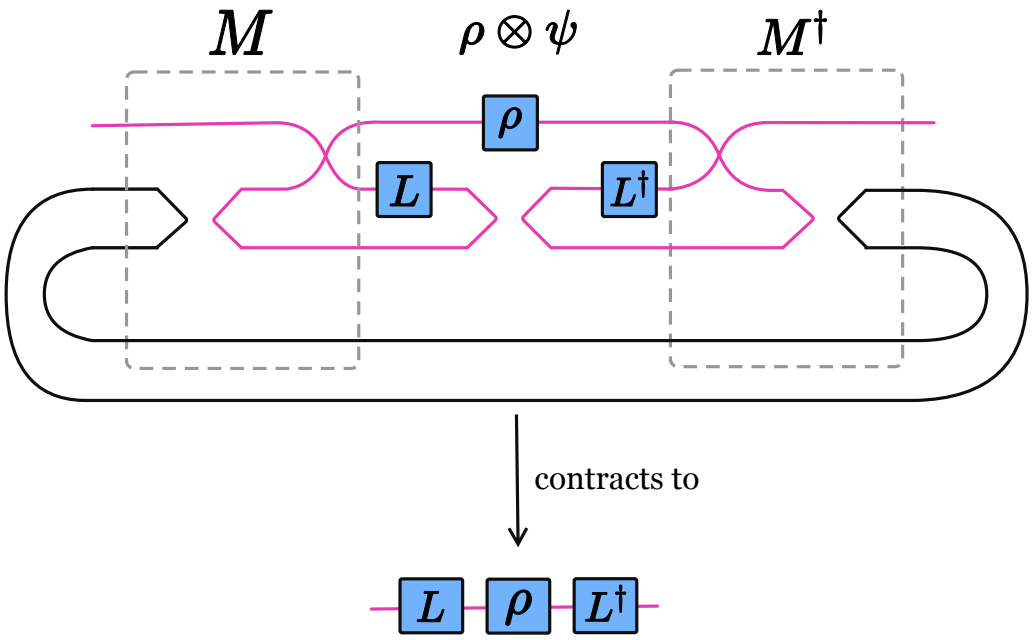


Figure 1.3: Tensor-network diagram of $\mathrm{Tr}_{23}\left[M(\rho\otimes\psi)M^\dagger\right]$. The whole network on the top contracts to the network on the bottom due to the partial trace operation over the 2nd and 3rd systems. Simply follow the pink line from the left of the first system to its right to observe this.

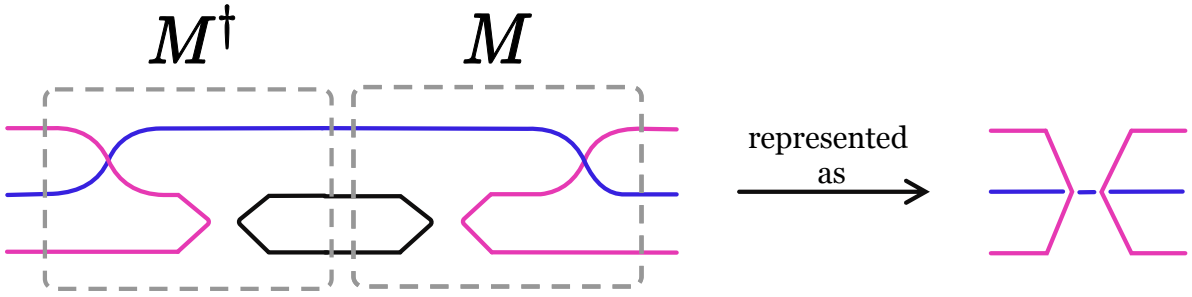


Figure 1.4: Tensor-network diagram of the operator $M^\dagger M$. This network will be used as a subroutine in the following figures. As a result, for the sake of brevity, we use the figure on the right to represent the figure on the left.

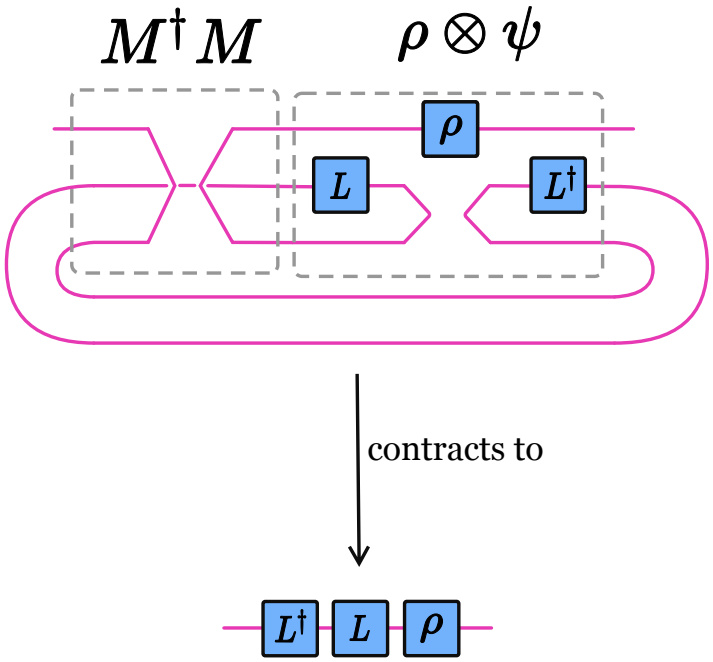


Figure 1.5: Tensor-network diagram of $\mathrm{Tr}_{23}\left[M^\dagger M\,(\rho\otimes\psi)\right]$. Please refer to Figure 1.4 to understand the tensor-network diagram of $M^\dagger M$. The whole network on the top contracts to the network on the bottom due to the partial trace operation over the 2nd and 3rd systems. Simply follow the pink line from left of the first system to its right to observe this.

considered previously (see (1.4)). On the other hand, we suppose that the Hamiltonian H is encoded into the density matrix of a quantum state σ. This type of encoding was first considered in [LMR14] for density matrix exponentiation, and further discussions of it are given in [KLL+17, Eqs. (1)–(2)]. Overall, the program state that encodes H and L is the following tensor-product state:

$$\omega := \sigma \otimes \psi, \tag{1.39}$$

and the Lindbladian to be simulated, given by (1.38), can now be rewritten as follows:

$$\mathcal{L}(\rho) := -i[\sigma,\rho] + L\rho L^\dagger - \frac{1}{2}\left\{L^\dagger L,\rho\right\}. \tag{1.40}$$

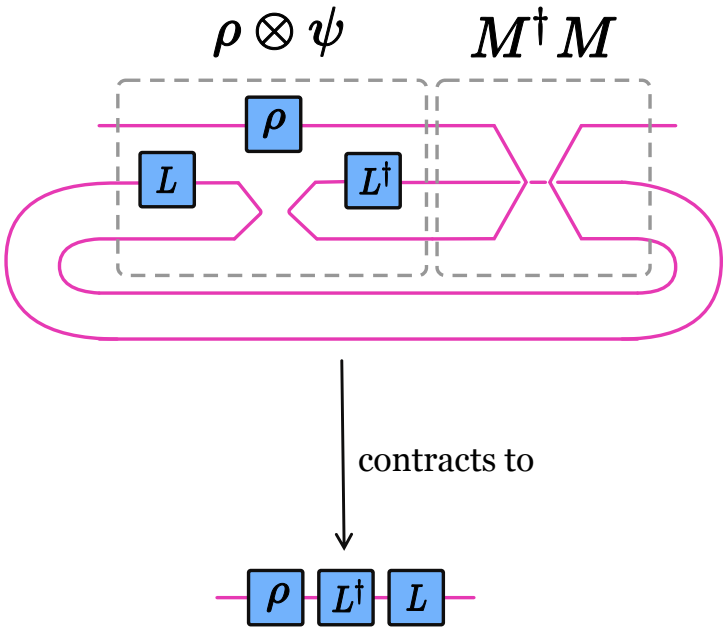


Figure 1.6: Tensor-network diagram of $\mathrm{Tr}_{23}\left[(\rho \otimes \psi)\, M^\dagger M\right]$. Please refer to Figure 1.4 to understand the tensor network diagram of $M^\dagger M$. The whole network on the top contracts to the network on the bottom due to the partial trace operation over the 2nd and 3rd systems. Simply follow the pink line from the left of the first system to its right to observe this.

We are now in a position to propose a quantum algorithm for simulating the quantum channel $e^{\mathcal{L}t}$, corresponding to the Lindbladian in (1.40), up to error ε in diamond distance, using n copies of the program state ω. As before, for providing an analysis related to the diamond distance, let $\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$ be an unknown quantum state given as input over the joint system RS, where the system R acts as a reference system. Furthermore, let the k^{th} copy of the program state ω be a quantum state of a joint system $H_k P_k Q_k$, where $\sigma \in \mathcal{D}(\mathcal{H}_{H_k})$ and $\psi \in \mathcal{D}(\mathcal{H}_{P_k} \otimes \mathcal{H}_{Q_k})$. For brevity, let us use $(HPQ)_k$ as a shorthand for $H_k P_k Q_k$.

Algorithm 2 — Set $n \in \mathbb{N}$, with a particular choice specified later. Set $k = 1$. Given the k^{th} copy of ω, i.e., $\omega_{(HPQ)_k}$, perform the following two steps:

1. Evolve the joint quantum state $\rho_{RS} \otimes \omega_{(HPQ)_k}$ according to the dynamics realized by the following Lindbladian $\mathcal{M}$, for some small duration of time $\Delta = t/n$:

$$\mathcal{M}(\rho_{RS} \otimes \omega_{(HPQ)_k}) := -i[\hat{H}, \rho_{RS} \otimes \omega_{(HPQ)_k}] + M(\rho_{RS} \otimes \omega_{(HPQ)_k})M^\dagger - \frac{1}{2}\left\{M^\dagger M, \rho_{RS} \otimes \omega_{(HPQ)_k}\right\}. \quad (1.41)$$

   The Hamiltonian $\hat{H}$ and Lindblad operator M act on the joint system $RS(HPQ)_k$, and we define them as

$$\hat{H} := (\mathrm{SWAP}_{SH_k} \otimes I), \quad (1.42)$$

$$M := \frac{1}{\sqrt{d}} (I \otimes |\Gamma\rangle\langle\Gamma|_{P_k Q_k}) (\mathrm{SWAP}_{SP_k} \otimes I). \quad (1.43)$$

   Here, we apply the identity operator I on all those systems that are not explicitly mentioned.

2. Trace out the program states, i.e., the systems $(HPQ)_k$.

We repeat the above procedure for each copy of ω, i.e., for all k ranging from 1 to n. As before, we have some flexibility in choosing M, as mentioned in Remark 2.

The following theorem states that the above algorithm uses $n = O(d^2t^2/\varepsilon)$ copies of ω to simulate the Lindbladian evolution of ρ_{RS}, according to the Lindbladian in (1.40), for time t, and the resulting state is ε-close in normalized trace distance to the target state $(\mathcal{I}_R \otimes e^{\mathcal{L}t})(\rho_{RS})$, for every input state ρ_{RS}.

Theorem 5 *Given access to n copies of the program state $\omega \in \mathcal{D}(\mathcal{H}_{HPQ})$, which is defined in (1.39) and encodes the Lindblad operator L and the Hamiltonian H, there exists a quantum algorithm $\mathcal{A}$ such that the following holds:*

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{A}\right\|_\diamond \leq \varepsilon, \quad (1.44)$$

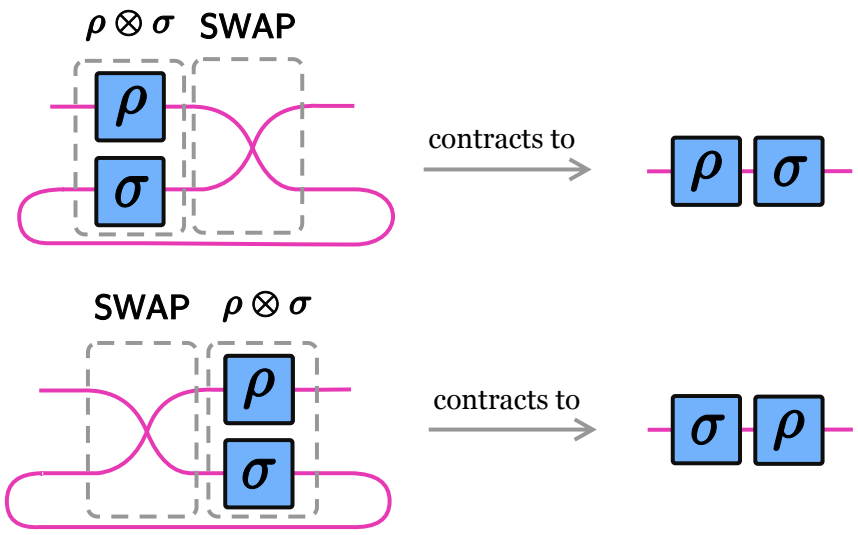


Figure 1.7: Tensor-network diagrams of $\mathrm{Tr}_2\left[\mathrm{SWAP}(\rho\otimes\sigma)\right]$ and $\mathrm{Tr}_2\left[(\rho\otimes\sigma)\,\mathrm{SWAP}\right]$. The networks on the left contract to the networks on the right due to the partial trace operation over the 2nd systems. Simply follow the pink line from the left of the first system to its right to observe this.

with only $n = O(d^2t^2/\varepsilon)$ copies of ω, where d is the dimension of the system of interest, which is the system S in our case. In other words, $\mathcal{A}$ uses only $n = O(d^2t^2/\varepsilon)$ copies of ω to approximate the channel $e^{\mathcal{L}t}$, defined from (1.40) and (1.2), up to ε error in diamond distance.

Proof. The proof style is very similar to that of Theorem 1. For clarity, we remove the system labels here as well.

In the first step of Algorithm 2, we simulate the Lindbladian evolution of $\rho\otimes\omega$, given by (1.40), with the Lindbladian $\mathcal{M}$ in (1.41) for some small duration of time Δ, and then trace out ω. The output state obtained after this step is

$$\mathrm{Tr}_{234}\left[e^{\mathcal{M}\Delta}(\rho\otimes\omega)\right] = \rho + \mathrm{Tr}_{234}\left[\mathcal{M}(\rho\otimes\omega)\right]\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2), \tag{1.45}$$

where we have used the expansion in (1.2). Next we rewrite the second term in the fol-

lowing way:

$$\mathrm{Tr}_{234}[\mathcal{M}(\rho\otimes\omega)] = -i\,\mathrm{Tr}_{234}\Big[[\hat{H},\rho\otimes\omega]\Big] + \mathrm{Tr}_{234}\Big[M(\rho\otimes\omega)M^\dagger\Big] \\ -\frac{1}{2}\,\mathrm{Tr}_{234}\Big[M^\dagger M\,(\rho\otimes\omega)\Big] - \frac{1}{2}\,\mathrm{Tr}_{234}\Big[(\rho\otimes\omega)\,M^\dagger M\Big]. \quad (1.46)$$

Observe that the operator M can be written as $M = M'\otimes I_2$, where I_2 is the identity acting on the second register. From (1.42) and (1.43) and the fact that $\omega = \sigma\otimes\psi$, we simplify the right-hand side of the above equation:

$$\mathrm{Tr}_{234}[\mathcal{M}(\rho\otimes\phi)] = -i\,\mathrm{Tr}_{2}[[\mathrm{SWAP},\rho\otimes\sigma]] + \mathrm{Tr}_{34}\Big[M'(\rho\otimes\psi)M'^\dagger\Big] \\ -\frac{1}{2}\,\mathrm{Tr}_{34}\Big[M'^\dagger M'\,(\rho\otimes\psi)\Big] - \frac{1}{2}\,\mathrm{Tr}_{34}\Big[(\rho\otimes\psi)\,M'^\dagger M'\Big]. \quad (1.47)$$

We simplify the first term even more by employing its related tensor-network diagrams, as illustrated in Figure 1.7. As before, please refer to Figures 1.3, 1.5, and 1.6 to simplify the remaining terms (please keep in mind that M is simply M' when referring to these figures). We finally obtain the following equalities:

$$-i\,\mathrm{Tr}_{2}[[\mathrm{SWAP},\rho\otimes\sigma)]] = -i[\sigma,\rho], \quad (1.48)$$

$$\mathrm{Tr}_{34}\Big[M'(\rho\otimes\psi)M'^\dagger\Big] = L\rho L^\dagger, \quad (1.49)$$

$$\mathrm{Tr}_{34}\Big[M'^\dagger M'\,(\rho\otimes\psi)\Big] = L^\dagger L\rho, \quad (1.50)$$

$$\mathrm{Tr}_{34}\Big[(\rho\otimes\psi)\,M'^\dagger M'\Big] = \rho L^\dagger L. \quad (1.51)$$

From the above equations and (1.40), we rewrite (1.45) as

$$\mathrm{Tr}_{234}\Big[e^{\mathcal{M}\Delta}(\rho\otimes\omega)\Big] = \rho + \mathrm{Tr}_{234}[\mathcal{M}(\rho\otimes\omega)]\,\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \quad (1.52)$$

$$= \rho + \mathcal{L}(\rho)\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \quad (1.53)$$

$$= e^{\mathcal{L}\Delta}(\rho) + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2). \quad (1.54)$$

Substituting $\Delta = t/n, \|\mathcal{M}\|_\diamond \le 2 + 2d$, and repeating Algorithm 2 for $n = O(d^2t^2/\varepsilon)$ times

produces a quantum state that is $O(\varepsilon)$-close to the target state $e^{\mathcal{L}t}(\rho)$. A detailed error analysis of this claim goes along the lines of that provided in Appendix 1.B. ■

When the Lindblad Operator is Local

As in Section 1.3.1, decompose $S = A\bar{A}$ and assume the single Lindblad operator is local:

$$L = L_A \otimes I_{\bar{A}}, \qquad d_A := \dim(\mathcal{H}_A). \tag{1.55}$$

We localize only the dissipative part by replacing M in (1.41) with the localized Lindblad operator M_A from (1.31), which acts nontrivially on $AP_k^{(A)}Q_k^{(A)}$ and trivially elsewhere. The Hamiltonian swap $\hat{H} = \mathrm{SWAP}_{SH_k} \otimes I$ in (1.42) is left unchanged, and the program state remains $\omega = \sigma \otimes \psi$, with ψ encoding L_A as in (1.4) (interpreting d as d_A).

Theorem 6 (Local Lindblad operator, arbitrary Hamiltonian) *Suppose the Lindblad operator L acts on a constant-size subsystem A so that $L = L_A \otimes I_{\bar{A}}$ with $d_A = \dim(\mathcal{H}_A) = O(1)$ and $\bar{A}$ is the subsystem of the remaining qubits and let H be arbitrary (encoded via σ as in (1.39)). Given n copies of $\omega = \sigma \otimes \psi$, there exists a quantum algorithm $\mathcal{A}_{\mathrm{loc}}$ such that*

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{A}_{\mathrm{loc}}\right\|_\diamond \le \varepsilon, \tag{1.56}$$

using

$$n = O\left(\frac{t^2}{\varepsilon}\right), \tag{1.57}$$

where the hidden constant depends on d_A (and not on system size d).

Proof. The argument is the same as in Theorem 5, with M replaced by M_A from (1.31). The Hamiltonian term proceeds identically because $\hat{H}$ is unchanged; the three dissipative contractions reduce exactly as before but on the subsystem A, yielding $-i[\sigma, \rho]$ and the

L-terms with $L_A \otimes I_{\bar{A}}$. The per-step truncation remains $O(\Delta^2)$ (constants depending on d_A), so choosing $n = O(t^2/\varepsilon)$ achieves total error $O(\varepsilon)$. ■

1.4 Conclusion and Open Problems

In this chapter, we proposed a quantum algorithm for approximately simulating Lindblad evolution to arbitrary accuracy. For the purpose of this chapter, we considered a simple case in which the Lindbladian consists of only one Lindblad operator, as this case can easily be extended to the more general case with multiple Lindblad operators. We further investigated the sample complexity of our algorithm for this case, i.e., the number of samples of the program state needed by our algorithm to achieve the desired accuracy. We then extended the single-operator case to include a Hamiltonian term in the Lindbladian, and we proposed a quantum algorithm for this case as well.

Here we list some directions for future work:

- Is there is an efficient implementation of Step 1 of Algorithm 1? Solving this problem will undoubtedly resolve the time or gate complexities of our algorithms.
- Another direction is to provide an extension to more complex cases, in which a Lindblad operator can be expressed as a linear combination or a polynomial of the operators encoded in the program states. This direction was considered in [KLL+17, Section 5] in the case of density matrix exponentiation / sample-based Hamiltonian simulation.
- In [KLL+17], the authors investigated a modified version of the Hamiltonian simulation problem, which they called sample-based Hamiltonian simulation. In this

problem, given an unknown quantum state ρ and n copies of the program state σ, the task is to implement the following transformation:

$$\rho \otimes \underbrace{\sigma \otimes \cdots \otimes \sigma}_{n \text{ times}} \overset{\varepsilon}{\approx\!\longrightarrow} e^{-i\sigma t} \rho e^{i\sigma t}. \tag{1.58}$$

The authors reported in [KLL+17, Theorem 5] that a quantum algorithm for this task needs at least $\Omega(t^2/\varepsilon)$ copies of σ to simulate the above channel within ε accuracy. It is an important open question to determine the sample complexity of the general task of sample-based Lindbladian simulation, which could help determine if Algorithm 1 has optimal sample complexity.

- Furthermore, one can investigate the number of samples needed to perform state tomography of the program state and approximately recover the entire Lindblad operator encoded in this state. Then, we can simply use this operator to approximately simulate the corresponding Lindbladian evolution. One crucial question here is to ask if the sample complexity needed for state tomography is larger than the sample complexity of wave matrix Lindbladization. If this turns out to be the case, then it implies that a party can send sufficiently many copies of the program state to another party to simulate the respective Lindbladian evolution without revealing anything about the encoded Lindblad operator. One can think of this as some instance of quantum copy-protection, as introduced in [Aar09], in which the quantum operation is the Lindbladian evolution of a quantum state. This may also address the question raised in [KLL+17], which asks whether there is a quantum operation other than Hamiltonian simulation that can be encoded in quantum states and executed without revealing much about the quantum operation itself.

BIBLIOGRAPHY

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242, Paris, France, July 2009. doi:10.1109/CCC.2009.42.

[BB17] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell. August 2017. arXiv:1708.00006.

[BCC+13] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 283–292, December 2013. doi:10.1145/2591796.2591854.

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, December 2014. doi:10.1103/PhysRevLett.114.090502.

[BCG14] Dominic W. Berry, Richard Cleve, and Sevag Gharibian. Gate-efficient discrete simulations of continuous-time quantum query algorithms. *Quantum Information and Computation*, 14(1–2):1–30, November 2014.

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, October 2015. doi:10.1109/FOCS.2015.54.

[BP02] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002.

[BWP+16] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, November 2016. doi:10.1038/nature23474.

[CL16] Andrew M. Childs and Tongyang Li. Efficient simulation of sparse Markovian quantum dynamics. *Quantum Information and Computation*, 17(11&12):901–947, November 2016. doi:10.26421/QIC17.11-12.

[CW17] Richard Cleve and Chunhao Wang. Efficient quantum algorithms for simulating Lindblad evolution. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.17.

[Fey82] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982. doi:10.1007/BF02650179/METRICS.

[GAN14] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153–185, March 2014. doi:10.1103/RevModPhys.86.153.

[GKP+25] Byeongseon Go, Hyukjoon Kwon, Siheon Park, Dhrumil Patel, and Mark M. Wilde. Sample-based Hamiltonian and Lindbladian simulation: Non-asymptotic analysis of sample complexity, 2025. arXiv:2412.02134.

[GKS08] Vittorio Gorini, Andrzej Kossakowski, and E. C. G. Sudarshan. Completely positive dynamical semigroups of N-level systems. *Journal of Mathematical Physics*, 17(5):821–825, August 2008. doi:10.1063/1.522979.

[GZ04] Crispin Gardiner and Peter Zoller. *Quantum Noise: A Handbook of Markovian and Non-Markovian Quantum Stochastic Methods with Applications to Quantum Optics*. Springer, 2004.

[KB14] Michael J. Kastoryano and Fernando G. S. L. Brandão. Quantum Gibbs samplers: the commuting case. *Communications in Mathematical Physics*, 344(3):915–957, September 2014. doi:10.1007/s00220-016-2641-8.

[KBD+08] B. Kraus, H. P. Büchler, S. Diehl, A. Kantian, A. Micheli, and P. Zoller. Preparation of entangled states by quantum Markov processes. *Physical Review A*, 78(4):042307, October 2008. doi:10.1103/PhysRevA.78.042307.

[Kit97] A. Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. doi:10.1070/RM1997v052n06ABEH002155.

[KLL+17] Shelby Kimmel, Cedric Yen Yu Lin, Guang Hao Low, Maris Ozols, and

Theodore J. Yoder. Hamiltonian simulation with optimal sample complexity. *npj Quantum Information*, 3(1):1–7, March 2017. doi:10.1038/s41534-017-0013-7.

[KRS11] M. J. Kastoryano, F. Reiter, and A. S. Sørensen. Dissipative preparation of entanglement in optical cavities. *Physical Review Letters*, 106(9):090502, February 2011. doi:10.1103/PhysRevLett.106.090502.

[KSMM22] Hirsh Kamakari, Shi-Ning Sun, Mario Motta, and Austin J. Minnich. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. *PRX Quantum*, 3(1):010320, February 2022. doi:10.1103/PRXQuantum.3.010320.

[LC17] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, January 2017. doi:10.1103/PhysRevLett.118.010501.

[Lin76] Göran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, June 1976. doi:10.1007/BF01608499/METRICS.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, August 1996. doi:10.1126/science.273.5278.1073.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, July 2014. doi:10.1038/nphys3029.

[MK08] Volkhard May and Oliver Kühn. *Charge and Energy Transfer Dynamics in Molecular Systems*. John Wiley & Sons, 2008.

[MOTT22] Alexander Miessen, Pauline J. Ollitrault, Francesco Tacchino, and Ivano Tavernelli. Quantum algorithms for quantum dynamics. *Nature Computational Science*, 3(1):25–37, December 2022. doi:10.1038/s43588-022-00374-2.

[MPGC13] Easwar Magesan, Daniel Puzzuoli, Christopher E. Granade, and David G. Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Physical Review A*, 87(1):012324, January 2013. doi:10.1103/PhysRevA.87.012324.

[MTAB12] Daniel Manzano, Markus Tiersch, Ali Asadian, and Hans J. Briegel. Quantum transport efficiency and Fourier's law. *Physical Review E*, 86(6):061118, December 2012. doi:10.1103/PhysRevE.86.061118.

[MZB23] Gerard McCaul, Dmitry V. Zhdanov, and Denys I. Bondar. The wave operator representation of quantum and classical dynamics. February 2023. arXiv:2302.13208.

[Nit06] Abraham Nitzan. *Chemical Dynamics in Condensed Phases: Relaxation, Transfer and Reactions in Condensed Molecular Systems*. Oxford University Press, 2006.

[OLG12] Beatriz Olmos, Igor Lesanovsky, and Juan P. Garrahan. Facilitated spin models of dissipative quantum glasses. *Physical Review Letters*, 109(2):020403, July 2012. doi:10.1103/PHYSREVLETT.109.020403/FIGURES/5/MEDIUM.

[PK98] Martin B. Plenio and Peter L. Knight. The quantum-jump approach to dissipative dynamics in quantum optics. *Reviews of Modern Physics*, 70(1):101–144, January 1998. doi:10.1103/RevModPhys.70.101.

[Pro11] Tomaz Prosen. Open *XXZ* spin chain: Nonequilibrium steady state and a strict bound on ballistic transport. *Physical Review Letters*, 106(21):217206, May 2011. doi:10.1103/PhysRevLett.106.217206.

[PW23] Dhrumil Patel and Mark M. Wilde. Wave matrix Lindbladization I: Quantum programs for simulating Markovian dynamics. *Open Systems and Information Dynamics*, 30(02):2350010, June 2023. doi:10.1142/s1230161223500105.

[RRS16] Florentin Reiter, David Reeb, and Anders S. Sørensen. Scalable dissipative preparation of many-body entanglement. *Physical Review Letters*, 117(4):040501, July 2016. doi:10.1103/PhysRevLett.117.040501.

[SBH+23] Nishchay Suri, Joseph Barreto, Stuart Hadfield, Nathan Wiebe, Filip Wudarski, and Jeffrey Marshall. Two-unitary decomposition algorithm and open quantum system simulation. *Quantum*, 7:1002, May 2023. doi:10.22331/q-2023-05-15-1002.

[SHMS+22] Anthony W. Schlimgen, Kade Head-Marsden, LeeAnn M. Sager, Prineha Narang, and David A. Mazziotti. Quantum simulation of the lindblad equation using a unitary decomposition of operators. *Physical Review Research*, 4(2):023216, June 2022. doi:10.1103/PhysRevResearch.4.023216.

[VWC09] Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5(9):633–636, 2009.

[Wei21] Ulrich Weiss. *Quantum Dissipative Systems*. World Scientific, 5th edition, 2021. arXiv:https://www.worldscientific.com/doi/pdf/10.1142/8334, doi:10.1142/8334.

APPENDIX

1.A Proof of the Key Lemma

Lemma 7 *Let $\mathcal{H}_1, \mathcal{H}_2$, and $\mathcal{H}_3$ be d-dimensional Hilbert spaces. Let M be a linear operator acting on $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3$ and defined as*

$$M_{123} \coloneqq \frac{1}{\sqrt{d}} \left(I_1 \otimes |\Gamma\rangle\langle\Gamma|_{23}\right) \left(\mathrm{SWAP}_{12} \otimes I_3\right). \tag{1.A.1}$$

Also, let ρ be a quantum state in $\mathcal{D}(\mathcal{H}_1)$, and let $|\psi^L\rangle$ be a pure quantum state in $\mathcal{H}_2 \otimes \mathcal{H}_3$ defined as $|\psi^L\rangle_{23} \coloneqq (L_2 \otimes I_3)\,|\Gamma\rangle_{23}$, where L is a $d \times d$-dimensional linear operator such that $\|L\|_2 = 1$. Then the following identities hold:

$$\mathrm{Tr}_{23}[M_{123} \left(\rho_1 \otimes \psi^L_{23}\right) M^\dagger_{123}] = \left[L\rho L^\dagger\right]_1, \tag{1.A.2}$$

$$\mathrm{Tr}_{23}[M^\dagger_{123} M_{123} \left(\rho_1 \otimes \psi^L_{23}\right)] = \left[L^\dagger L\rho\right]_1, \tag{1.A.3}$$

$$\mathrm{Tr}_{23}[\left(\rho_1 \otimes \psi^L_{23}\right) M^\dagger_{123} M_{123}] = \left[\rho L^\dagger L\right]_1, \tag{1.A.4}$$

where we have used the shorthand $\psi^L_{23} \equiv |\psi^L\rangle\langle\psi^L|_{23}$.

Proof. Recall that

$$|\Gamma\rangle\langle\Gamma|_{23} = \sum_{i,j} |i\rangle\langle j|_2 \otimes |i\rangle\langle j|_3, \tag{1.A.5}$$

$$\mathrm{SWAP}_{12} = \sum_{k,\ell} |k\rangle\langle\ell|_1 \otimes |\ell\rangle\langle k|_2. \tag{1.A.6}$$

Using the above equalities, observe that

$$M_{123} = \frac{1}{\sqrt{d}} \left(I_1 \otimes |\Gamma\rangle\langle\Gamma|_{23}\right) \left(\mathrm{SWAP}_{12} \otimes I_3\right) \tag{1.A.7}$$

$$= \frac{1}{\sqrt{d}} \left(I_1 \otimes \sum_{i,j} |i\rangle\langle j|_2 \otimes |i\rangle\langle j|_3\right) \left(\sum_{k,\ell} |k\rangle\langle\ell|_1 \otimes |\ell\rangle\langle k|_2 \otimes I_3\right) \tag{1.A.8}$$

$$= \frac{1}{\sqrt{d}} \sum_{i,j,k,\ell} (I_1 \otimes |i\rangle\langle j|_2 \otimes |i\rangle\langle j|_3)\, (|k\rangle\langle \ell|_1 \otimes |\ell\rangle\langle k|_2 \otimes I_3) \tag{1.A.9}$$

$$= \frac{1}{\sqrt{d}} \sum_{i,j,k,\ell} |k\rangle\langle \ell|_1 \otimes |i\rangle\langle j|\ell\rangle\langle k|_2 \otimes |i\rangle\langle j|_3 \tag{1.A.10}$$

$$= \frac{1}{\sqrt{d}} \sum_{i,j,k} |k\rangle\langle j|_1 \otimes |i\rangle\langle k|_2 \otimes |i\rangle\langle j|_3. \tag{1.A.11}$$

Let us prove the first identity, i.e., that in (1.A.2), as follows:

$$\begin{aligned}
&\mathrm{Tr}_{23}\Big[M_{123}\left(\rho_1 \otimes \psi^L_{23}\right) M^\dagger_{123}\Big] \\
&= \mathrm{Tr}_{23}\Bigg[\left(\frac{1}{\sqrt{d}} \sum_{i,j,k} |k\rangle\langle j|_1 \otimes |i\rangle\langle k|_2 \otimes |i\rangle\langle j|_3\right)\left(\rho_1 \otimes \psi^L_{23}\right)\times \\
&\qquad\qquad \left(\frac{1}{\sqrt{d}} \sum_{i',j',k'} |j'\rangle\langle k'|_1 \otimes |k'\rangle\langle i'|_2 \otimes |j'\rangle\langle i'|_3\right)\Bigg] &(1.A.12)\\
&= \frac{1}{d} \sum_{i,j,k,i',j',k'} \mathrm{Tr}_{23}\left[\, (|k\rangle\langle j|_1 \otimes |i\rangle\langle k|_2 \otimes |i\rangle\langle j|_3)\left(\rho_1 \otimes \psi^L_{23}\right)\times\right. \\
&\qquad\qquad \left.(|j'\rangle\langle k'|_1 \otimes |k'\rangle\langle i'|_2 \otimes |j'\rangle\langle i'|_3)\,\right] &(1.A.13)\\
&= \frac{1}{d} \sum_{i,j,k,i',j',k'} |k\rangle\langle j|_1 \rho_1 |j'\rangle\langle k'|_1\, \mathrm{Tr}_{23}\Big[(|i\rangle\langle k|_2 \otimes |i\rangle\langle j|_3)\, \psi^L_{23}\, (|k'\rangle\langle i'|_2 \otimes |j'\rangle\langle i'|_3)\Big] &(1.A.14)\\
&= \frac{1}{d} \sum_{i,j,k,i',j',k'} |k\rangle\langle j|_1 \rho_1 |j'\rangle\langle k'|_1\ (\langle k|_2 \otimes \langle j|_3)\, \psi^L_{23}\, (|k'\rangle_2 \otimes |j'\rangle_3)\, \langle i'|i\rangle_2 \langle i'|i\rangle_3 &(1.A.15)\\
&= \sum_{j,k,j',k'} |k\rangle\langle j|_1 \rho_1 |j'\rangle\langle k'|_1\ (\langle k|_2 \otimes \langle j|_3)\, \psi^L_{23}\, (|k'\rangle_2 \otimes |j'\rangle_3) &(1.A.16)\\
&= \sum_{j,k,j',k'} |k\rangle\langle j|_1 \rho_1 |j'\rangle\langle k'|_1\ (\langle k|_2 \otimes \langle j|_3)\, (L_2 \otimes I_3)\, |\Gamma\rangle\langle\Gamma|_{23}\left(L_2^\dagger \otimes I_3\right)\times \\
&\qquad\qquad (|k'\rangle_2 \otimes |j'\rangle_3) &(1.A.17)\\
&= \sum_{j,k,j',k',i,i'} |k\rangle\langle j|_1 \rho_1 |j'\rangle\langle k'|_1\ (\langle k|_2 \otimes \langle j|_3)\left(L_2 |i\rangle\langle i'|_2 L_2^\dagger \otimes |i\rangle\langle i'|_3\right)\times \\
&\qquad\qquad (|k'\rangle_2 \otimes |j'\rangle_3) &(1.A.18)\\
&= \sum_{j,k,j',k',i,i'} |k\rangle\langle j|\rho|j'\rangle\langle k'|_1\, \langle k|L|i\rangle\langle i'|L^\dagger|k'\rangle\langle j|i\rangle\langle i'|j'\rangle &(1.A.19)\\
&= \sum_{j,k,j',k'} |k\rangle\langle j|\rho|j'\rangle\langle k'|_1\, \langle k|L|j\rangle\langle j'|L^\dagger|k'\rangle &(1.A.20)
\end{aligned}$$

$$= \sum_{j,k,j',k'} |k\rangle\langle k|L|j\rangle\langle j|\rho|j'\rangle\langle j'|L^\dagger|k'\rangle\langle k'|_1 \tag{1.A.21}$$

$$= \left[L\rho L^\dagger\right]_1 . \tag{1.A.22}$$

Now consider that

$$M_{123}^\dagger M_{123} = \left(\frac{1}{\sqrt{d}} \sum_{i',j',k'} |j'\rangle\langle k'|_1 \otimes |k'\rangle\langle i'|_2 \otimes |j'\rangle\langle i'|_3\right) \times \left(\frac{1}{\sqrt{d}} \sum_{i,j,k} |k\rangle\langle j|_1 \otimes |i\rangle\langle k|_2 \otimes |i\rangle\langle j|_3\right) \tag{1.A.23}$$

$$= \frac{1}{d} \sum_{i',j',k',i,j,k} |j'\rangle\langle k'|k\rangle\langle j|_1 \otimes |k'\rangle\langle i'|i\rangle\langle k|_2 \otimes |j'\rangle\langle i'|i\rangle\langle j|_3 \tag{1.A.24}$$

$$= \sum_{j',j,k} |j'\rangle\langle j|_1 \otimes |k\rangle\langle k|_2 \otimes |j'\rangle\langle j|_3 \tag{1.A.25}$$

$$= \sum_{j',j} |j'\rangle\langle j|_1 \otimes I_2 \otimes |j'\rangle\langle j|_3 \tag{1.A.26}$$

$$= \sum_{i,j} |i\rangle\langle j|_1 \otimes I_2 \otimes |i\rangle\langle j|_3 . \tag{1.A.27}$$

Then, for checking the second identity, i.e., that in (1.A.3), we find that

$$\mathrm{Tr}_{23}\left[M_{123}^\dagger M_{123} \left(\rho_1 \otimes \psi_{23}^L\right)\right]$$

$$= \mathrm{Tr}_{23}\left[\left(\sum_{i,j} |i\rangle\langle j|_1 \otimes I_2 \otimes |i\rangle\langle j|_3\right)\left(\rho_1 \otimes \psi_{23}^L\right)\right] \tag{1.A.28}$$

$$= \sum_{i,j} \mathrm{Tr}_{23}\left[\left(|i\rangle\langle j|_1 \otimes I_2 \otimes |i\rangle\langle j|_3\right)\left(\rho_1 \otimes \psi_{23}^L\right)\right] \tag{1.A.29}$$

$$= \sum_{i,j} |i\rangle\langle j|_1 \rho_1 \ \mathrm{Tr}_{23}\left[\left(I_2 \otimes |i\rangle\langle j|_3\right)\left(\psi_{23}^L\right)\right] \tag{1.A.30}$$

$$= \sum_{i,j} |i\rangle\langle j|_1 \rho_1 \ \mathrm{Tr}_{23}\left[\left(I_2 \otimes |i\rangle\langle j|_3\right)\left(L_2 \otimes I_3\right)|\Gamma\rangle\langle\Gamma|_{23}\left(L_2^\dagger \otimes I_3\right)\right] \tag{1.A.31}$$

$$= \sum_{i,j,k,\ell} |i\rangle\langle j|_1 \rho_1 \ \mathrm{Tr}_{23}\left[\left(I_2 \otimes |i\rangle\langle j|_3\right)\left(L_2 \otimes I_3\right)\left(|k\rangle\langle \ell|_2 \otimes |k\rangle\langle \ell|_3\right)\left(L_2^\dagger \otimes I_3\right)\right] \tag{1.A.32}$$

$$= \sum_{i,j,k,\ell} |i\rangle\langle j|_1 \rho_1 \ \mathrm{Tr}_{23}\left[L_2|k\rangle\langle \ell|_2 L_2^\dagger \otimes |i\rangle\langle j|k\rangle\langle \ell|_3\right] \tag{1.A.33}$$

$$= \sum_{i,j,k,\ell} |i\rangle\langle j|_1 \rho_1 \ \mathrm{Tr}[L_2 |k\rangle\langle \ell|_2 L_2^\dagger] \, \mathrm{Tr}[|i\rangle\langle j|k\rangle\langle \ell|_3] \tag{1.A.34}$$

$$= \sum_{i,j,k,\ell} |i\rangle\langle j|_1 \rho_1 \ \langle \ell|L^\dagger L|k\rangle\langle \ell|i\rangle\langle j|k\rangle \tag{1.A.35}$$

$$= \sum_{i,j} |i\rangle\langle j|_1 \rho_1 \ \langle i|L^\dagger L|j\rangle \tag{1.A.36}$$

$$= \sum_{i,j} |i\rangle\langle i|L^\dagger L|j\rangle\langle j|\rho \tag{1.A.37}$$

$$= \left[L^\dagger L \rho\right]_1 . \tag{1.A.38}$$

Then the third identity in (1.A.4) is the Hermitian conjugate of the above identity, and so we find that

$$\mathrm{Tr}_{23}\left[\left(\rho_1 \otimes \psi_{23}^L\right) M_{123}^\dagger M_{123}\right] = \left[\rho L^\dagger L\right]_1 . \tag{1.A.39}$$

This concludes the proof. ■

1.B Proof of Theorem 1

Consider the following appending quantum channel:

$$\mathcal{P}(\rho) := \rho \otimes \psi. \tag{1.B.1}$$

Using this definition, the WML algorithm (i.e., the algorithm presented in Section 1.3.1) can be expressed in the form of a quantum channel:

$$\left(\mathrm{Tr}_{PQ} \circ e^{\mathcal{M}\Delta} \circ \mathcal{P}\right)^{\circ n} \tag{1.B.2}$$

Now consider that

$$\frac{1}{2} \left\| e^{\mathcal{L}t} - \left(\mathrm{Tr}_{PQ} \circ e^{\mathcal{M}\Delta} \circ \mathcal{P}\right)^{\circ n} \right\|_\diamond$$

$$= \frac{1}{2}\left\|\left(e^{\mathcal{L}\Delta}\right)^{\circ n} - \left(\mathrm{Tr}_{PQ}\circ e^{\mathcal{M}\Delta}\circ\mathcal{P}\right)^{\circ n}\right\|_\diamond \tag{1.B.3}$$

$$\leq \frac{n}{2}\left\|e^{\mathcal{L}\Delta} - \mathrm{Tr}_{PQ}\circ e^{\mathcal{M}\Delta}\circ\mathcal{P}\right\|_\diamond \tag{1.B.4}$$

$$= \frac{n}{2}\left\|\mathcal{I} + \Delta\mathcal{L} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathcal{L}^r - \left(\mathrm{Tr}_{PQ}\circ\left(\mathcal{I} + \Delta\mathcal{M} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathcal{M}^r\right)\circ\mathcal{P}\right)^{\circ n}\right\|_\diamond \tag{1.B.5}$$

$$= \frac{n}{2}\left\|\mathcal{I} + \Delta\mathcal{L} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathcal{L}^r - \left(\mathcal{I} + \Delta\,\mathrm{Tr}_{PQ}\circ\mathcal{M}\circ\mathcal{P} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathrm{Tr}_{PQ}\circ\mathcal{M}^r\circ\mathcal{P}\right)\right\|_\diamond \tag{1.B.6}$$

$$= \frac{n}{2}\left\|\mathcal{I} + \Delta\mathcal{L} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathcal{L}^r - \left(\mathcal{I} + \Delta\mathcal{L} + \sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\mathrm{Tr}_{PQ}\circ\mathcal{M}^r\circ\mathcal{P}\right)\right\|_\diamond \tag{1.B.7}$$

$$\leq \frac{n}{2}\sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\left(\|\mathcal{L}^r\|_\diamond + \left\|\mathrm{Tr}_{PQ}\circ\mathcal{M}^r\circ\mathcal{P}\right\|_\diamond\right) \tag{1.B.8}$$

$$\leq \frac{n}{2}\sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\left(\|\mathcal{L}\|_\diamond^r + \left\|\mathrm{Tr}_{PQ}\right\|_\diamond\|\mathcal{M}\|_\diamond^r\|\mathcal{P}\|_\diamond\right) \tag{1.B.9}$$

$$\leq \frac{n}{2}\sum_{r=2}^{\infty}\frac{\Delta^r}{r!}\left(2^r + 1\cdot(2d)^r\cdot 1\right) \tag{1.B.10}$$

$$\leq n\sum_{r=2}^{\infty}\frac{\Delta^r(2d)^r}{r!} \tag{1.B.11}$$

$$\leq \frac{3n}{4}\Delta^2(2d)^2 \tag{1.B.12}$$

$$= \frac{3t^2(2d)^2}{4n}. \tag{1.B.13}$$

The first inequality follows from the submultiplicativity property of the diamond norm under composition. The fourth inequality follows directly from the following identities, which hold due to Lemma 7:

$$\mathrm{Tr}_{PQ}\Big[M(\rho\otimes\psi)M^\dagger\Big] = L\rho L^\dagger, \tag{1.B.14}$$

$$\mathrm{Tr}_{PQ}\Big[M^\dagger M\,(\rho\otimes\psi)\Big] = L^\dagger L\rho, \tag{1.B.15}$$

$$\mathrm{Tr}_{PQ}\Big[(\rho\otimes\psi)\,M^\dagger M\Big] = \rho L^\dagger L. \tag{1.B.16}$$

Then the second inequality follows directly from the triangle inequality, and the third inequality follows from the submultiplicativity property of the diamond norm under composition. We will later justify the fourth inequality. The sixth inequality holds due to the

following exponential tail bound: for all $x \in [0, 1]$, we have $\sum_{r=2}^{\infty} \frac{x^r}{r!} \leq \frac{3}{4}x^2$, and for our case, this holds when $2d\Delta \leq 1$ or $n \geq 2dt$. We will see that this condition is true for our case.

Now, the justification for the fourth inequality is provided in Appendix 4 of [GKP+25]; we reproduce it here for completeness. Let τ be an arbitrary bipartite quantum state of a reference system R and an input system S. Then, let τ be an arbitrary bipartite quantum state of a reference system R and an input system S. Then,

$$\begin{aligned}
&\|(\mathrm{id} \otimes \mathcal{L})(\tau)\|_1 \\
&= \left\| (I \otimes L)\tau(I \otimes L^\dagger) - \frac{1}{2}\left\{ I \otimes L^\dagger L, \tau \right\} \right\|_1 && \text{(1.B.17)} \\
&\leq \left\| (I \otimes L)\tau(I \otimes L^\dagger) \right\|_1 + \frac{1}{2}\left\| (I \otimes L^\dagger L)\tau \right\|_1 + \frac{1}{2}\left\| \tau(I \otimes L^\dagger L) \right\|_1 && \text{(1.B.18)} \\
&= \left\| \tau^{1/2}(I \otimes L^\dagger L)\tau^{1/2} \right\|_1 + \left\| (I \otimes L^\dagger L)\tau \right\|_1 && \text{(1.B.19)} \\
&\leq \left\| (I \otimes L^\dagger L) \right\|_\infty \left\| \tau^{1/2}\tau^{1/2} \right\|_1 + \left\| (I \otimes L^\dagger L) \right\|_\infty \|\tau\|_1 && \text{(1.B.20)} \\
&= 2\left\| L^\dagger L \right\|_\infty && \text{(1.B.21)} \\
&\leq 2\left\| L^\dagger L \right\|_1 && \text{(1.B.22)} \\
&= 2, && \text{(1.B.23)}
\end{aligned}$$

where the last equality follows from the assumption that the Lindblad operator has a unit Hilbert–Schmidt norm. Therefore we conclude that

$$\|\mathcal{L}\|_\diamond = \sup_\tau \|(\mathrm{id} \otimes \mathcal{L})(\tau)\|_1 \leq 2. \tag{1.B.24}$$

Similarly,

$$\begin{aligned}
&\|(\mathrm{id} \otimes \mathcal{M})(\tau)\|_1 \\
&= \left\| (I \otimes M)\tau(I \otimes M^\dagger) - \frac{1}{2}\left\{ I \otimes M^\dagger M, \tau \right\} \right\|_1 && \text{(1.B.25)} \\
&\leq \left\| (I \otimes M)\tau(I \otimes M^\dagger) \right\|_1 + \frac{1}{2}\left\| (I \otimes M^\dagger M)\tau \right\|_1 + \frac{1}{2}\left\| \tau(I \otimes M^\dagger M) \right\|_1 && \text{(1.B.26)}
\end{aligned}$$

$$= \left\|\tau^{1/2}(I \otimes M^\dagger M)\tau^{1/2}\right\|_1 + \left\|(I \otimes M^\dagger M)\tau\right\|_1 \tag{1.B.27}$$

$$\leq \left\|I \otimes M^\dagger M\right\|_\infty \left\|\tau^{1/2}\tau^{1/2}\right\|_1 + \left\|I \otimes M^\dagger M\right\|_\infty \|\tau\|_1 \tag{1.B.28}$$

$$= 2\left\|M^\dagger M\right\|_\infty \tag{1.B.29}$$

$$= 2d. \tag{1.B.30}$$

The last equality follows from the fact that

$$M^\dagger M = |\Gamma\rangle\langle\Gamma|_{13} \otimes I_2 = d|\Phi\rangle\langle\Phi|_{13} \otimes I_2, \tag{1.B.31}$$

so that

$$\left\|M^\dagger M\right\|_\infty = d. \tag{1.B.32}$$

Therefore we conclude that

$$\|\mathcal{M}\|_\diamond = \sup_\tau \|\operatorname{id} \otimes \mathcal{M}\|_1 \leq 2d. \tag{1.B.33}$$

This concludes the justification of the fourth inequality.

That being said, if we want the final error to be less than or equal to ε, then from (1.B.13), it suffices to take

$$n \geq \frac{3t^2(2d)^2}{4\varepsilon} = O\left(\frac{d^2t^2}{\varepsilon}\right). \tag{1.B.34}$$

samples of the program state. This concludes the proof.

CHAPTER 2

WAVE MATRIX LINDBLADIZATION II: GENERAL LINDBLADIANS, LINEAR COMBINATIONS, AND POLYNOMIALS[1]

2.1 Abstract

In this chapter, we investigate the problem of simulating open system dynamics governed by the well-known Lindblad master equation. In the previous chapter, we introduced an input model in which Lindblad operators are encoded into pure quantum states, called program states, and we also introduced a method, called wave matrix Lindbladization, for simulating Lindbladian evolution by means of interacting the system of interest with these program states. Therein, we focused on a simple case in which the Lindbladian consists of only one Lindblad operator and a Hamiltonian. Here, we extend the method to simulating general Lindbladians with local Lindblad operators and other cases in which a Lindblad operator is expressed as a linear combination or a polynomial of the operators encoded into the program states. We propose quantum algorithms for all these cases and also investigate their sample complexity, i.e., the number of program states needed to simulate a given Lindbladian evolution approximately. We also investigate the gate complexity of one of our algorithms for simulating general Lindbladians, quantifying the number of elementary gates required. Finally, we show that our algorithms achieve a quadratic improvement in sample complexity with respect to the approximation error, offering a more efficient route to simulating Lindbladian evolution than approaches based on tomography. This suggests a form of quantum copy-protection: a party can send pro-

[1]This chapter combines material from two papers. The majority is taken verbatim from [PW23b], with modifications to incorporate local Lindblad operators and an updated analysis of the Trotter-based algorithm. The gate complexity analysis of the sampling-based algorithm is adapted from [SPP+25], which is currently under review at *Physical Review Research*.

gram states that enable Lindbladian simulation without disclosing details of the encoded operator, thereby addressing an open question of [KLL+17] on privately executable operations beyond Hamiltonian simulation.

2.2 Introduction

Quantum simulation involves simulating or modeling the behavior of a complex quantum system using a quantum computer, allowing researchers to further investigate its properties in detail. The problem of simulating a closed quantum system, also known as Hamiltonian simulation, is well-studied, and hitherto, many quantum algorithms have been proposed to solve it [Llo96, BCC+14a, BCG14, LMR14, BCC+14b, BCK15, LC17, KLL+17]. However, many practical systems are not closed; rather, they interact with their environment, resulting in more complex dynamics that are well captured by the Lindblad master equation if the system under consideration is Markovian in nature [Lin76, GKS76]. This equation is critical to understanding the behavior of many open quantum systems [BP02, Wei21], in condensed matter [Pro11, MTAB12, OLG12], quantum chemistry [Nit06, MK08], quantum optics [PK98, GZ04], entanglement preparation [KBD+08, KRS11, RRS16], thermal state preparation [KB14], quantum state engineering [VWC09], and the effects of noise on quantum computers [MPGC13].

In this chapter, we address the problem of simulating Lindbladian evolution of a finite-dimensional quantum system. Starting from an initial state ρ, we aim to simulate its dynamics over a period of time $t \geq 0$, as governed by the Lindblad master equation:

$$\frac{\partial \rho}{\partial t} = \mathcal{L}(\rho) := -i[H, \rho] + \sum_{k=1}^{K} L_k \rho L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho\right\}. \tag{2.2.1}$$

Here, the first term $-i[H, \rho]$ accounts for the unitary evolution of the system under the

system Hamiltonian H. The second term $\sum_{k=1}^{K} L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\}$ captures non-unitary dynamics from interactions with the environment, described by local Lindblad operators $\{L_k\}_{k=1}^K$. By local, we mean that these operators act nontrivially only on a constant number of qubits. Furthermore, these operators are not necessarily Hermitian and in fact have no constraints on them.

By simulating the evolution mentioned above for time t, we are referring to implementing its corresponding quantum channel $e^{\mathcal{L}t}$, which is the solution of (2.2.1), where

$$e^{\mathcal{L}t}(\rho) = \sum_{s=0}^{\infty} \frac{\mathcal{L}^s(\rho)t^s}{s!}, \tag{2.2.2}$$

and $\mathcal{L}^s$ denotes s sequential applications of the Lindbladian $\mathcal{L}$. The equality above simply comes from the Taylor series expansion of the exponential. For small t, we have the expansion $e^{\mathcal{L}t}(\rho) = \rho + \mathcal{L}(\rho)t + O(t^2)$, and we make use of it in what follows.

There has been an increasing interest in developing efficient quantum algorithms for simulating the dynamics of open quantum systems in recent years [CL17, CW17, KSMM22, SHMS+22, SBH+23] (see [MOTT22] for a review). These works are based on an assumption that a succinct representation of or black-box access to a set of Lindblad operators is provided beforehand. One such succinct representation is a list of non-zero coefficients when writing these operators as a linear combination of Pauli strings [CW17].

As introduced in the previous chapter, we take a different approach to the above problem. Specifically, our approach differs from the methods mentioned above in how the Lindblad operators are provided as input. Since each operator L_k is local, for each $k \in \{1, \ldots, K\}$ there exists a constant-size subsystem A_k such that

$$L_k = L_{A_k} \otimes I_{\overline{A_k}}, \qquad d_{A_k} \coloneqq \dim(\mathcal{H}_{A_k}) = O(1), \tag{2.2.3}$$

where $\overline{A_k}$ denotes the complement of A_k, so that the total system $S = A_k\overline{A_k}$. Now suppose

that each Lindblad operator L_k is encoded into a pure state $|\psi_k\rangle$ in the following way:

$$|\psi_k\rangle \coloneqq (L_{A_k} \otimes I)|\Gamma\rangle, \tag{2.2.4}$$

where

$$|\Gamma\rangle \coloneqq \sum_{j=1}^{d_{A_k}} |j\rangle|j\rangle \tag{2.2.5}$$

is a maximally entangled vector. Suppose further that we have access to multiple copies of this state. We refer to such a state as a program state because it can encode any unit-norm linear operator. By unit-norm, we mean $\left\|L_{A_k}\right\|_2 = 1$, where $\|A\|_2 \coloneqq \sqrt{\mathrm{Tr}[A^\dagger A]}$ is the Schatten-2 norm of a matrix A (also known as the Hilbert–Schmidt norm). This constraint on L_{A_k} is due to the requirement that $|\psi_k\rangle$ should be a quantum state (i.e., normalized). Furthermore, for the Hamiltonian term in (2.2.1), we follow the input model of [KLL+17] and suppose that the Hamiltonian H is given as a linear combination of program states $\{\sigma_j\}_{j=1}^J$, i.e.,

$$H \coloneqq \sum_{j=1}^{J} c_j \sigma_j, \tag{2.2.6}$$

where $c_j \in \mathbb{R}$, and that we have access to multiple copies of σ_j for all $j \in \{1, \ldots, J\}$.

In essence, we are interested in answering the following question: given one copy of an unknown quantum state ρ, some number n_j of copies of the program state σ_j for all $j \in \{1, \ldots, J\}$, and m_k copies of the program state $\psi_k \coloneqq |\psi_k\rangle\langle\psi_k|$ for all $k \in \{1, \ldots, K\}$, can we approximately simulate the quantum channel $e^{\mathcal{L}t}$ up to an error ε? In other words, can we realize the following transformation?

$$\rho \otimes \sigma_1^{\otimes n_1} \otimes \cdots \otimes \sigma_J^{\otimes n_J} \otimes \psi_1^{\otimes m_1} \otimes \cdots \otimes \psi_K^{\otimes m_K} \overset{\varepsilon}{\approx\!\longrightarrow} e^{\mathcal{L}t}(\rho). \tag{2.2.7}$$

As done in the previous chapter, we call this modified problem sample-based Lindbladian simulation, and it can be seen as a natural analogue to sample-based Hamiltonian simulation [LMR14, KLL+17]. In the previous chapter, we introduced an approach for solving

a simpler version of this problem, called *Wave Matrix Lindbladization* (WML), which is an analogue to *Density Matrix Exponentiation* [LMR14]. That is, in [PW23a], we focused on a relatively simple case in which the Lindbladian consists of only one Lindblad operator and a Hamiltonian term. We did this so that the reader could grasp the intuition behind the techniques introduced. That being said, in this chapter, we extend wave matrix Lindbladization to general Lindbladians and beyond.

2.2.1 Summary of Main Results

In this chapter, we present two quantum algorithms that implement the quantum channel $e^{\mathcal{L}t}$ with some desired accuracy $\varepsilon \in (0, 1)$. We call these the sampling-based approach and the Trotter-like approach. We then investigate the sample complexity of these algorithms (see Section 2.3.1). By sample complexity, we mean the number of copies of the program states $\{\sigma_j\}_{j=1}^J$ and $\{\psi_k\}_{k=1}^K$ needed to achieve the above task; i.e., this number is equal to $\sum_{j=1}^J n_j + \sum_{k=1}^K m_k$. Additionally, we investigate the gate complexity of the sampling-based approach, quantifying the number of elementary gates required (see Section 2.3.1).

Furthermore, we provide another, different extension of the method introduced in [PW23a]. Previously, we considered a Lindbladian with only one Lindblad operator and a Hamiltonian, and this entire Lindblad operator was given encoded in a single program state. We extend this simple case to that in which this Lindblad operator is provided as a linear combination or a polynomial of the operators encoded in the program states. We propose quantum algorithms for these cases as well under the assumption that we have access to unitaries that prepare these program states (see Sections 2.3.2 and 2.3.3). We then investigate the query complexity of these algorithms, i.e., the number of times these state-preparation unitaries are queried to approximate a given evolution.

Finally, we show in Section 2.3.4 that our algorithms achieve a quadratic improvement in sample complexity with respect to the approximation error, offering a more efficient route to simulating Lindbladian evolution than approaches based on tomography. This suggests a form of quantum copy-protection: a party can send program states that enable Lindbladian simulation without disclosing details of the encoded operator, thereby addressing an open question of [KLL+17] on privately executable operations beyond Hamiltonian simulation.

2.2.2 Notation

We employ the same notation used in the previous chapter, but recall it here for convenience. Let $\mathcal{H}_S$ denote a d-dimensional Hilbert space associated with a quantum system S. We denote the set of quantum states acting on $\mathcal{H}_S$ by $\mathcal{D}(\mathcal{H}_S)$. Let $\mathrm{Tr}[X]$ denote the trace of a matrix X, i.e., the sum of its diagonal elements. Also, let $X^\dagger$ denote the Hermitian conjugate (or adjoint) of the matrix X. The Schatten p-norm of a matrix X is defined for $p \in [1, \infty)$ as follows:

$$\|X\|_p := \left(\mathrm{Tr}\left[\left(X^\dagger X\right)^{\frac{p}{2}}\right]\right)^{\frac{1}{p}}. \tag{2.2.8}$$

For the purposes of this chapter, we use Schatten norms with $p = 1$ (also called trace norm) and $p = 2$ (Hilbert–Schmidt norm). Furthermore, let $[X, Y] := XY - YX$ and $\{X, Y\} := XY + YX$ denote the commutator and anti-commutator of the operators X and Y, respectively.

The diamond distance between two quantum channels $\mathcal{N}$ and $\mathcal{M}$ is defined as follows [Kit97]:

$$\|\mathcal{N} - \mathcal{M}\|_\diamond := \sup_{\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)} \|(\mathcal{I}_R \otimes \mathcal{N})(\rho) - (\mathcal{I}_R \otimes \mathcal{M})(\rho)\|_1, \tag{2.2.9}$$

where R is a reference system and $\mathcal{I}_R$ is the identity channel acting on the system R. An

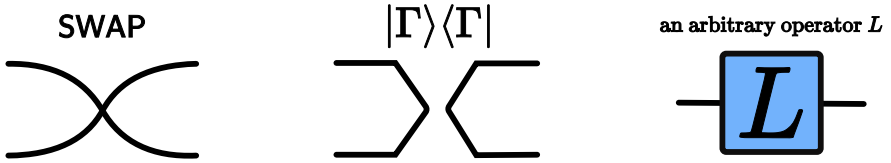


Figure 2.1: Tensor-network diagrams of operators SWAP, $|\Gamma\rangle\langle\Gamma|$, and L.

important point to note here is that, in the above definition, the dimension of R is arbitrarily large. However, it is known that it suffices to perform the optimization over pure bipartite states with the dimension of R equal to the dimension of S. Furthermore, the quantity in the objective function of the above optimization is the trace distance, defined as $\|\rho - \sigma\|_1$ for two quantum states $\rho, \sigma \in \mathcal{D}(\mathcal{H}_S)$. In what follows, we employ the normalized diamond distance $\frac{1}{2}\|\mathcal{N} - \mathcal{M}\|_\diamond$ to measure approximation error—the normalization factor of $\frac{1}{2}$ guarantees that $\frac{1}{2}\|\mathcal{N} - \mathcal{M}\|_\diamond \in [0, 1]$ for quantum channels $\mathcal{N}$ and $\mathcal{M}$.

For ρ a state in $\mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$, we denote the partial trace over the Hilbert space $\mathcal{H}_R$ by $\mathrm{Tr}_R[\rho]$. We also sometimes use a different notation for partial trace; i.e., given a multipartite state ρ, we use the notation $\mathrm{Tr}_k[\rho]$ to denote the action of tracing out the k^{th} party. Furthermore, we define the maximally entangled vector in $\mathcal{H}_R \otimes \mathcal{H}_S$ as

$$|\Gamma\rangle_{RS} := \sum_i |i\rangle_R |i\rangle_S, \tag{2.2.10}$$

and we denote its normalized version, i.e., the maximally entangled state, by $|\Phi\rangle$. We also define the unitary swap operation and unitary cyclic permutation operation (cyclic swap) in the following way:

$$\mathrm{SWAP} := \sum_{i,j} |i\rangle\langle j| \otimes |j\rangle\langle i|, \tag{2.2.11}$$

$$\mathrm{CYCSWAP} := \sum_{i_1, i_2, \ldots, i_n} |i_n\rangle\langle i_1| \otimes |i_1\rangle\langle i_2| \otimes \cdots \otimes |i_{n-1}\rangle\langle i_n|, \tag{2.2.12}$$

where n is the number of systems on which the cyclic swap is performed.

In this chapter, we make extensive use of tensor-network diagrams. Figure 2.1 depicts tensor-network diagrams for some basic operators defined above, such as SWAP and $|\Gamma\rangle\langle\Gamma|$. For more background on tensor-network diagrams, please refer to [BB17]. Throughout this chapter, we sometimes suppress system labels for ease of notation; however, they should be clear from the context.

2.3 Quantum Algorithms for Simulating Markovian Dynamics

2.3.1 Simulating General Lindbladians

In what follows, we consider the case in which the Lindbladian consists of multiple Lindblad operators, as well as a Hamiltonian. To be more precise, we are interested in simulating the action of the dynamics specified by (2.2.1) and the following Lindbladian:

$$\mathcal{L}(\rho) := -i[H,\rho] + \sum_{k=1}^{K} L_k \rho L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho\right\}, \tag{2.3.1}$$

on a quantum state $\rho \in \mathcal{D}(S)$ for time t.

Here, for the Hamiltonian term, we follow the input model of [KLL+17], assuming that H is provided as a linear combination of program states $\{\sigma_j\}_{j=1}^{J}$, i.e.,

$$H := \sum_{j=1}^{J} c_j \sigma_j, \tag{2.3.2}$$

where $c_j \in \mathbb{R}$.

Throughout this section, we assume that every Lindblad operator L_k is local, that is, it acts non-trivially on a constant number of qubits. In other words, for each $k \in \{1,\ldots,K\}$,

there is a constant-size subsystem A_k such that

$$L_k = L_{A_k} \otimes I_{\overline{A_k}}, \qquad d_{A_k} \coloneqq \dim(\mathcal{H}_{A_k}) = O(1), \tag{2.3.3}$$

where $\overline{A_k}$ denotes the complement of A_k, so that $S = A_k\overline{A_k}$. We impose no locality restriction on H. In addition, let us suppose that each L_{A_k} is encoded in a pure state ψ_k as described in (2.2.4). While L_{A_k} can have an arbitrary Hilbert–Schmidt norm in general, we encode its normalized version $L_{A_k}/\left\|L_{A_k}\right\|_2$ using the maximally entangled vector:

$$|\psi_k\rangle \coloneqq \frac{(L_{A_k} \otimes I)\,|\Gamma_k\rangle}{\left\|L_{A_k}\right\|_2}, \qquad |\Gamma_k\rangle \coloneqq \sum_{j=1}^{d_{A_k}} |j\rangle \otimes |j\rangle. \tag{2.3.4}$$

Consequently,

$$\left\|L_{A_k}\right\|_2^2\, \psi_k = (L_{A_k} \otimes I)\,|\Gamma_k\rangle\langle\Gamma_k|\,(L_{A_k} \otimes I)^\dagger. \tag{2.3.5}$$

Sampling-based Approach

We assume that we are given access to $\{(c_j, \sigma_j)\}_{j=1}^J$ and $\{(\|L_{A_k}\|_2^2, \psi_k)\}_{k=1}^K$, and we define

$$c \coloneqq \sum_{j=1}^{J} \left|c_j\right| + \sum_{k=1}^{K} \left\|L_{A_k}\right\|_2^2. \tag{2.3.6}$$

The pseudocode for the sampling-based WML algorithm for simulating the channel $e^{\mathcal{L}t}$ corresponding to the Lindbladian in (2.3.1), up to error ε in diamond distance is as follows:

Algorithm 1 – Set $n = O(\frac{c^2t^2}{\varepsilon})$ and $\Delta = \frac{ct}{n}$. Repeat the following steps n times:

1. Randomly sample a Hamiltonian program state σ_j or a Lindbladian program state ψ_k, where σ_j has probability $\frac{|c_j|}{c}$ of being sampled and ψ_k has probability $\frac{\|L_k\|_2^2}{c}$ of being sampled.

2. Initialize the program register(s) according to the state sampled in Step 1:

   - If σ_j is sampled, allocate a single program register P isomorphic to the system register S and load σ_j on P.
   - If ψ_k is sampled, allocate two program registers P_k and Q_k with $\dim P_k = \dim Q_k = d_{A_k}$ and load ψ_k on $P_k Q_k$.

3. If a Hamiltonian program state σ_j is sampled in Step 1, apply the quantum channel $e^{\mathcal{N}_j \Delta}$ on the joint system SP. Here, $\mathcal{N}_j$ is a Lindbladian:

$$\mathcal{N}_j(\cdot) \coloneqq -i\Big[\operatorname{sgn}(c_j)\ \mathrm{SWAP}_{SP},\ (\cdot)\Big], \tag{2.3.7}$$

   where $\operatorname{sgn}(x)$ evaluates to 1 if x is non-negative and -1 otherwise.

4. If a Lindbladian program state ψ_k is sampled in Step 1 instead, apply the quantum channel $e^{\mathcal{M}_k \Delta}$ on the joint system SP_kQ_k. Here, $\mathcal{M}_k$ is a single-operator Lindbladian:

$$\mathcal{M}_k(\cdot) \coloneqq M_k(\cdot)M_k^\dagger - \frac{1}{2}\left\{M_k^\dagger M_k, \cdot\right\}, \tag{2.3.8}$$

   with Lindblad operator

$$M_k \coloneqq \frac{1}{\sqrt{d_{A_k}}} \left(I_S \otimes |\Gamma_k\rangle\langle\Gamma_k|_{P_kQ_k}\right)\left(I_{\overline{A_k}} \otimes \mathrm{SWAP}_{A_kP_k} \otimes I_{Q_k}\right). \tag{2.3.9}$$

   It is important to note that since L_k is local and acts non-trivially only on A_k, the operator M_k is also local: it acts non-trivially only on $A_kP_kQ_k$ and trivially on $\overline{A_k}$.

5. Trace out the program register(s) that are used in this round.

The step-by-step flow of this algorithm is illustrated in Figure 2.2. It is clear that since the sampling probabilities are $|c_j|/c$ for σ_j and $\|L_{A_k}\|_2^2/c$ for ψ_k, the algorithm above on average over n iterations uses

$$n_j = |c_j|/c \cdot O(c^2t^2/\varepsilon) = O(|c_j|ct^2/\varepsilon) \tag{2.3.10}$$

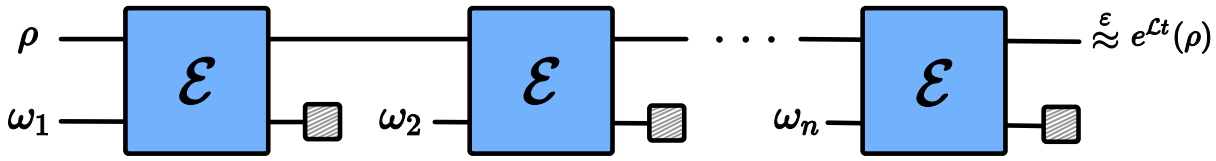


Figure 2.2: Our quantum algorithm, repeated $n = O(c^2t^2/\varepsilon)$ times, for approximating the target quantum channel, i.e., $e^{\mathcal{L}t}$, with an approximation error of ε. At the k^{th} iteration, the state ω_k denotes the sampled program state. The channel $\mathcal{E}$ applied in that step is either the unitary channel $e^{\mathcal{N}_j\Delta}$ or the Lindbladian channel $e^{\mathcal{M}_k\Delta}$, depending on whether the sampled program state encodes a Hamiltonian or a Lindbladian operator. Here, the Lindbladians $\mathcal{M}_k$ and $\mathcal{N}_j$ are defined in (2.3.8) and (2.3.7), respectively. Each hatched rectangle represents the trace-out operation.

copies of the program state σ_j and

$$m_k = \left\|L_{A_k}\right\|_2^2 / c \cdot O(c^2t^2/\varepsilon) = O\left(\left\|L_{A_k}\right\|_2^2 ct^2/\varepsilon\right) \tag{2.3.11}$$

copies of the program state ψ_k.

Note that in Step 3 of the above algorithm, we apply the quantum channel $e^{\mathcal{N}_j\Delta}$ on the joint system SP. Since this is a unitary channel, it can be implemented in principle on a quantum computer. To be more precise, when the Hamiltonian program register P is isomorphic to the full system register S (of dimension d), implementing this unitary channel uses $O(\log d)$ elementary one- and two-qubit gates [KLL+17]. If, instead, one makes the realistic assumption that H is local so that each of the Hamiltonian program states σ_j consists of only a constant number of qubits, then this implies that the required program register P has constant size and the same unitary can be implemented with $O(1)$ elementary gates. On the other hand, we employ the algorithm of [CW17] to implement the quantum channel $e^{\mathcal{M}_k\Delta}$ used in Step 4. This algorithm is based on the linear combination of unitaries (LCU) framework.

We are now ready to state the main result on the gate complexity of the sampling-based WML algorithm, where by gate complexity we mean the total number of one- and two-qubit gates required to execute the algorithm.

Theorem 8 (Gate Complexity) *Let $\mathcal{L}$ be a Lindbladian as defined in (2.3.1). The sampling-based WML algorithm can simulate the channel $e^{\mathcal{L}t}$ up to error ε in the normalized diamond distance using*

$$O\left(\frac{c^2t^2}{\varepsilon}\cdot\max\left\{\log d,\ \frac{\log^2(ct/\varepsilon)}{\log\log(ct/\varepsilon)}\right\}\right) \tag{2.3.12}$$

one- and two-qubit gates, where c is defined in (2.3.6) and d is the dimension of the system S. In the special case, where the Hamiltonian part of $\mathcal{L}$ is local, the gate complexity reduces to

$$O\left(\frac{c^2t^2\ \log^2(ct/\varepsilon)}{\varepsilon\ \log\log(ct/\varepsilon)}\right). \tag{2.3.13}$$

Proof. The proof is given in Appendix 2.A. ■

Trotter-like Approach

Here, we present another quantum algorithm for simulating the quantum channel $e^{\mathcal{L}t}$ with the Lindbladian $\mathcal{L}$ given by (2.3.1). Before delving into the specifics of this algorithm, we first provide a brief overview of it in order to gain an intuition behind its inner workings. Conceptually, one can think of the Lindbladian $\mathcal{L}$ as a linear combination of single-operator Lindbladians $\mathcal{L}_1,\ldots,\mathcal{L}_r$, i.e., $\sum_{i=1}^r \mathcal{L}_i$. By "single-operator," we mean that each $\mathcal{L}_i$ consists of only one term, and this term can be either a Hamiltonian term or a non-Hamiltonian term with a single Lindblad operator. Now, if the quantum channels $e^{\mathcal{L}_1 t},\ldots,e^{\mathcal{L}_r t}$ associated with the single-operator Lindbladians can be simulated efficiently, then the intuition behind this approach is to approximate $e^{\mathcal{L}t}$ by sequentially applying

these simple channels for short time steps and repeating this composite sequence multiple times. Due to its similarity with Trotter-based approaches in Hamiltonian [Llo96] and Lindbladian [CL17] simulation, we call it the Trotter-like approach.

With this high-level intuition established, we now present the full algorithm.

Algorithm 2 – Set

$$n = O\left(\frac{(J+K)^{3/2} \|\mathcal{L}\|_{\max}^2 t^2}{\varepsilon}\right), \tag{2.3.14}$$

where

$$\|\mathcal{L}\|_{\max} := \max\left\{|c_1|, \ldots, |c_J|, \left\|L_{A_1}\right\|_2^2, \ldots, \left\|L_{A_K}\right\|_2^2\right\}. \tag{2.3.15}$$

Repeat the following steps n times:

1. To begin with, repeat the following three steps for all k ranging from K to 1:
   (a) Allocate two program registers P_k and Q_k with $\dim P_k = \dim Q_k = d_{A_k}$ and load ψ_k (defined in (2.3.4)) on $P_k Q_k$.
   (b) Apply the quantum channel $e^{\mathcal{M}_k \Delta_k}$ on the joint system $S P_k Q_k$, where $\Delta_k := \frac{\|L_{A_k}\|_2^2 t}{2n}$ and the Lindbladian $\mathcal{M}_k$ is defined in (2.3.8).
   (c) Trace out the program registers P_k and Q_k.
   (d) Set ρ to be the state obtained after performing the above two steps.
2. Then, repeat the following three steps for all j ranging from J to 1:
   (a) Allocate a single program register P isomorphic to the system register S and load σ_j on P.
   (b) Apply the quantum channel $e^{\mathcal{N}_j \Delta'_j}$ on the joint system $S P$, where $\Delta'_j := \frac{|c_j| t}{2n}$ and the Lindbladian $\mathcal{N}_j$ is defined in (2.3.7).

(c) Trace out the program register P.

(d) Set ρ to be the state obtained after performing the above two steps.

3. Repeat Steps 2(a)–2(c) for all j ranging from 1 to J.

4. Repeat Steps 1(a)–1(c) for all k ranging from 1 to K.

The step-by-step flow of this algorithm is illustrated in Figure 2.3. Let us note that the particular ordering of steps used in Algorithm 2 is inspired by that used in [CL17, Proposition 2], and it has the effect of reducing the dependence of the algorithm's sample complexity on $J + K$.

Theorem 9 below states that the algorithm presented above approximates the channel $e^{\mathcal{L}t}$ up to ε error in diamond distance using $n = O((J + K)^{3/2} \|\mathcal{L}\|_{\max}^2 t^2/\varepsilon)$ copies of each of the program states $\sigma_1, \ldots, \sigma_J, \psi_1, \ldots, \psi_K$, so that the total number of program states used is $O((J + K)^{5/2} \|\mathcal{L}\|_{\max}^2 t^2/\varepsilon)$.

Theorem 9 *Given access to n copies of the program states $\sigma_1, \ldots, \sigma_J \in \mathcal{D}(\mathcal{H}_S)$ and $\psi_1, \ldots, \psi_K \in \mathcal{D}(\mathcal{H}_P \otimes \mathcal{H}_Q)$, there exists an algorithm $\mathcal{A}$ satisfying:*

$$\frac{1}{2} \left\| e^{\mathcal{L}t} - \mathcal{A} \right\|_\diamond \leq \varepsilon. \tag{2.3.16}$$

This approximation to within ε in diamond distance is achieved using only $n = O((J + K)^{3/2} \|\mathcal{L}\|_{\max}^2 t^2/\varepsilon)$ copies of each provided program state, where $\|\mathcal{L}\|_{\max}$ is defined in (2.3.15). The total number of program states used is then

$$O\Big((J + K)^{5/2} \|\mathcal{L}\|_{\max}^2 t^2/\varepsilon\Big). \tag{2.3.17}$$

Proof. The proof is given in Appendix 2.C. ■

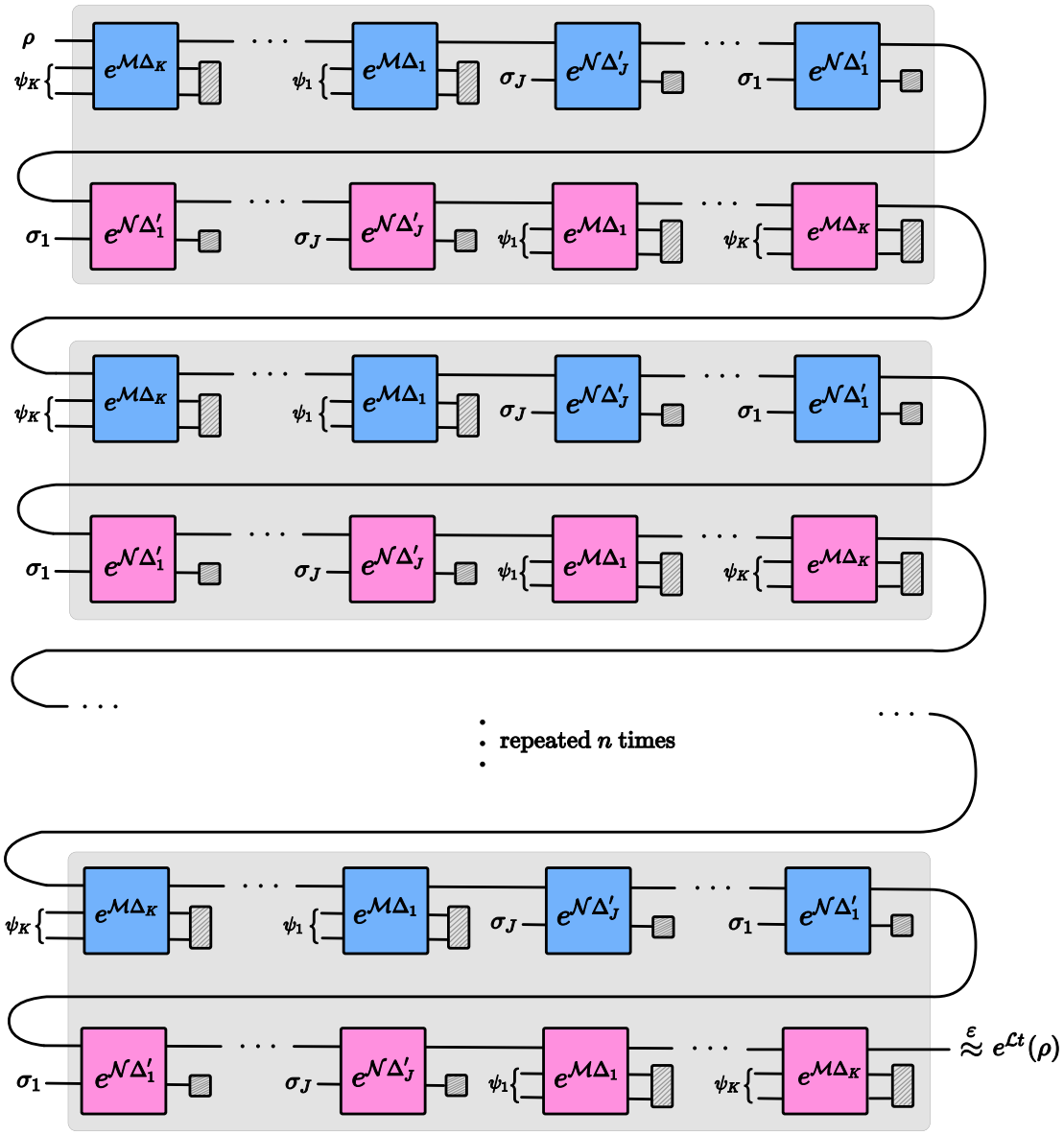


Figure 2.3: The gray strip at the very top represents the flow of Algorithm 2 and is repeated $n = O((J+K)^{3/2} \|\mathcal{L}\|_{\max}^2 t^2/\varepsilon)$ times. The blue boxes represent the flow of Steps 1 and 2 of Algorithm 2, while the pink boxes represent that of Steps 3 and 4 (reverse of Steps 1 and 2). The algorithm approximates the target quantum channel $e^{\mathcal{L}t}$ with an approximation error of ε. Each hatched rectangle represents the trace-out operation. Initially, the state ρ is in register 1. The program states $\psi_1, \ldots, \psi_K$ are in registers 2 and 3, while the program states $\sigma_1, \ldots, \sigma_J$ are in register 2. The channels $e^{\mathcal{M}_k \Delta_k}$ and $e^{\mathcal{N}_j \Delta'_j}$ correspond to Lindbladian evolutions with Lindbladians $\mathcal{M}_k$ and $\mathcal{N}_j$ defined in (2.3.8) and (2.3.7), respectively.

Remark 10 *To compare the sample complexity of Algorithms 1 and 2, consider that the following inequality holds:*

$$c \leq (J + K) \|\mathcal{L}\|_{\max} , \tag{2.3.18}$$

where c is defined in (2.3.6)*. As a consequence, it follows that the sample complexity of Algorithm 1 never exceeds that of Algorithm 2. Regardless, for completeness and comparison purposes, we have provided a full analysis of the Trotter-like approach in Appendix 2.C.*

2.3.2 Simulating Linear Combinations

In this section, we consider the case of simulating a Lindbladian evolution with a single Lindblad operator L and no Hamiltonian. To be more specific, we want to simulate the action of the dynamics on a quantum state ρ for time t according to the following Lindbladian:

$$\mathcal{L}(\rho) = L\rho L^\dagger - \frac{1}{2}\left\{L^\dagger L, \rho\right\}. \tag{2.3.19}$$

Furthermore, in this case, let us suppose that the Lindblad operator L is provided beforehand as a linear combination of operators $L_1, \ldots, L_K$:

$$L \coloneqq \sum_{k=1}^{K} c_k L_k, \tag{2.3.20}$$

where $c_k > 0$ and $\|L_k\|_2 = 1$, for all k ranging from 1 to K. For now, we do not assume that the operators $L_1, \ldots, L_K$ are local for maintaining generality. We discuss this special case in the next section, that is, in Section 2.3.2.

That being said, we assume that the operator L_k is encoded in the program state ψ_k in the same way that was considered previously in (2.2.4) (i.e., $|\psi_k\rangle \coloneqq (L_k \otimes I)|\Gamma\rangle$). Let us

further suppose that we have access to unitaries $U_1, \ldots, U_K$ that prepare these program states:

$$U_k|0\rangle = |\psi_k\rangle \qquad \forall k \in \{1, \ldots, K\}. \tag{2.3.21}$$

Given efficient implementations of the above unitaries, we can efficiently implement a unitary select-U defined in the following way:

$$\text{select-}U := \sum_{k=1}^{K} |k\rangle\langle k| \otimes U_k. \tag{2.3.22}$$

Let us additionally suppose that we have access to a unitary U_A that can efficiently prepare the following state:

$$U_A|0\rangle = |A\rangle := \frac{1}{\sqrt{\sum_{k=1}^{K} c_k}} \sum_{k=1}^{K} \sqrt{c_k}|k\rangle. \tag{2.3.23}$$

Given access to the unitaries $\{U_k\}_{k=1}^K$ and their Hermitian conjugates $\{U_k^\dagger\}_{k=1}^K$ (for realizing the unitaries select-U and select-$U^\dagger$), as well as unitaries U_A and $U_A^\dagger$, we can use the standard linear combination of unitaries (LCU) technique [CW12] to prepare the following state:

$$|\phi\rangle := \frac{1}{\sqrt{c}} \sum_{k=1}^{K} c_k|\psi_k\rangle = \frac{1}{\sqrt{c}}(L \otimes I)|\Gamma\rangle, \tag{2.3.24}$$

where $c := \|L\|_2^2$. Due to the fact that the LCU technique is inherently probabilistic, we use the amplitude amplification (AA) method [BHMT02] to boost the probability of success to one. We do not go into specifics here because LCU and AA are both fairly common quantum algorithmic primitives. However, for completeness, we show in Figure 2.4 the exact quantum circuit that implements this method for preparing our desired state given by (2.3.24). Please refer to [Kot14, Section 2.2] and [Cha23, Section II-B] for more precise information on the LCU method and using AA to boost the success probability. Overall, we prepare n copies of this state for our purposes.

Now, we propose a quantum algorithm for simulating the quantum channel $e^{\mathcal{L}t}$, which corresponds to the Lindbladian in (2.3.19), using n copies of the program state $\phi := |\phi\rangle\langle\phi|$ up to error ε in diamond distance. As before, for providing an analysis related to the diamond distance, the input is an unknown quantum state $\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$ on joint system RS, with R as a reference system. Furthermore, let the i^{th} copy of the program state ϕ be a quantum state of the joint system $P_i Q_i$.

Algorithm 3 — Set $n \in \mathbb{N}$, with the particular choice specified later. Set $i = 1$. Given the i^{th} copy of ϕ, i.e., $\phi_{P_i Q_i}$, perform the following two steps:

1. Evolve the joint quantum state $\rho_{RS} \otimes \phi_{P_i Q_i}$ according to the following Lindbladian $\mathcal{M}$ acting on $S P_i Q_i$, for some small duration of time $\Delta = ct/n$:

$$\mathcal{M}(\cdot) := M(\cdot)M^\dagger - \frac{1}{2}\left\{M^\dagger M, (\cdot)\right\}. \tag{2.3.25}$$

   In the above, the Lindblad operator M acts on the joint system $S P_i Q_i$, and we define it as

$$M := \frac{1}{\sqrt{d}} \left(I_S \otimes |\Gamma\rangle\langle\Gamma|_{P_i Q_i}\right) \left(\text{SWAP}_{S P_i} \otimes I_{Q_i}\right). \tag{2.3.26}$$

2. Trace out systems $P_i Q_i$.

We repeat the above procedure using each copy of ϕ, i.e., for all i ranging from 1 to n.

The following theorem states that the above algorithm uses $n = O(d^2 c^2 t^2/\varepsilon)$ copies of ϕ to simulate the Lindbladian evolution of ρ_{RS}, according to the Lindbladian in (2.3.19), for time t, such that the final state is ε-close in normalized trace distance to the ideal target state $\left(\mathcal{I}_R \otimes e^{\mathcal{L}t}\right)(\rho_{RS})$, for an arbitrary input state ρ_{RS}. For preparing one copy of ϕ, we need to make

$$O\left(\frac{1}{\sqrt{c}} \sum_{k=1}^{K} c_k\right) \tag{2.3.27}$$

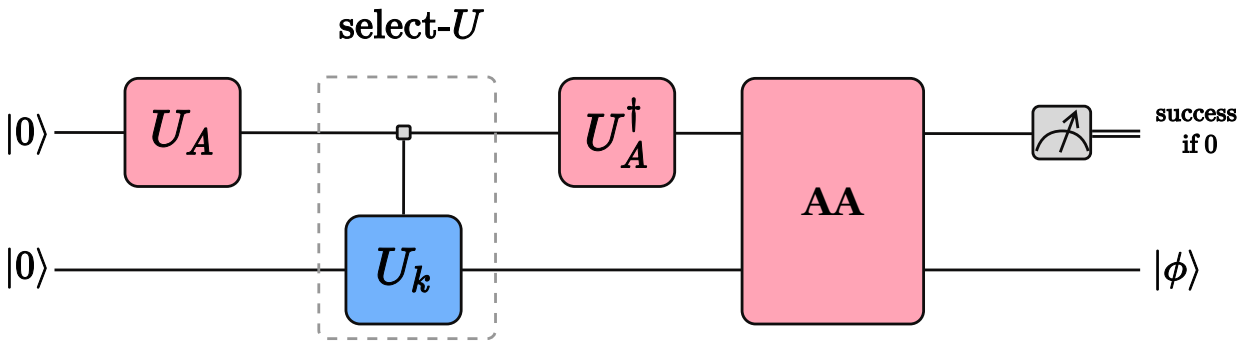


Figure 2.4: A quantum circuit for preparing $|\phi\rangle$ using the technique of linear combination of unitaries. The rectangle labelled "AA" represents multiple rounds of amplitude amplification.

queries to the unitaries $\{U_k\}_{k=1}^K$, $\{U_k^\dagger\}_{k=1}^K$, U_A, and $U_A^\dagger$ [Cha23, Section II-B]. Therefore, for preparing $O(d^2c^2t^2/\varepsilon)$ copies of ϕ, we need to make

$$O\left(\frac{d^2}{\varepsilon}\left(\sum_{k=1}^K c_k\right)c^{3/2}t^2\right) \tag{2.3.28}$$

queries to these unitaries.

Theorem 11 *Given access to n copies of the program state $\phi \in \mathcal{D}(\mathcal{H}_P \otimes \mathcal{H}_Q)$, defined in (2.3.24), each of which can be prepared using unitaries $\{U_k\}_{k=1}^K$, $\{U_k^\dagger\}_{k=1}^K$, U_A, and $U_A^\dagger$, there exists a quantum algorithm $\mathcal{A}$ such that the following error bound holds:*

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{A}\right\|_\diamond \leq \varepsilon, \tag{2.3.29}$$

with only $n = O(d^2c^2t^2/\varepsilon)$ copies of ϕ, where $c = \|L\|_2^2$ and d is the dimension of the system S. In other words, $\mathcal{A}$ uses only $n = O(d^2c^2t^2/\varepsilon)$ copies of ϕ to approximate the channel $e^{\mathcal{L}t}$ up to ε error in normalized diamond distance.

Proof. In what follows, we give a brief sketch of the proof. We will not go into detail because the proof follows a similar line of reasoning as the proof of Theorem 1 of [PW23a].

We leave out system labels for clarity and simplicity, and let us also suppose that the input state ρ does not have a reference system R for the purposes of this proof sketch.

Let us begin by expanding the target state $e^{\mathcal{L}t}(\rho)$ at $t = 0$ using its Taylor series:

$$e^{\mathcal{L}t}(\rho) = \rho + \mathcal{L}(\rho)t + \frac{1}{2}(\mathcal{L} \circ \mathcal{L})(\rho)t^2 + \ldots. \tag{2.3.30}$$

In the first step of Algorithm 3, we simulate the Lindbladian evolution of $\rho \otimes \phi$, given by (2.3.25), with the Lindbladian $\mathcal{M}$ for some small duration of time Δ, and we then trace out ϕ. The output state obtained after this step is

$$\mathrm{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho \otimes \phi)\right] = \rho + \mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \phi)]\,\Delta + O\left(\|\mathcal{M}\|_\diamond^2\,\Delta^2\right). \tag{2.3.31}$$

We first expand the second term and rewrite it in the following way:

$$\mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \phi)] = \mathrm{Tr}_{23}\left[M(\rho \otimes \phi)M^\dagger\right] - \frac{1}{2}\,\mathrm{Tr}_{23}\left[M^\dagger M\,(\rho \otimes \phi)\right] - \frac{1}{2}\,\mathrm{Tr}_{23}\left[(\rho \otimes \phi)\,M^\dagger M\right]. \tag{2.3.32}$$

Substituting $\phi = \frac{1}{c}(L \otimes I)|\Gamma\rangle\langle\Gamma|(L^\dagger \otimes I)$ into the above equation, we get

$$\begin{aligned}
\mathrm{Tr}_{23}[\mathcal{M}(\rho \otimes \phi)] = \frac{1}{c}\Bigg(& \mathrm{Tr}_{23}\left[M\left(\rho \otimes (L \otimes I)|\Gamma\rangle\langle\Gamma|(L^\dagger \otimes I)\right)M^\dagger\right] \\
& - \frac{1}{2}\,\mathrm{Tr}_{23}\left[M^\dagger M\left(\rho \otimes (L \otimes I)|\Gamma\rangle\langle\Gamma|(L^\dagger \otimes I)\right)\right] \\
& - \frac{1}{2}\,\mathrm{Tr}_{23}\left[\left(\rho \otimes (L \otimes I)|\Gamma\rangle\langle\Gamma|(L^\dagger \otimes I)\right)M^\dagger M\right]\Bigg) && (2.3.33) \\
= \frac{1}{c}\left(L\rho L^\dagger - \frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L\right) && (2.3.34) \\
= \frac{1}{c}\mathcal{L}(\rho). && (2.3.35)
\end{aligned}$$

We use tensor-network diagrams, as shown in Figures 1.3, 1.4, 1.5, and 1.6 of the previous chapter, to obtain (2.3.34) from (2.3.33) (alternatively, see Appendix 1.A).

Using (2.3.35), we rewrite (2.3.31) as

$$\mathrm{Tr}_{23}\left[e^{\mathcal{M}\Delta}(\rho \otimes \phi)\right] = \rho + \frac{1}{c}\mathcal{L}(\rho)\Delta + O\left(\|\mathcal{M}\|_\diamond^2\,\Delta^2\right) \tag{2.3.36}$$

$$= e^{(\mathcal{L}/c)\Delta}(\rho) + O\left(\|\mathcal{M}\|_\diamond^2 \Delta^2\right). \tag{2.3.37}$$

Substituting $\Delta = ct/n$, $\|\mathcal{M}\|_\diamond \leq 2d$ (see Appendix 1.B of Chapter 1 for more details on this inequality), and repeating Algorithm 3 for $n = O(d^2c^2t^2/\varepsilon)$ times produces a quantum state that is ε-close to the ideal target state $e^{\mathcal{L}t}(\rho)$ in normalized trace distance. ■

When Lindblad Operator is Local

In many practical settings, the Lindblad operator is local, that is, it acts only on a constant number of qubits. Let the system decompose as $S = A\bar{A}$, where A is the constant-size subsystem on which L acts nontrivially and $\bar{A}$ is the subsystem of the remaining qubits. In this case, we can write

$$L = L_A \otimes I_{\bar{A}}, \qquad d_A \coloneqq \dim(\mathcal{H}_A) = O(1). \tag{2.3.38}$$

Recall from (2.3.20) that L is given as a linear combination

$$L \coloneqq \sum_{k=1}^{K} c_k \, (L_{k,A} \otimes I_{\bar{A}}), \tag{2.3.39}$$

where $c_k > 0$ and $\left\|L_{k,A}\right\|_2 = 1$ for all $k \in \{1, \ldots, K\}$. Because the operator L is local, we perform the fixed Lindbladian evolution (Step 1 of Algorithm 3) using a localized operator that acts only on A and the program registers. Let the program registers be denoted $P^{(A)}$ and $Q^{(A)}$, and define the maximally entangled vector as

$$|\Gamma_A\rangle \coloneqq \sum_{j=1}^{d_A} |j\rangle \otimes |j\rangle. \tag{2.3.40}$$

The localized Lindblad operator is then

$$M_A \coloneqq \frac{1}{\sqrt{d_A}} \Big(I_{RS} \otimes |\Gamma_A\rangle\langle\Gamma_A|_{P^{(A)}Q^{(A)}}\Big)\Big(I_{R\bar{A}} \otimes \mathsf{SWAP}_{AP^{(A)}} \otimes I_{Q^{(A)}}\Big), \tag{2.3.41}$$

with the corresponding fixed Lindbladian

$$\mathcal{M}_A(X) \coloneqq M_A X M_A^\dagger - \tfrac{1}{2}\{M_A^\dagger M_A,\, X\}. \tag{2.3.42}$$

As in the global case, one step of duration $\Delta = ct/n$ evolves $\rho \otimes \phi$ by $e^{\mathcal{M}_A \Delta}$ and then we trace out the program registers $P^{(A)}Q^{(A)}$.

Theorem 12 (Local linear combination) *Suppose L is given as a linear combination $L = \sum_{k=1}^{K} c_k L_{k,A} \otimes I_{\bar{A}}$ with $c_k > 0$, $\|L_{k,A}\|_2 = 1$, and $d_A = \dim(\mathcal{H}_A) = O(1)$. Given access to n copies of the program state $\phi \in \mathcal{D}(\mathcal{H}_{P^{(A)}} \otimes \mathcal{H}_{Q^{(A)}})$ that encodes L_A as in (2.3.24), there exists a quantum algorithm $\mathcal{A}_{\mathrm{loc}}$ such that*

$$\frac{1}{2}\left\| e^{\mathcal{L}t} - \mathcal{A}_{\mathrm{loc}} \right\|_\diamond \le \varepsilon, \tag{2.3.43}$$

using only

$$n \;=\; O\!\left(d_A^2\, \frac{c^2 t^2}{\varepsilon}\right) \;=\; O\!\left(\frac{c^2 t^2}{\varepsilon}\right), \tag{2.3.44}$$

copies of ϕ, where the hidden constant depends on d_A (and thus is independent of the global system size d).

Proof. The proof is structurally identical to that of Theorem 11, but restricted to the subsystem A. In particular, replacing M with the localized operator M_A from (2.3.41), the same tensor-network contractions yield

$$\mathrm{Tr}_{P^{(A)}Q^{(A)}}[\mathcal{M}_A(\rho \otimes \phi)] = \tfrac{1}{c}\,\mathcal{L}(\rho),$$

where $\mathcal{L}$ is defined in (2.3.19). The per-step truncation error is $O(\|\mathcal{M}_A\|_\diamond^2\, \Delta^2)$; since $\|\mathcal{M}\|_\diamond \le 2d_A = O(1)$, this is $O(\Delta^2)$. Choosing $n = O(c^2 t^2/\varepsilon)$ ensures total error $O(\varepsilon)$, as claimed. ∎

Remark 13 *In this local Lindblad case, the number of queries to the ϕ-preparing unitaries $\{U_k\}_{k=1}^K$, $\{U_k^\dagger\}_{k=1}^K$, U_A, and $U_A^\dagger$ is now independent of the system dimension, and is given by*

$$O\left(\frac{1}{\varepsilon}\left(\sum_{k=1}^{K} c_k\right) c^{3/2} t^2\right). \tag{2.3.45}$$

2.3.3 Simulating Lindbladian Polynomials

In this section, we consider the case of simulating a Lindbladian evolution with a single Lindblad operator L and no Hamiltonian term, as in the previous section (see (2.3.19)). Here, we suppose that the Lindblad operator L is represented as a polynomial of linear operators encoded in program states.

To be more specific, let us suppose that L can be decomposed as follows:

$$L \coloneqq \sum_{s \in \mathcal{S}} c_s T_s, \tag{2.3.46}$$

where $c_s > 0$ and each term T_s, defined as follows, is of degree $|s|$:

$$T_s = L_{s[1]} L_{s[2]} \cdots L_{s[|s|]}. \tag{2.3.47}$$

Here, we define $\mathcal{S}$ as a set of strings over an alphabet $\{1, 2, \ldots, K\}$. Let the notation $s[i]$ denote the i^{th} character of the string s. Furthermore, we use the notation $|s|$ to denote the length of the string s, and we use the notation D to denote the degree of the polynomial given by (2.3.46), i.e., $D \coloneqq \max_{s \in \mathcal{S}}\{|s|\}$. In what follows, we assume that $D = O(1)$. The operators $L_1, \ldots, L_K$ are encoded in the program states $\psi_1, \ldots, \psi_K$, respectively, in the same way that was considered previously in (2.2.4), so that $\|L_1\|_2 = \cdots = \|L_K\|_2 = 1$. For this case as well, let us suppose that we have access to unitaries $U_1, \ldots, U_K$ that prepare these program states by acting on the maximally entangled state $|\Phi\rangle$ as follows:

$$U_k |\Phi\rangle = |\psi_k\rangle. \tag{2.3.48}$$

Given efficient implementations of the above unitaries, we can efficiently implement a unitary select-W defined as:

$$\text{select-}W \coloneqq \sum_{i=0}^{|\mathcal{S}|-1} |i\rangle\langle i| \otimes W_{s_i}, \tag{2.3.49}$$

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$ and s_i is the i^{th} string of the set $\mathcal{S}$ (the order is irrelevant for indexing purposes) and

$$W_{s_i} \coloneqq U_{s_i[1]} \otimes U_{s_i[2]} \otimes \cdots \otimes U_{s_i[|s|]} \otimes I^{\otimes D-|s|}. \tag{2.3.50}$$

Moreover, let us suppose that we have access to a unitary U_A that can efficiently prepare a pure quantum state defined as follows:

$$|A\rangle \coloneqq \frac{1}{\sqrt{\sum_{i=0}^{|\mathcal{S}|-1} c_{s_i}}} \sum_{i=0}^{|\mathcal{S}|-1} \sqrt{c_{s_i}}|i\rangle. \tag{2.3.51}$$

Given access to the set $\{U_k\}_{k=1}^K$ of unitaries and their Hermitian conjugates $\{U_k^\dagger\}_{k=1}^K$ (for realizing unitaries select-W and select-$W^\dagger$), as well as unitaries U_A and $U_A^\dagger$, we can prepare the following state using the LCU technique:

$$|\phi\rangle \coloneqq \frac{1}{\sqrt{c}} \sum_{s\in\mathcal{S}} c_s|\phi_s\rangle, \tag{2.3.52}$$

where

$$|\phi_s\rangle \coloneqq |\psi_{s[1]}\rangle|\psi_{s[2]}\rangle \cdots |\psi_{s[|s|]}\rangle \underbrace{|\Phi\rangle \cdots |\Phi\rangle}_{D-|s| \text{ times}}, \tag{2.3.53}$$

$$c \coloneqq \left\| \sum_{s\in\mathcal{S}} c_s|\phi_s\rangle \right\|_2^2. \tag{2.3.54}$$

As stated before in the previous section, the LCU technique is inherently probabilistic. As a result, we employ the amplitude amplification method to boost its probability of success to nearly one. Figure 2.5 depicts the quantum circuit that implements this technique for preparing our desired state given by (2.3.52). Overall, we prepare n copies of this state for our purposes.

We are now in a position to propose a quantum algorithm for simulating the quantum channel $e^{\mathcal{L}t}$, which corresponds to the Lindbladian with a single Lindblad operator given

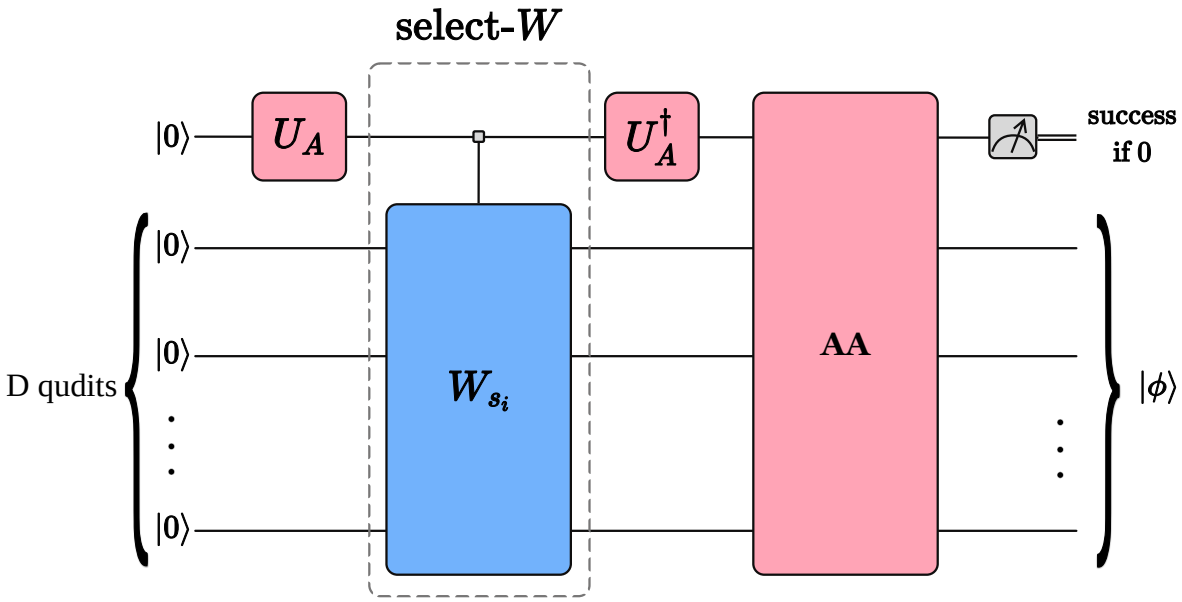


Figure 2.5: A quantum circuit for preparing $|\phi\rangle$ using the technique of linear combination of unitaries. The rectangle labelled "AA" represents multiple rounds of amplitude amplification.

by (2.3.46), using n copies of the program state $\phi := |\phi\rangle\langle\phi|$ up to error ε in diamond distance. As previously stated, in order to provide an analysis of the diamond distance, let $\rho \in \mathcal{D}(\mathcal{H}_R \otimes \mathcal{H}_S)$ be an unknown quantum state given as input over the joint system RS, where the system R acts as a reference system. Also, let the i^{th} copy of the program state ϕ be a quantum state over a joint system $P_i^1 Q_i^1 \cdots P_i^D Q_i^D$, i.e., $\phi \in \mathcal{D}(\mathcal{H}_{P_i^1} \otimes \mathcal{H}_{Q_i^1} \otimes \cdots \otimes \mathcal{H}_{P_i^D} \otimes \mathcal{H}_{Q_i^D})$. For brevity, we use $(PQ)_i^D$ as a shorthand for $P_i^1 Q_i^1 \cdots P_i^D Q_i^D$.

Algorithm 4 — Set $n \in \mathbb{N}$, with a particular choice specified later. Set $i = 1$. Given the i^{th} copy of ϕ, denoted by $\phi_{(PQ)_i^D}$, perform the following two steps:

1. Evolve the joint quantum state $\rho_{RS} \otimes \phi_{(PQ)_i^D}$ according to the following Lindbladian

$\mathcal{M}$ acting on systems $S(PQ)_i^D$, for some small duration of time $\Delta = ct/n$:

$$\mathcal{M}(\cdot) := M(\cdot)M^\dagger - \frac{1}{2}\left\{M^\dagger M, (\cdot)\right\}, \tag{2.3.55}$$

where the Lindblad operator M acts on the joint system $S(PQ)_i^D$, and we define it as

$$M := \frac{1}{d^{D/2}}\left(I_S \otimes \left(|\Gamma\rangle\langle\Gamma|^{\otimes D}\right)_{(PQ)_i^D}\right)\left(\text{CYCSWAP}_{S P_i^1 \cdots P_i^D} \otimes I_{Q_i^1 \cdots Q_i^D}\right). \tag{2.3.56}$$

2. Trace out the systems $(PQ)_i^D$.

We repeat the above procedure with each copy of ϕ, for i ranging from 1 to n.

According to the following theorem, the algorithm above uses $n = O(d^{O(1)}c^2t^2/\varepsilon)$ copies of ϕ to simulate the Lindbladian evolution of ρ_{RS}, given by (2.3.19), for time t, such that the final state is ε-close to the target state $\left(\mathcal{I}_R \otimes e^{\mathcal{L}t}\right)(\rho_{RS})$ in normalized trace distance, for an arbitrary input state ρ_{RS}. For preparing one copy of ϕ, we need to make

$$O\left(\frac{1}{\sqrt{c}}\sum_{i=0}^{|S|-1} c_{s_i}\right) \tag{2.3.57}$$

queries to the unitaries $\{U_k\}_{k=1}^K$, $\{U_k^\dagger\}_{k=1}^K$, U_A, and $U_A^\dagger$ [Cha23, Section II-B]. Therefore, for preparing $O(d^{O(1)}c^2t^2/\varepsilon)$ copies of ϕ, we need to make

$$O\left(\frac{d^{O(1)}}{\varepsilon}\left(\sum_{i=0}^{|S|-1} c_{s_i}\right)c^{3/2}t^2\right) \tag{2.3.58}$$

queries to these unitaries.

Theorem 14 *Given access to n copies of the program state $\phi \in \mathcal{D}(\mathcal{H}_{P_i^1} \otimes \mathcal{H}_{Q_i^1} \otimes \cdots \otimes \mathcal{H}_{P_i^D} \otimes \mathcal{H}_{Q_i^D})$, each of which can be prepared using unitaries $\{U_k\}_{k=1}^K$, $\{U_k^\dagger\}_{k=1}^K$, U_A, and $U_A^\dagger$, there exists a quantum algorithm $\mathcal{A}$ such that the following error bound holds:*

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{A}\right\|_\diamond \leq \varepsilon, \tag{2.3.59}$$

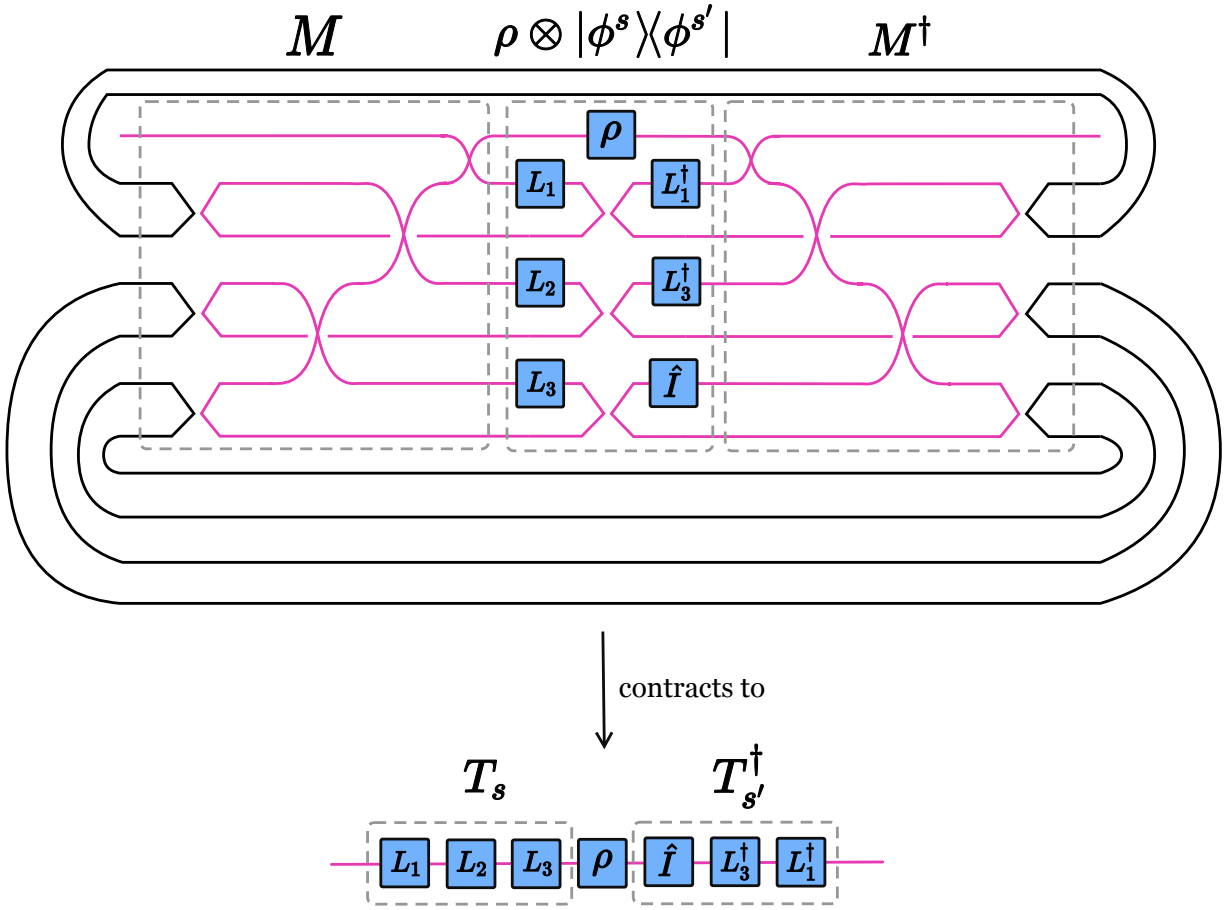


Figure 2.6: Tensor-network diagram of $\mathrm{Tr}_{2,\ldots,7}\left[M\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)M^\dagger\right]$, where $s = 123$ and $s' = 13$. The whole network on the top contracts to the network on the bottom as a result of tracing out registers $2, \ldots, 7$. The pink line flowing from the left end of the first system to the right end illustrates how the networks are connected after the partial trace operation.

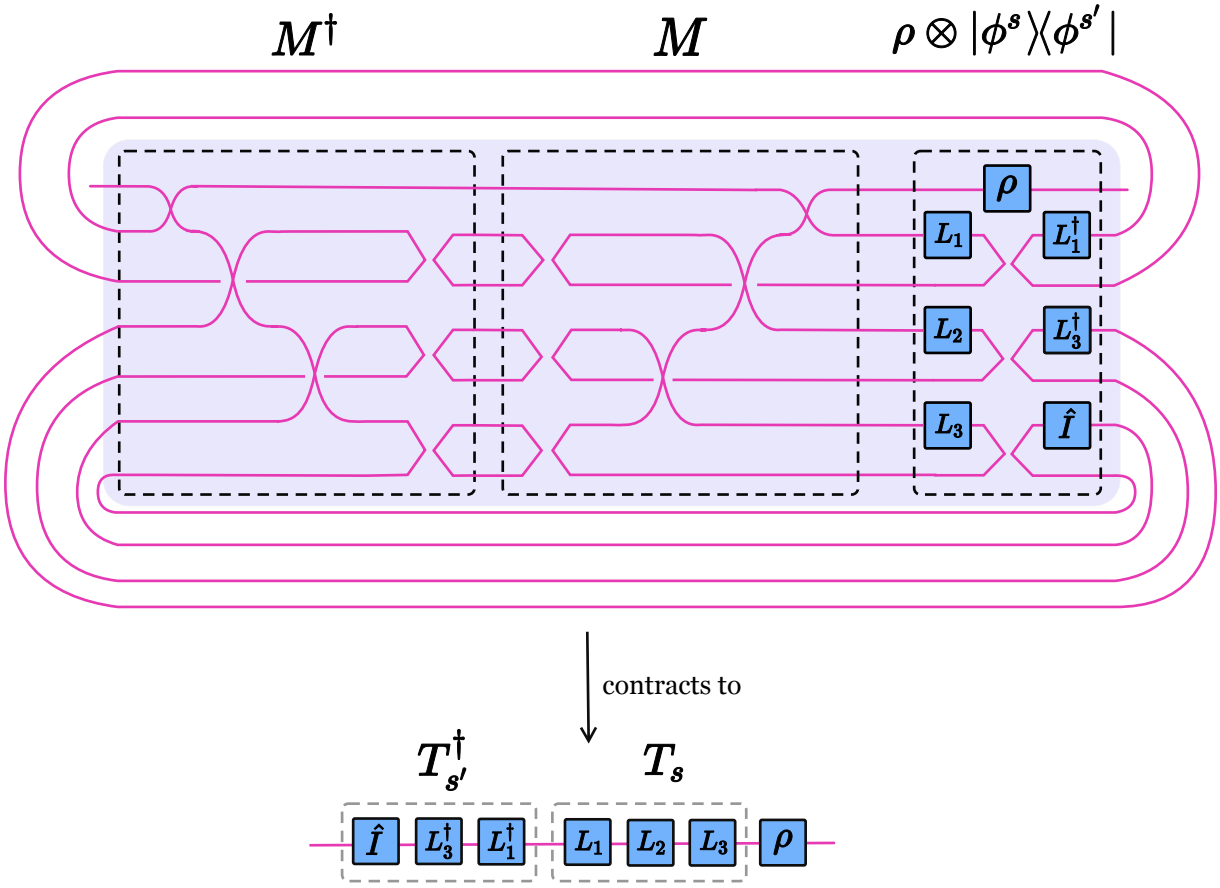


Figure 2.7: Tensor-network diagram of $\mathrm{Tr}_{2,\ldots,7}\left[M^\dagger M\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)\right]$, where $s = 123$ and $s' = 13$. The whole network on the top contracts to the network on the bottom as a result of tracing out registers $2, \ldots, 7$. The tensor-network diagram enclosed in the light-purple box depicts $M^\dagger M\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)$. The lines flowing out and back in represent the trace-out operation being performed on this network. Finally, in order to visualize how the network on the top contracts to that on the bottom after the trace-out operation, simply follow the pink line flowing from the left end of the first system to its right end.

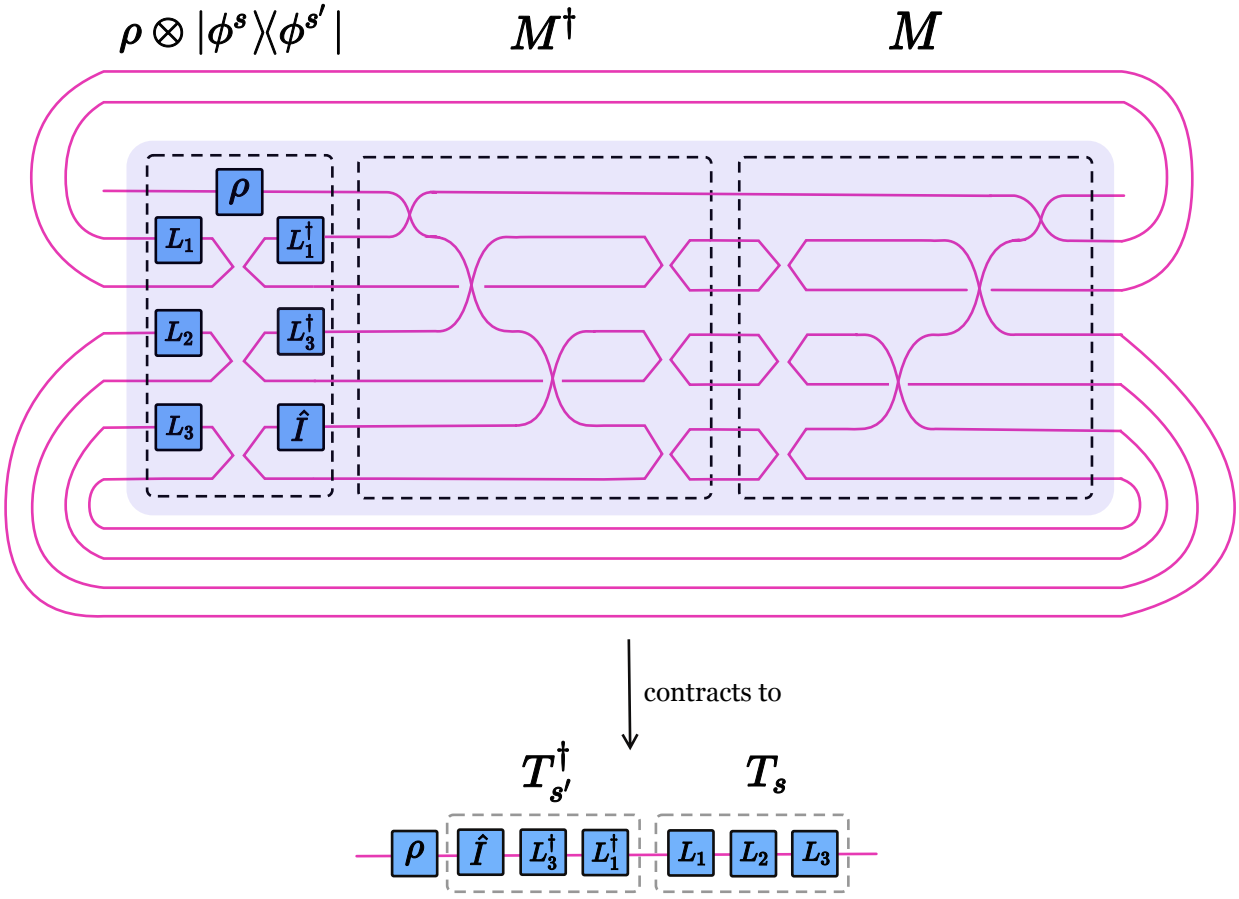


Figure 2.8: Tensor-network diagram of $\mathrm{Tr}_{2,\ldots,7}\left[\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right) M^\dagger M\right]$, where $s = 123$ and $s' = 13$. The whole network on the top contracts to the network on the bottom as a result of tracing out registers $2, \ldots, 7$. The tensor-network diagram enclosed in the light-purple box depicts $\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right) M^\dagger M$. The lines flowing out and back in represent the trace-out operation being performed on this network. Finally, in order to visualize how the network on the top contracts to that on the bottom after the trace-out operation, simply follow the pink line flowing from the left end of the first system to its right end.

with only $n = O(d^{O(1)}c^2t^2/\varepsilon)$ *copies of* ϕ*, where* c *is defined in* (2.3.54)*. In other words,* $\mathcal{A}$ *uses only* $n = O(d^{O(1)}c^2t^2/\varepsilon)$ *copies of* ϕ *to approximate the channel* $e^{\mathcal{L}t}$ *up to* ε *error in normalized diamond distance.*

Proof. For ease of notation, we do not explicitly write system labels, and let us suppose that the input state ρ does not have a reference system R.

The key point here is to demonstrate that the following three equalities hold for all $s, s' \in \mathcal{S}$:

$$\mathrm{Tr}_{2,3,\ldots,2D+1}\left[M\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)M^\dagger\right] = T_s \rho T_{s'}^\dagger, \tag{2.3.60}$$

$$\mathrm{Tr}_{2,3,\ldots,2D+1}\left[M^\dagger M\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)\right] = T_s^\dagger T_{s'} \rho, \tag{2.3.61}$$

$$\mathrm{Tr}_{2,3,\ldots,2D+1}\left[\left(\rho \otimes |\phi^s\rangle\langle\phi^{s'}|\right)M^\dagger M\right] = \rho T_s^\dagger T_{s'}, \tag{2.3.62}$$

for M as defined in (2.3.56). Lemma 19 in Appendix 2.D provides a full statement and proof. One can alternatively employ tensor-network diagrams to establish these equalities. For visualizing these diagrams for a simple case with $s = 123$ and $s' = 13$, please refer to Figures 2.6, 2.7, and 2.8.

With the above equalities holding for all $s, s' \in \mathcal{S}$, then we can essentially use the same type of reasoning as in the proof of Theorem 11 to conclude that

$$\mathrm{Tr}_{2,3,\ldots,2D+1}[\mathcal{M}(\rho \otimes \phi)] = \frac{1}{c}\mathcal{L}(\rho). \tag{2.3.63}$$

This further implies that

$$\mathrm{Tr}_{2,3,\ldots,2D+1}\left[e^{\mathcal{M}\Delta}(\rho \otimes \phi)\right] = \rho + \mathrm{Tr}_{2,3,\ldots,2D+1}[\mathcal{M}(\rho \otimes \phi)]\,\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \tag{2.3.64}$$

$$= \rho + \frac{1}{c}\mathcal{L}(\rho)\Delta + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2) \tag{2.3.65}$$

$$= e^{(\mathcal{L}/c)\Delta}(\rho) + O(\|\mathcal{M}\|_\diamond^2\,\Delta^2). \tag{2.3.66}$$

Substituting $\Delta = ct/n, \|\mathcal{M}\|_\diamond \leq d^D = d^{O(1)}$ and repeating Algorithm 4 for $n = O(d^{O(1)}c^2t^2/\varepsilon)$ times produces a quantum state that is $O(\varepsilon)$-close to the ideal target state $e^{\mathcal{L}t}(\rho)$ in normalized trace distance. ■

When Lindblad Operator is Local

As motivated earlier, in many practical scenarios the Lindblad operator acts only on a constant-size subsystem. Let the system decompose as $S = A\bar{A}$, and assume that every base operator in the polynomial is local on A:

$$L_k \equiv L_{k,A} \otimes I_{\bar{A}}, \qquad d_A := \dim(\mathcal{H}_A). \tag{2.3.67}$$

Accordingly, each monomial T_s in (2.3.47) factors as

$$T_s = \left(L_{s[1],A} L_{s[2],A} \cdots L_{s[|s|],A}\right) \otimes I_{\bar{A}} =: T_{s,A} \otimes I_{\bar{A}}, \tag{2.3.68}$$

and the polynomial Lindblad operator is $L = (\sum_{s\in\mathcal{S}} c_s T_{s,A}) \otimes I_{\bar{A}}$.

We encode each $L_{k,A}$ in a local program state $\psi_{k,A}$, and prepare the polynomial program state exactly as in (2.3.52) but using local encodings:

$$|\phi_A\rangle := \frac{1}{\sqrt{c}} \sum_{s\in\mathcal{S}} c_s |\phi_{s,A}\rangle, \tag{2.3.69}$$

where

$$|\phi_{s,A}\rangle := |\psi_{s[1],A}\rangle |\psi_{s[2],A}\rangle \cdots |\psi_{s[|s|],A}\rangle \underbrace{|\Phi_A\rangle \cdots |\Phi_A\rangle}_{D-|s| \text{ times}}, \tag{2.3.70}$$

$$c := \left\| \sum_{s\in\mathcal{S}} c_s |\phi_{s,A}\rangle \right\|_2^2, \tag{2.3.71}$$

and $|\Phi_A\rangle$ is the maximally entangled state on two d_A-dimensional registers. Let the i^{th} copy of ϕ_A live on the program registers $(PQ)_i^{D,(A)} \equiv P_i^{1,(A)} Q_i^{1,(A)} \cdots P_i^{D,(A)} Q_i^{D,(A)}$.

To perform the fixed evolution step, we act only on A and the program registers, using the localized version of (2.3.56):

$$M_A \coloneqq \frac{1}{d_A^{D/2}} \left(I_{RS} \otimes (|\Gamma_A\rangle\langle\Gamma_A|^{\otimes D})_{(PQ)_i^{D,(A)}} \right) \times \left(I_{R\bar{A}} \otimes \mathrm{CYCSWAP}_{A\, P_i^{1,(A)} \cdots P_i^{D,(A)}} \otimes I_{Q_i^{1,(A)} \cdots Q_i^{D,(A)}} \right), \quad (2.3.72)$$

and define $\mathcal{M}_A(\cdot) \coloneqq M_A(\cdot)M_A^\dagger - \frac{1}{2}\{M_A^\dagger M_A, (\cdot)\}$. One Trotter step of duration $\Delta = ct/n$ applies $e^{\mathcal{M}_A \Delta}$ to $\rho \otimes \phi_A$ and then traces out the program registers $(PQ)_i^{D,(A)}$.

Theorem 15 (Local polynomial Lindblad operator) *Assume $D = O(1)$ and that every base operator is local on a constant-size block A, i.e., $L_k = L_{k,A} \otimes I_{\bar{A}}$ with $d_A = \dim(\mathcal{H}_A) = O(1)$. Given access to n copies of the program state ϕ_A defined above (prepared via the same LCU-and-amplitude-amplification construction as in (2.3.52), but locally on A), there exists a quantum algorithm $\mathcal{A}_{\mathrm{loc}}$ such that*

$$\frac{1}{2} \left\| e^{\mathcal{L}t} - \mathcal{A}_{\mathrm{loc}} \right\|_\diamond \leq \varepsilon, \quad (2.3.73)$$

using only

$$n \;=\; O\left(d_A^{O(1)} \, \frac{c^2 t^2}{\varepsilon} \right) \;=\; O\left(\frac{c^2 t^2}{\varepsilon} \right) \quad (2.3.74)$$

copies of ϕ_A, where c is given in (2.3.54).

Proof. The argument mirrors the global proof with all actions restricted to the subsystem A. The key contractions (provable via the localized analog of Lemma 19 or tensor-network diagrams) give, for all $s, s' \in \mathcal{S}$,

$$\mathrm{Tr}_{(PQ)^{D,(A)}}[M_A(\rho \otimes |\phi_{s,A}\rangle\langle\phi_{s',A}|)M_A^\dagger] = (T_{s,A} \otimes I_{\bar{A}})\, \rho\, (T_{s',A}^\dagger \otimes I_{\bar{A}}),$$

and similarly for the two anticommutator terms, which yields $\mathrm{Tr}_{(PQ)^{D,(A)}}[\mathcal{M}_A(\rho \otimes \phi_A)] = \frac{1}{c}\mathcal{L}(\rho)$. The per-step truncation error is $O(\|\mathcal{M}_A\|_\diamond^2 \, \Delta^2)$ with $\|\mathcal{M}_A\|_\diamond \leq d_A^D = d_A^{O(1)}$. Choosing $n = O(d_A^{O(1)} c^2 t^2/\varepsilon)$ makes the total error $O(\varepsilon)$. ∎

Remark 16 *All tensor-network diagrams in Figures 2.6–2.8 remain valid after relabeling the system wire as A; wires corresponding to $\bar{A}$ and R simply pass through as identities.*

2.3.4 Wave Matrix Lindbladization Improves upon Tomography

In this section, we provide a comparison between two different methods of simulating the channel $e^{\mathcal{L}t}$, either by WML or by performing tomography to determine a classical description of the Lindbladian $\mathcal{L}$, followed by simulation from this description. The main finding, summarized in Figure 2.9, is that WML achieves a quadratic improvement in sample complexity with respect to the approximation error, thereby offering a more efficient route to simulating $e^{\mathcal{L}t}$ than approaches based on tomography. Let us note that a similar comparison was made between density matrix exponentiation and state tomography in [KLL+17, Section 2], demonstrating the advantage of the former over the latter in terms of the sample complexity required for sample-based Hamiltonian simulation.

Beyond efficiency, this quadratic improvement suggests a form of quantum copy-protection, as introduced in [Aar09]: a party could send sufficiently many program states to another party, enabling them to simulate Lindblad dynamics without revealing details of the encoded operator. In this way, our results also contribute to addressing the open question posed in [KLL+17] concerning what other quantum operations, besides Hamiltonian simulation, can be encoded in quantum states and executed privately.

To set up the comparison, let us consider a Lindbladian $\mathcal{L}$ with a single Lindblad operator L, as in (2.3.19), and aim to simulate the corresponding channel $e^{\mathcal{L}t}$ for time t. We assume that L is provided upfront in the form of a program state $|\psi\rangle$, as in (2.2.4), and that n copies of this state are available. For the analysis in this section, we make no locality

assumptions and instead treat the worst-case scenario where L acts on the full Hilbert space of dimension d. Nevertheless, the results easily extend to the more practical local setting: in that case, d is replaced by the dimension d_A of the subsystem A on which L acts, and since $d_A = O(1)$, the bounds simplify accordingly.

As mentioned above, one naive way to implement the channel $e^{\mathcal{L}t}$ is to first obtain the full classical description of the encoded Lindblad operator L by using multiple copies of the state $|\psi\rangle$. By classical description, we mean the complete matrix representation of L. Once we have such a description of L, we can then use known algorithms [CL17, CW17, MOTT22, SHMS+22, SBH+23] to actually implement the channel $e^{\mathcal{L}t}$ on a quantum computer. However, an important point to note here is that obtaining an exact classical description of an arbitrary operator encoded in a pure state may not be computationally efficient in general. Furthermore, obtaining an operator that is "close" to the encoded operator is sufficient for many practical purposes. As a result, we look into a slightly relaxed version of the above problem, in which the task is to output the full classical description of an operator $\tilde{L}$ that is δ-close to L in the Hilbert–Schmidt distance measure, given n copies of $|\psi\rangle$. We further require that the operator $\tilde{L}$ satisfies the constraint $\left\|\tilde{L}\right\|_2 = 1$. We refer to this modified relaxed problem as *wave-matrix tomography*. More formally, the task is to output the full classical description of an operator $\tilde{L}$ such that $\left\|\tilde{L}\right\|_2 = 1$ and

$$\left\|\tilde{L} - L\right\|_2 \leq \delta, \tag{2.3.75}$$

by using multiple copies of the state $|\psi\rangle$.

Having said that, a natural question in the context of Lindbladian simulation is how well the quantum channel $e^{\tilde{\mathcal{L}}t}$ approximates the channel $e^{\mathcal{L}t}$, where $\tilde{\mathcal{L}}$ is the Lindbladian with a single Lindblad operator $\tilde{L}$. The following theorem states that if the two Lindblad operators $\tilde{L}$ and L are $\delta = O(\varepsilon/t)$-close in Hilbert–Schmidt distance, then the quantum

channel $e^{\tilde{\mathcal{L}}t}$ approximates the target channel $e^{\mathcal{L}t}$ up to $O(\varepsilon)$ error in diamond distance.

Theorem 17 *Let L and $\tilde{L}$ be two $d \times d$-dimensional linear operators such that $\|L\|_2 = \left\|\tilde{L}\right\|_2 = 1$ and*

$$\left\|\tilde{L} - L\right\|_2 = O(\varepsilon/t). \tag{2.3.76}$$

Then, the following holds:

$$\frac{1}{2}\left\|e^{\tilde{\mathcal{L}}t} - e^{\mathcal{L}t}\right\|_\diamond = O(\varepsilon), \tag{2.3.77}$$

where $\tilde{\mathcal{L}}$ and $\mathcal{L}$ are Lindbladians with Lindblad operators $\tilde{L}$ and L, respectively.

Proof. The proof is provided in Appendix 2.E. ■

We now turn to examining the sample complexity of wave-matrix tomography in order to compare it with our WML approach. The key question here is: how many copies of the state $|\psi\rangle$ are needed to solve the wave-matrix tomography problem?

To begin with, let us first look at the sample complexity of a related problem: pure-state tomography. In pure-state tomography, we are given n copies of a state $|\psi\rangle$, and we want to output a state $|\tilde{\psi}\rangle$ such that

$$\frac{1}{2}\left\||\tilde{\psi}\rangle\langle\tilde{\psi}| - |\psi\rangle\langle\psi|\right\|_1 \leq \delta, \tag{2.3.78}$$

where $\delta \in (0, 1]$. In order to solve the above problem, [HHJ+17, Theorem 3] establishes that it is necessary to use n copies of $|\psi\rangle$, where

$$n = \Omega\left(\frac{d^2(1-\delta)^2}{\delta^2 \log(d^2/\delta)}\right) \tag{2.3.79}$$

In the above expression, d^2 is the dimension of the underlying Hilbert space of $|\psi\rangle$, and so we see that the sample complexity is dimension-dependent. In addition, the Helstrom

bound [Hel69] implies a separate lower bound of $n = \Omega(1/\delta^2)$. Combining these two results gives

$$n = \Omega\left(\frac{d^2(1-\delta)^2}{\delta^2 \log(d^2/\delta)} + \frac{1}{\delta^2}\right). \tag{2.3.80}$$

Let us now connect pure-state tomography to wave-matrix tomography by reducing the former to the latter. In other words, we need to show that we can solve the pure-state tomography problem if we can solve the wave-matrix tomography problem. The primary reason for showing this reduction is to obtain a lower bound on the sample complexity of the wave-matrix tomography problem. Note that the input for both problems is the same, i.e., some number of copies of $|\psi\rangle$. On the contrary, their outputs are different. The output of the wave-matrix tomography problem is an operator $\tilde{L}$ that is δ-close to L in Hilbert–Schmidt distance (see (2.3.75)), while the output of pure-state tomography is a pure state $|\tilde{\psi}\rangle$ that is δ-close to $|\psi\rangle$ in normalized trace distance (see (2.3.78)). We now relate these outputs as follows. Let $|\tilde{\psi}\rangle$ be the following state:

$$|\tilde{\psi}\rangle = (\tilde{L} \otimes I)|\Gamma\rangle. \tag{2.3.81}$$

Then Lemma 18 below states that $|\tilde{\psi}\rangle$ is δ-close to $|\psi\rangle$ if L is δ-close to $\tilde{L}$ in Hilbert–Schmidt distance. As such, $|\tilde{\psi}\rangle$ is a solution to the pure-state tomography problem when the input is $|\psi\rangle$. This concludes the reduction of pure-state tomography to wave-matrix tomography. This further implies that the lower bound on the sample complexity of the pure-state tomography problem, given by (2.3.80), is also a lower bound on the sample complexity of the wave-matrix tomography problem.

Lemma 18 *Let L and $\tilde{L}$ be two $d \times d$-dimensional linear operators such that $\|L\|_2 = \left\|\tilde{L}\right\|_2 = 1$ and*

$$\left\|\tilde{L} - L\right\|_2 \leq \delta. \tag{2.3.82}$$

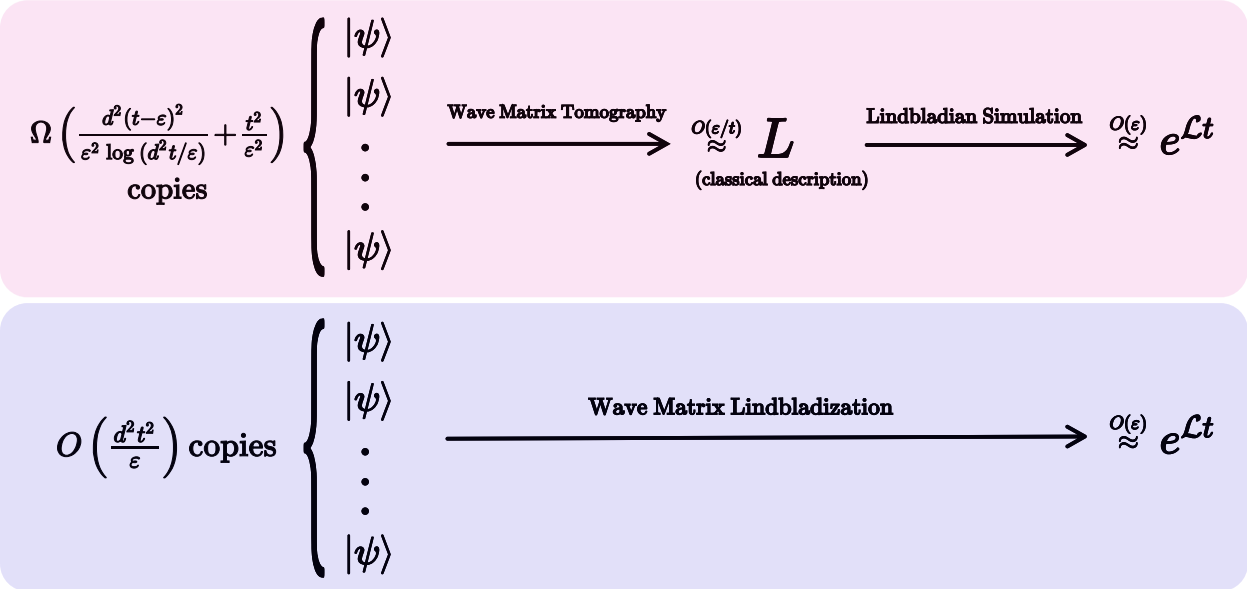


Figure 2.9: Comparing sample complexities of two different approaches for approximately simulating the channel $e^{\mathcal{L}t}$: (pink box) performing wave-matrix tomography to first obtain the classical description of the Lindblad operator L, followed by simulation from this description and (blue box) wave matrix Lindbladization.

Then,

$$\frac{1}{2}\left\||\tilde{\psi}\rangle\langle\tilde{\psi}| - |\psi\rangle\langle\psi|\right\|_1 \leq \delta, \tag{2.3.83}$$

where $|\tilde{\psi}\rangle = (\tilde{L} \otimes I)|\Gamma\rangle$ *and* $|\psi\rangle = (L \otimes I)|\Gamma\rangle$ *are pure states that encode* $\tilde{L}$ *and* L*, respectively.*

Proof. See Appendix 2.F. ■

According to Theorem 17, the quantum channel $e^{\tilde{\mathcal{L}}t}$ approximates the target channel $e^{\mathcal{L}t}$ up to $O(\varepsilon)$ error in diamond distance if $\delta = O(\varepsilon/t)$. Substituting this into (2.3.80), we get

$$n = \Omega\left(\frac{d^2(t-\varepsilon)^2}{\varepsilon^2 \log(d^2 t/\varepsilon)} + \frac{t^2}{\varepsilon^2}\right). \tag{2.3.84}$$

It is evident from above that wave-matrix tomography requires more copies of $|\psi\rangle$ to implement the channel $e^{\mathcal{L}t}$ than wave matrix Lindbladization. As shown in [PW23a, Theorem 1], our approach needs $O(d^2t^2/\varepsilon)$ copies of $|\psi\rangle$. Therefore, it is clear that there is a quadratic improvement in sample complexity with respect to the approximation error ε. Therefore, WML thus provides an efficient route for implementing $e^{\mathcal{L}t}$ without needing full tomography of L. Please refer to Figure 2.9 for a visual representation of the comparison between these two approaches.

2.4 Conclusion and Open Problems

In this chapter, we focused on the problem of simulating open system dynamics governed by the Lindblad master equation when the Lindblad operators are provided beforehand through encoding in program states. In the previous chapter, we investigated a relatively simple case in which the Lindbladian consists of a single Lindblad operator and a Hamiltonian. In this chapter, we extended this case to simulating general Lindbladians and other situations in which the Lindblad operator can be expressed as a linear combination or polynomial of the operators encoded across multiple program states. We proposed quantum algorithms for all these cases and investigated their sample complexities – the number of copies of program states needed. Finally, we showed that our quantum algorithms provide a more efficient route to simulating Lindbladian evolution when compared to full tomography of the encoded operators. We demonstrated this by proving that the sample complexity for tomography is dependent on the system dimension, while that for wave matrix Lindbladization is not.

In the previous chapter, we listed some open questions that we then managed to an-

swer in this chapter. However, one of the open problems about the optimality still remains unaddressed.

2.5 Acknowledgements

We are indebted to Andrew Childs, Tongyang Li, and Marina Radulaski for a helpful email exchange regarding [CL17, Proposition 2]. MMW is especially grateful to Prof. Ingemar Bengtsson for the opportunity to have met and discussed research with Prof. Lindblad in Stockholm, Sweden, during April 2019.

BIBLIOGRAPHY

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242, Paris, France, July 2009. doi:10.1109/CCC.2009.42.

[BB17] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell. arXiv:1708.00006, August 2017.

[BCC+14a] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pages 283–292, December 2014. doi:10.1145/2591796.2591854.

[BCC+14b] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, December 2014. doi:10.1103/PhysRevLett.114.090502.

[BCG14] Dominic W. Berry, Richard Cleve, and Sevag Gharibian. Gate-efficient discrete simulations of continuous-time quantum query algorithms. *Quantum Information and Computation*, 14(1–2):1–30, November 2014.

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, October 2015. doi:10.1109/FOCS.2015.54.

[Bei13] Salman Beigi. Sandwiched Rényi divergence satisfies data processing inequality. *Journal of Mathematical Physics*, 54(12):122202, 2013.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[BP02] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002.

[Cha23] Shantanav Chakraborty. Implementing any linear combination of unitaries on intermediate-term quantum computers, August 2023.

arXiv:2302.13555. arXiv:https://arxiv.org/abs/2302.13555, doi:10.48550/ARXIV.2302.13555.

[CL17] Andrew M. Childs and Tongyang Li. Efficient simulation of sparse Markovian quantum dynamics. *Quantum Information and Computation*, 17(11&12):901–947, November 2017. doi:10.26421/QIC17.11-12.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation*, 12(11-12):901–924, November 2012.

[CW17] Richard Cleve and Chunhao Wang. Efficient quantum algorithms for simulating Lindblad evolution. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.17.

[Dup10] Frédéric Dupuis. *The decoupling approach to quantum information theory*. PhD thesis, Université de Montréal, 2010. arXiv:1004.1641.

[GKS76] Vittorio Gorini, Andrzej Kossakowski, and E. C. G. Sudarshan. Completely positive dynamical semigroups of *N*-level systems. *Journal of Mathematical Physics*, 17(5):821–825, May 1976. doi:10.1063/1.522979.

[GZ04] Crispin Gardiner and Peter Zoller. *Quantum Noise: A Handbook of Markovian and Non-Markovian Quantum Stochastic Methods with Applications to Quantum Optics*. Springer, 2004.

[Hel69] Carl W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, June 1969. doi:10.1007/bf01007479.

[HHJ+17] Jeongwan Haah, Aram W. Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-optimal tomography of quantum states. *IEEE Transactions on Information Theory*, 63(9):5628–5641, September 2017. doi:10.1109/tit.2017.2719044.

[KB14] Michael J. Kastoryano and Fernando G. S. L. Brandão. Quantum Gibbs samplers: the commuting case. *Communications in Mathematical Physics*, 344(3):915–957, September 2014. doi:10.1007/s00220-016-2641-8.

[KBD+08] B. Kraus, H. P. Büchler, S. Diehl, A. Kantian, A. Micheli, and P. Zoller. Preparation of entangled states by quantum Markov processes. *Physical Review A*, 78(4):042307, October 2008. doi:10.1103/PhysRevA.78.042307.

[Kit97] A. Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. doi:10.1070/RM1997v052n06ABEH002155.

[KLL+17] Shelby Kimmel, Cedric Yen Yu Lin, Guang Hao Low, Maris Ozols, and Theodore J. Yoder. Hamiltonian simulation with optimal sample complexity. *npj Quantum Information*, 3(1):1–7, March 2017. doi:10.1038/s41534-017-0013-7.

[Kot14] Robin Kothari. *Efficient algorithms in quantum query complexity*. PhD thesis, University of Waterloo, School of Computer Science, August 2014. http://hdl.handle.net/10012/8625.

[KRS11] M. J. Kastoryano, F. Reiter, and A. S. Sørensen. Dissipative preparation of entanglement in optical cavities. *Physical Review Letters*, 106(9):090502, February 2011. doi:10.1103/PhysRevLett.106.090502.

[KSMM22] Hirsh Kamakari, Shi-Ning Sun, Mario Motta, and Austin J. Minnich. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. *PRX Quantum*, 3(1):010320, February 2022. doi:10.1103/PRXQuantum.3.010320.

[LC17] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, January 2017. doi:10.1103/PhysRevLett.118.010501.

[Lin76] Göran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, June 1976. doi:10.1007/BF01608499/METRICS.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, August 1996. doi:10.1126/science.273.5278.1073.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, July 2014. doi:10.1038/nphys3029.

[MK08] Volkhard May and Oliver Kühn. *Charge and Energy Transfer Dynamics in Molecular Systems*. John Wiley & Sons, 2008.

[MOTT22] Alexander Miessen, Pauline J. Ollitrault, Francesco Tacchino, and Ivano Tavernelli. Quantum algorithms for quantum dynamics. *Nature Computational Science*, 3(1):25–37, December 2022. doi:10.1038/s43588-022-00374-2.

[MPGC13] Easwar Magesan, Daniel Puzzuoli, Christopher E. Granade, and David G. Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Physical Review A*, 87(1):012324, January 2013. doi:10.1103/PhysRevA.87.012324.

[MTAB12] Daniel Manzano, Markus Tiersch, Ali Asadian, and Hans J. Briegel. Quantum transport efficiency and Fourier's law. *Physical Review E*, 86(6):061118, December 2012. doi:10.1103/PhysRevE.86.061118.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. 10th Anniversary Edition.

[Nit06] Abraham Nitzan. *Chemical Dynamics in Condensed Phases: Relaxation, Transfer and Reactions in Condensed Molecular Systems*. Oxford University Press, 2006.

[OLG12] Beatriz Olmos, Igor Lesanovsky, and Juan P. Garrahan. Facilitated spin models of dissipative quantum glasses. *Physical Review Letters*, 109(2):020403, July 2012. doi:10.1103/PHYSREVLETT.109.020403/FIGURES/5/MEDIUM.

[PK98] Martin B. Plenio and Peter L. Knight. The quantum-jump approach to dissipative dynamics in quantum optics. *Reviews of Modern Physics*, 70(1):101–144, January 1998. doi:10.1103/RevModPhys.70.101.

[Pro11] Tomaz Prosen. Open *XXZ* spin chain: Nonequilibrium steady state and a strict bound on ballistic transport. *Physical Review Letters*, 106(21):217206, May 2011. doi:10.1103/PhysRevLett.106.217206.

[PW23a] Dhrumil Patel and Mark M. Wilde. Wave matrix Lindbladization I: Quantum programs for simulating Markovian dynamics. *Open Systems and Information Dynamics*, 30(02):2350010, June 2023. doi:10.1142/s1230161223500105.

[PW23b] Dhrumil Patel and Mark M. Wilde. Wave matrix lindbladization ii: General lindbladians, linear combinations, and polynomials. *Open Systems and Information Dynamics*, 30(03):2350014, September 2023. doi:10.1142/s1230161223500142.

[RRS16] Florentin Reiter, David Reeb, and Anders S. Sørensen. Scalable dissipative preparation of many-body entanglement. *Physical Review Letters*, 117(4):040501, July 2016. doi:10.1103/PhysRevLett.117.040501.

[SBH+23] Nishchay Suri, Joseph Barreto, Stuart Hadfield, Nathan Wiebe, Filip Wudarski, and Jeffrey Marshall. Two-unitary decomposition algorithm and open quantum system simulation. *Quantum*, 7:1002, May 2023. doi:10.22331/q-2023-05-15-1002.

[SHMS+22] Anthony W. Schlimgen, Kade Head-Marsden, LeeAnn M. Sager, Prineha Narang, and David A. Mazziotti. Quantum simulation of the Lindblad equation using a unitary decomposition of operators. *Physical Review Research*, 4(2):023216, June 2022. doi:10.1103/PhysRevResearch.4.023216.

[SPP+25] Aidan N. Sims, Dhrumil Patel, Aby Philip, Alex H. Rubin, Rahul Bandyopadhyay, Marina Radulaski, and Mark M. Wilde. Digital quantum simulations of the non-resonant open tavis-cummings model, January 2025. arXiv:https://doi.org/10.48550/arXiv.2501.18522, doi:10.48550/ARXIV.2501.18522.

[VWC09] Frank Verstraete, Michael M. Wolf, and J. Ignacio Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5(9):633–636, 2009.

[Wei21] Ulrich Weiss. *Quantum Dissipative Systems*. World Scientific, 5th edition, 2021. arXiv:https://www.worldscientific.com/doi/pdf/10.1142/8334, doi:10.1142/8334.

[Wil17] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, Cambridge, UK, second edition, 2017. doi:10.1017/9781316809976.

APPENDIX

2.A Gate Complexity Analysis for Sampling-based Approach

In this section, we present a detailed analysis of the gate complexity of the sampling-based WML algorithm (see Section 2.3.1). We write this section as thoroughly and self-contained as possible, so that one can read this section independently of the main text.

The Lindbladian $\mathcal{L}$, as defined in (2.3.1), can be expressed as

$$\mathcal{L} = \sum_{j=1}^{J} c_j \mathcal{H}_j + \sum_{k=1}^{K} \left\|L_{A_k}\right\|_2^2 \widehat{\mathcal{L}}_k, \tag{2.A.1}$$

where

$$\mathcal{H}_j(\cdot) := -i\,[\sigma_j, (\cdot)], \qquad \widehat{\mathcal{L}}_k(\cdot) := \widehat{L}_k(\cdot)\widehat{L}_k^\dagger - \frac{1}{2}\{\widehat{L}_k^\dagger \widehat{L}_k, (\cdot)\}. \tag{2.A.2}$$

Here

$$\widehat{L}_k := \widehat{L}_{A_k} \otimes I_{\overline{A_k}}, \tag{2.A.3}$$

where $\widehat{L}_{A_k}$ is a normalized operator, that is, $\left\|\widehat{L}_{A_k}\right\|_2 = 1$. For the purpose of this section assume that each Lindblad operator L_k acts non-trivially on at most q qubits, with the identity acting on the remainder of the system. From the definition of L_k in (2.3.3), it follows that $d_{A_k} = 2^q =: Q$. We keep q as a parameter in the analysis to maintain generality. At the end, we substitute $q = O(1)$, since in our setting the Lindblad operators act on only a constant number of qubits.

Step 1 of the sampling-based WML algorithm is the sampling step where the state σ_j is sampled with probability $\frac{c_j}{c}$ (Case 1), the state σ_j is sampled with probability $\frac{(-c_j)}{c}$ (Case 2), and the state ψ_k is sampled with probability $\frac{\left\|L_{A_k}\right\|_2^2}{c}$ (Case 3), where c is defined in (2.3.6). Step 2 is simply initializing the program register with the sampled state. Note

that the system register is in the state ρ. Depending on the case, Step 2 can be represented as the following appending channels, defined for all $j \in [J]$ and $k \in [K]$:

$$\text{(Case 1 and Case 2):} \quad \mathcal{P}_{1,j}(\rho) \coloneqq \rho \otimes \sigma_j \tag{2.A.4}$$

$$\text{(Case 3):} \quad \mathcal{P}_{2,k}(\rho) \coloneqq \rho \otimes \psi_k. \tag{2.A.5}$$

Step 3 of the algorithm involves applying one of the following three quantum channels jointly to the system and program registers, also depending on the case:

$$\text{(Case 1):} \quad e^{\mathcal{N}_1 c\tau}(\rho \otimes \sigma_j) \tag{2.A.6}$$

$$\text{(Case 2):} \quad e^{\mathcal{N}_2 c\tau}(\rho \otimes \sigma_j) \tag{2.A.7}$$

$$\text{(Case 3):} \quad e^{\mathcal{M}_k c\tau}(\rho \otimes \psi_k), \tag{2.A.8}$$

where $\tau \coloneqq t/n$,

$$\mathcal{N}_1(\cdot) \coloneqq -i[\mathrm{SWAP}, \cdot] \tag{2.A.9}$$

$$\mathcal{N}_2(\cdot) \coloneqq i[\mathrm{SWAP}, \cdot] \tag{2.A.10}$$

$$\mathcal{M}_k(\cdot) \coloneqq M_k(\cdot)M_k^\dagger - \frac{1}{2}\left\{M_k^\dagger M_k, \cdot\right\}, \tag{2.A.11}$$

with the Lindblad operator M_k defined as in (2.3.9). Finally, Step 4 of the algorithm is to trace out the program register(s), and we repeat all the above-mentioned steps n times.

We represent each iteration of the above algorithm, i.e., Steps 1 to 4, as a quantum channel $\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML},\tau}$, where $\tau \coloneqq t/n$. This channel is defined as follows:

$$\begin{aligned}\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML},\tau} \coloneqq & \sum_{j:c_j>0} \frac{c_j}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_1 c\tau} \circ \mathcal{P}_{1,j} \\ & + \sum_{j:c_j<0} \frac{(-c_j)}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_2 c\tau} \circ \mathcal{P}_{1,j} \\ & + \sum_k \frac{\left\|L_{A_k}\right\|_2^2}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k}. \end{aligned} \tag{2.A.12}$$

Here, the subscript "ps" indicates the program state, and the channel $\mathrm{Tr}_{\mathrm{ps}}$ denotes the partial trace over the program state registers. The above channel form of a single step of the algorithm implies that the entire algorithm can be expressed as the composition of this channel n times:

$$\left(\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML},\tau}\right)^{\circ n}. \tag{2.A.13}$$

Note that we use a superscript "ideal" because we assume that the channels $e^{\mathcal{N}_1 c\tau}$, $e^{\mathcal{N}_2 c\tau}$, and $e^{\mathcal{M}_k c\tau}$ can be implemented exactly without any errors. However, this assumption is not practical.

As briefly mentioned before in Section 2.3.1, we aim to implement the Lindbladian channel $e^{\mathcal{M}_k c\tau}$ using an LCU-based algorithm introduced in [CW17]. Let us represent this algorithm as a quantum channel $\mathcal{R}_{kc\tau}$. Additionally, we represent this version of the WML algorithm that employs algorithm $\mathcal{R}_{kc\tau}$ as a subroutine for implementing $e^{\mathcal{M}_k c\tau}$ as

$$\left(\mathcal{A}^{(\mathrm{LCU})}_{\mathrm{WML},\tau}\right)^{\circ n}, \tag{2.A.14}$$

where

$$\begin{aligned}\mathcal{A}^{(\mathrm{LCU})}_{\mathrm{WML},\tau} := &\sum_{j:c_j>0} \frac{c_j}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_1 c\tau} \circ \mathcal{P}_{1,j} \\ &+ \sum_{j:c_j<0} \frac{(-c_j)}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_2 c\tau} \circ \mathcal{P}_{1,j} \\ &+ \sum_k \frac{\left\|L_{A_k}\right\|_2^2}{c} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{R}_{kc\tau} \circ \mathcal{P}_{2,k}\end{aligned} \tag{2.A.15}$$

In what follows, we prove Theorem 8 by breaking the analysis into two parts. To make sense of these two parts, consider the following:

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \left(\mathcal{A}^{(\mathrm{LCU})}_{\mathrm{WML}}\right)^{\circ n}\right\|_\diamond = \frac{1}{2}\left\|e^{\mathcal{L}t} - \left(\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML}}\right)^{\circ n} + \left(\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML}}\right)^{\circ n} - \left(\mathcal{A}^{(\mathrm{LCU})}_{\mathrm{WML}}\right)^{\circ n}\right\|_\diamond \tag{2.A.16}$$

$$\leq \frac{1}{2}\left\|e^{\mathcal{L}t} - \left(\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML}}\right)^{\circ n}\right\|_\diamond + \frac{1}{2}\left\|\left(\mathcal{A}^{(\mathrm{ideal})}_{\mathrm{WML}}\right)^{\circ n} - \left(\mathcal{A}^{(\mathrm{LCU})}_{\mathrm{WML}}\right)^{\circ n}\right\|_\diamond \tag{2.A.17}$$

$$= \frac{1}{2}\left\|\left(e^{\mathcal{L}\tau}\right)^{\circ n} - \left(\mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right)^{\circ n}\right\|_{\diamond} + \frac{1}{2}\left\|\left(\mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right)^{\circ n} - \left(\mathcal{A}_{\mathrm{WML}}^{(\mathrm{LCU})}\right)^{\circ n}\right\|_{\diamond} \tag{2.A.18}$$

$$\leq \frac{n}{2}\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right\|_{\diamond} + \frac{n}{2}\left\|\mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{LCU})}\right\|_{\diamond}. \tag{2.A.19}$$

To achieve a final error of at most ε, we can ensure that each of the two terms on the right-hand side of the inequality is bounded from above by $\frac{\varepsilon}{2}$:

$$\frac{n}{2}\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right\|_{\diamond} \leq \frac{\varepsilon}{2}, \tag{2.A.20}$$

$$\frac{n}{2}\left\|\mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{LCU})}\right\|_{\diamond} \leq \frac{\varepsilon}{2}. \tag{2.A.21}$$

To simplify the subsequent analysis, we divide it into two parts. In the first part, we analyze the initial inequality, which resolves the sample complexity of the algorithm, i.e., n. In the second part, we analyze the second inequality, which resolves the gate complexity of the algorithm.

2.A.1 Sample Complexity

Consider the following:

$$\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right\|_{\diamond} = \left\|e^{\mathcal{L}\tau} - \left(\begin{array}{c} \sum_{j:c_j>0} \frac{c_j}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_1 c\tau} \circ \mathcal{P}_{1,j} + \sum_{j:c_j<0} \frac{(-c_j)}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_2 c\tau} \circ \mathcal{P}_{1,j} \\ + \sum_k \frac{\|L_{A_k}\|_2^2}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k} \end{array}\right)\right\|_{\diamond} \tag{2.A.22}$$

Expanding the second term on the right-hand side of the above equation, we obtain:

$$\sum_{j:c_j>0} \frac{c_j}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_1 c\tau} \circ \mathcal{P}_{1,j}$$

$$= \sum_{j:c_j>0} \frac{c_j}{c} \mathrm{Tr}_{\mathrm{ps}} \circ \left(\mathcal{I} + c\tau\mathcal{N}_1 + \sum_{r=2}^{\infty} \frac{c^r\tau^r}{r!}\mathcal{N}_1^r\right) \circ \mathcal{P}_{1,j} \tag{2.A.23}$$

$$= \sum_{j:c_j>0} \frac{c_j}{c}\mathcal{I} + \sum_{j:c_j>0} c_j\tau \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1 \circ \mathcal{P}_{1,j} + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j} \tag{2.A.24}$$

$$= \sum_{j:c_j>0} \frac{c_j}{c} \mathcal{I} + \sum_{j:c_j>0} c_j \tau \mathcal{H}_j + \sum_{j:c_j>0} \sum_{r=2}^{\infty} \frac{c_j c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j}. \tag{2.A.25}$$

The second equality there follows from the fact that, for all $j \in [J]$, we have $\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1 \circ \mathcal{P}_{1,j} = \mathcal{H}_j$. For a visual proof of this identity, see the tensor-network diagrams in Figure 7 of [PW23a].

Similarly, we obtain the following expression for the third term:

$$\sum_{j:c_j<0} \frac{(-c_j)}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_2 c\tau} \circ \mathcal{P}_{1,j} = \sum_{j:c_j<0} \frac{(-c_j)}{c} \mathcal{I} + \sum_{j:c_j<0} (-c_j) \tau \mathcal{H}_j + \sum_{j:c_j<0} \sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2^r \circ \mathcal{P}_{1,j}, \tag{2.A.26}$$

and the following expression for the fourth term:

$$\sum_k \frac{\|L_{A_k}\|_2^2}{c} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k} = \sum_k \frac{\|L_{A_k}\|_2^2}{c} \mathcal{I} + \sum_k \|L_{A_k}\|_2^2 \tau \mathcal{L}_k + \sum_k \sum_{r=2}^{\infty} \frac{\|L_{A_k}\|_2^2 c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}. \tag{2.A.27}$$

To obtain the expression for the third term, we use the fact that for all $j \in [J]$, $\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2 \circ \mathcal{P}_{1,j} = \mathcal{H}_j$. A visual proof of this equality can be found in the tensor-network diagrams in Figure 7 of [PW23a]. Similarly, to obtain the expression for the fourth term, we use that for every $k \in [K]$, $\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k \circ \mathcal{P}_{2,k} = \widehat{\mathcal{L}}_k$, with the corresponding tensor-network diagrams provided in Figures 3, 5, and 6 of [PW23a].

Combining (2.A.25), (2.A.26), and (2.A.27) and rearranging, we get

$$\begin{aligned} &\left(\underbrace{\sum_{j:c_j>0} \frac{c_j}{c} + \sum_{j:c_j<0} \frac{(-c_j)}{c} + \sum_k \frac{\|L_{A_k}\|_2^2}{c}}_{=1} \right) \mathcal{I} + \tau \left(\underbrace{\sum_{j:c_j>0} c_j \mathcal{H}_j + \sum_{j:c_j<0} (-c_j) \mathcal{H}_j + \sum_k \|L_{A_k}\|_2^2 \widehat{\mathcal{L}}_k}_{=\mathcal{L}} \right) \\ &\quad + \sum_{j:c_j>0} \sum_{r=2}^{\infty} \frac{c_j c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j} + \sum_{j:c_j<0} \sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2^r \circ \mathcal{P}_{1,j} \\ &\quad + \sum_k \sum_{r=2}^{\infty} \frac{\|L_{A_k}\|_2^2 c^{r-1} \tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k} \end{aligned} \tag{2.A.28}$$

$$= \mathcal{I} + \tau\mathcal{L} + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j} + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2^r \circ \mathcal{P}_{1,j}$$
$$+ \sum_{k}\sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}. \tag{2.A.29}$$

By substituting the right-hand side of the above equation into (2.A.22) and expanding the term $e^{\mathcal{L}\tau}$ using its Taylor series, the first two terms of the Taylor series get canceled. As a result, we get the following:

$$\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})}\right\|_\diamond$$
$$= \left\| \sum_{r=2}^{\infty} \frac{\tau^r}{r!}\mathcal{L}^r - \left(\begin{array}{c} \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j} + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2^r \circ \mathcal{P}_{1,j} \\ + \sum_{k}\sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!} \mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k} \end{array} \right) \right\|_\diamond \tag{2.A.30}$$
$$\leq \sum_{r=2}^{\infty} \frac{\tau^r}{r!} \left\|\mathcal{L}^r\right\|_\diamond + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \left\|\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_1^r \circ \mathcal{P}_{1,j}\right\|_\diamond + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1}\tau^r}{r!} \left\|\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{N}_2^r \circ \mathcal{P}_{1,j}\right\|_\diamond$$
$$+ \sum_{k}\sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!} \left\|\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.A.31}$$
$$\leq \sum_{r=2}^{\infty} \frac{\tau^r}{r!} \left\|\mathcal{L}^r\right\|_\diamond + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \left\|\mathcal{N}_1^r\right\|_\diamond + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1}\tau^r}{r!} \left\|\mathcal{N}_2^r\right\|_\diamond$$
$$+ \sum_{k}\sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!} \left\|\mathcal{M}_k^r\right\|_\diamond \tag{2.A.32}$$
$$\leq \sum_{r=2}^{\infty} \frac{\tau^r}{r!} \left\|\mathcal{L}\right\|_\diamond^r + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!} \left\|\mathcal{N}_1\right\|_\diamond^r + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j) c^{r-1}\tau^r}{r!} \left\|\mathcal{N}_2\right\|_\diamond^r$$
$$+ \sum_{k}\sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!} \left\|\mathcal{M}_k\right\|_\diamond^r. \tag{2.A.33}$$

The first inequality follows from the triangle inequality. The second inequality follows from the following two facts: The diamond norm is submultiplicative under composition of maps, i.e., for maps $\mathcal{Q}$ and $\mathcal{R}$, it holds that $\|\mathcal{Q} \circ \mathcal{R}\|_\diamond \leq \|\mathcal{Q}\|_\diamond \|\mathcal{R}\|_\diamond$, and 2) the diamond norm for a quantum channel is equal to one, i.e., for a quantum channel $\mathcal{Q}$, it holds that

$\|Q\|_\diamond = 1$. Finally, the third inequality also follows from the submultiplicativity of the diamond norm under composition of maps.

Now, consider the following:

$$\|\mathcal{L}\|_\diamond = \left\| \sum_{j:c_j>0} c_j \mathcal{H}_j + \sum_{j:c_j<0} (-c_j)\mathcal{H}_j + \sum_k \left\|L_{A_k}\right\|_2^2 \widehat{\mathcal{L}}_k \right\|_\diamond \tag{2.A.34}$$

$$\leq \sum_{j:c_j>0} c_j \left\|\mathcal{H}_j\right\|_\diamond + \sum_{j:c_j<0} (-c_j) \left\|\mathcal{H}_j\right\|_\diamond + \sum_k \left\|L_{A_k}\right\|_2^2 \left\|\widehat{\mathcal{L}}_k\right\|_\diamond \tag{2.A.35}$$

$$\leq \sum_{j:c_j>0} c_j(2) + \sum_{j:c_j<0} (-c_j)(2) + \sum_k \left\|L_{A_k}\right\|_2^2 (2) \tag{2.A.36}$$

$$= 2\left(\sum_{j:c_j>0} c_j + \sum_{j:c_j<0} (-c_j) + \sum_k \left\|L_{A_k}\right\|_2^2 \right) \tag{2.A.37}$$

$$= 2c. \tag{2.A.38}$$

The first inequality follows from the triangle inequality. The second inequality holds due to the following:

$$\left\|\mathcal{H}_j\right\|_\diamond = \sup_\omega \left\|\mathcal{H}_j(\omega)\right\|_1 = \sup_\omega \left\|(-i)[\sigma_j, \omega]\right\|_1 \leq 2, \tag{2.A.39}$$

$$\left\|\widehat{\mathcal{L}}_k(\cdot)\right\|_\diamond = \sup_\omega \left\|\widehat{\mathcal{L}}_k(\omega)\right\|_1 = \sup_\omega \left\|\widehat{L}_k \omega \widehat{L}_k^\dagger - \frac{1}{2}\left\{\widehat{L}_k^\dagger \widehat{L}_k, \omega\right\}\right\|_1 \leq 2, \tag{2.A.40}$$

where recall that $\widehat{\mathcal{L}}_k = \mathcal{L}_k / \left\|L_{A_k}\right\|_2^2$.

Now, similar to bounding $\|\mathcal{L}\|_\diamond$, we bound $\|\mathcal{N}_1\|_\diamond$, $\|\mathcal{N}_2\|_\diamond$, and $\|\mathcal{M}_k\|_\diamond$ from above:

$$\|\mathcal{N}_1\|_\diamond = \sup_\omega \|\mathcal{N}_1(\omega)\|_1 = \sup_\omega \|(-i)[\text{SWAP}, \omega]\|_1 \tag{2.A.41}$$

$$= \sup_\omega \|(\text{SWAP}\,\omega - \omega\,\text{SWAP})\|_1 \leq 2\,\|\text{SWAP}\| \leq 2, \tag{2.A.42}$$

$$\|\mathcal{N}_2\|_\diamond = \sup_\omega \|\mathcal{N}_1(\omega)\|_1 = \sup_\omega \|(-i)[\text{-SWAP}, \omega]\|_1 \tag{2.A.43}$$

$$= \sup_\omega \|(\text{SWAP}\,\omega - \omega\,\text{SWAP})\|_1 \leq 2\,\|\text{SWAP}\| \leq 2, \tag{2.A.44}$$

$$\|\mathcal{M}_k\|_\diamond = \sup_\omega \|\mathcal{M}_k(\omega)\|_1 = \sup_\omega \left\| M_k \omega M_k^\dagger - \frac{1}{2}\{M_k^\dagger M_k, \omega\} \right\|_1 \tag{2.A.45}$$

$$\leq 2\|M_k\|^2 \leq 2Q. \tag{2.A.46}$$

Here, the last inequality follows directly from the definition of M_k in (2.3.9) and the fact that $d_{A_k} = Q$ for all $k \in [K]$, as noted at the beginning of this appendix.

Using the bounds (2.A.38), (2.A.42), (2.A.44), and (2.A.46) in (2.A.33), we get

$$\left\| e^{\mathcal{L}\tau} - \mathcal{A}_{\mathrm{WML}}^{(\mathrm{ideal})} \right\|_\diamond$$

$$\leq \sum_{r=2}^{\infty} \frac{\tau^r}{r!}(2c)^r + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}\tau^r}{r!}2^r + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j)c^{r-1}\tau^r}{r!}2^r + \sum_k \sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}\tau^r}{r!}(2Q)^r \tag{2.A.47}$$

$$\leq \sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_{j:c_j>0}\sum_{r=2}^{\infty} \frac{c_j c^{r-1}(2\tau)^r}{r!} + \sum_{j:c_j<0}\sum_{r=2}^{\infty} \frac{(-c_j)c^{r-1}(2\tau)^r}{r!} + \sum_k \sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}(2\tau)^r}{r!}(1 + Q^r - 1) \tag{2.A.48}$$

$$= \sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_{r=2}^{\infty} \frac{c^{r-1}(2\tau)^r}{r!}\left(\underbrace{\sum_{j:c_j>0} c_j + \sum_{j:c_j<0}(-c_j) + \sum_k \left\|L_{A_k}\right\|_2^2}_{=c} \right) + \sum_k \sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}(2\tau)^r}{r!}(Q^r - 1) \tag{2.A.49}$$

$$= \sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_k \sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}(2\tau)^r}{r!}(Q^r - 1) \tag{2.A.50}$$

$$= 2\sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_k \sum_{r=2}^{\infty} \frac{\left\|L_{A_k}\right\|_2^2 c^{r-1}(2\tau)^r}{r!}(Q^r - 1) \tag{2.A.51}$$

$$\leq 2\sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_{r=2}^{\infty} \frac{c \cdot c^{r-1}(2\tau)^r}{r!}Q^r \tag{2.A.52}$$

$$\leq 2\sum_{r=2}^{\infty} \frac{(2c\tau)^r}{r!} + \sum_{r=2}^{\infty} \frac{(2cQ\tau)^r}{r!}. \tag{2.A.53}$$

The third inequality follows from the fact that $\sum_k \left\|L_{A_k}\right\|_2^2 \leq c$ and $Q^r - 1 \leq Q^r$. Now, sub-

stituting $\tau = \frac{t}{n}$ in the above inequality and dividing by two on both sides for normalizing the diamond distance, we get

$$\frac{1}{2}\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\text{WML}}^{(\text{ideal})}\right\|_\diamond \leq \sum_{r=2}^{\infty} \frac{1}{r!}\left(\frac{2ct}{n}\right)^r + \frac{1}{2}\sum_{r=2}^{\infty}\frac{1}{r!}\left(\frac{2cQt}{n}\right)^r. \tag{2.A.54}$$

To bound the right-hand side of the inequality from above for $n \geq 2cQt$, we utilize the fact that for all $0 \leq x \leq 1$, $\sum_{r=2}^{\infty} \frac{x^r}{r!} \leq x^2$:

$$\frac{1}{2}\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\text{WML}}^{(\text{ideal})}\right\|_\diamond \leq \frac{(2ct)^2}{n^2} + \frac{1}{2}\frac{(2cQt)^2}{n^2} \tag{2.A.55}$$

$$\leq \frac{4(cQt)^2}{n^2}, \tag{2.A.56}$$

where the last inequality follows due to the fact that $Q \geq 2$.

Now, we use the above inequality to further bound the first term of (2.A.19) from above:

$$n \cdot \frac{1}{2}\left\|e^{\mathcal{L}\tau} - \mathcal{A}_{\text{WML}}^{(\text{ideal})}\right\|_\diamond \leq n \cdot \frac{4(cQt)^2}{n^2} = \frac{4(cQt)^2}{n}. \tag{2.A.57}$$

If we want the final error to be less than $\frac{\varepsilon}{2}$, then we need

$$n \geq \frac{8(cQt)^2}{\varepsilon} = O\left(\frac{c^2t^2}{\varepsilon}\right), \tag{2.A.58}$$

where we use that fact that $Q = 2^q = 2^{O(1)} = O(1)$. This resolves the sample complexity of the WML algorithm.

2.A.2 Gate Complexity

Substituting (2.A.15) and (2.A.12) into (2.A.19), the first two terms of (2.A.15) and (2.A.12) cancel out, leaving us with the following expression:

$$\frac{n}{2}\left\|\mathcal{A}_{\text{WML}}^{(\text{ideal})} - \mathcal{A}_{\text{WML}}^{(\text{LCU})}\right\|_\diamond = \frac{n}{2}\left\|\sum_k \frac{\|L_{A_k}\|_2^2}{c}\,\text{Tr}_{23} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k} - \sum_k \frac{\|L_{A_k}\|_2^2}{c}\,\text{Tr}_{23} \circ \mathcal{R}_{kc\tau} \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.A.59}$$

$$\leq \frac{n}{2} \sum_k \frac{\left\|L_{A_k}\right\|_2^2}{c} \left\|\mathrm{Tr}_{23} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k} - \mathrm{Tr}_{23} \circ \mathcal{R}_{kc\tau} \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.A.60}$$

$$\leq \frac{n}{2} \max_k \left\|\mathrm{Tr}_{23} \circ e^{\mathcal{M}_k c\tau} \circ \mathcal{P}_{2,k} - \mathrm{Tr}_{23} \circ \mathcal{R}_{kc\tau} \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.A.61}$$

$$\leq \frac{n}{2} \max_k \left\|\mathrm{Tr}_{23} \circ \left(e^{\mathcal{M}_k c\tau} - \mathcal{R}_{kc\tau}\right) \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.A.62}$$

$$\leq \frac{n}{2} \max_k \left\|e^{\mathcal{M}_k c\tau} - \mathcal{R}_{kc\tau}\right\|_\diamond, \tag{2.A.63}$$

where the first inequality follows from the triangle inequality, the second inequality follows due to the following fact:

$$\sum_k \frac{\left\|L_{A_k}\right\|_2^2}{c} \leq 1, \tag{2.A.64}$$

and the last inequality follows from the following two facts: The diamond norm is submultiplicative under composition of maps, i.e., for all maps $\mathcal{Q}$ and $\mathcal{R}$, it holds that $\|\mathcal{Q} \circ \mathcal{R}\|_\diamond \leq \|\mathcal{Q}\|_\diamond \|\mathcal{R}\|_\diamond$, and 2) the diamond norm for a quantum channel is equal to one, i.e., for all quantum channels $\mathcal{Q}$, it holds that $\|\mathcal{Q}\|_\diamond = 1$. Now, if we want the final error in (2.A.63) to be at most $\frac{\varepsilon}{2}$, then it suffices to have the following:

$$\max_k \frac{1}{2} \left\|e^{\mathcal{M}_k c\tau} - \mathcal{R}_{kc\tau}\right\|_\diamond \leq \frac{\varepsilon}{2n}. \tag{2.A.65}$$

Recall that $\mathcal{R}_{kc\tau}$ is an LCU-based quantum algorithm proposed in [CW17] for simulating Lindbladian channels. In our case, the channel of interest is $e^{\mathcal{M}_k c\tau}$. The algorithm $\mathcal{R}_{kc\tau}$ assumes an input model where the Lindblad operators are expressed as linear combinations of Pauli strings. Therefore, before applying the algorithm, we need to first express the Lindblad operators of the Lindbladian $\mathcal{M}_k$ into this required form, which we have in Appendix 2.B.

Observe that there are 16 terms in (2.B.8). This implies that there are 16^q or 2^{4q} terms in the linear-combination expression for M_k. This resolves the number of terms in the linear combination expression of M_k.

Additionally, note that a coefficient $\alpha_{i,k}$ in the linear-combination expression for M_k is either $+1/2^{q/2}$ or $-1/2^{q/2}$, which is clear to see from (2.B.5) and (2.B.8). Using this fact, we resolve the quantity $\|\mathcal{M}_k\|_{\text{Pauli}}$ in the following way:

$$\|\mathcal{M}_k\|_{\text{Pauli}} := \left(\sum_{i=0}^{2^{4q}-1} \alpha_{i,k}\right)^2 \leq \left(\sum_{i=0}^{2^{4q}-1} \frac{1}{2^{q/2}}\right)^2 \tag{2.A.66}$$

$$= \left(\frac{1}{2^{q/2}} 2^{4q}\right)^2 = 2^{7q}. \tag{2.A.67}$$

Using the development above together with Theorem 1 of [CW17], the gate complexity G of the subroutine $\mathcal{R}_{kc\tau}$, which implements the channel $e^{\mathcal{M}_k c\tau}$ within error ε/n in diamond distance, is

$$G = O\left(2^{15q} c\tau \frac{(\log(2^{15q} nc\tau/\varepsilon) + q)\, \log(nc\tau/\varepsilon)}{\log\log(nc\tau/\varepsilon)}\right). \tag{2.A.68}$$

Since $q = O(1)$ and $c\tau \leq 1$, this expression simplifies to

$$G = O\left(\frac{\log^2(n/\varepsilon)}{\log\log(n/\varepsilon)}\right) = O\left(\frac{\log^2(ct/\varepsilon)}{\log\log(ct/\varepsilon)}\right). \tag{2.A.69}$$

On the other hand, from [KLL+17] it follows that the unitary channel $e^{\mathcal{N}_j c\tau}$ can be implemented with $O(\log d)$ one- and two-qubit gates. Combining these two bounds, the total gate complexity of the full algorithm is

$$n \cdot \max\left\{\log d,\ \frac{\log^2(ct/\varepsilon)}{\log\log(ct/\varepsilon)}\right\} = O\left(\frac{c^2t^2}{\varepsilon} \cdot \max\left\{\log d,\ \frac{\log^2(ct/\varepsilon)}{\log\log(ct/\varepsilon)}\right\}\right). \tag{2.A.70}$$

This establishes the gate complexity of the sampling-based WML algorithm as stated in Theorem 8, thereby completing the proof.

2.B Decomposition of the Lindblad Operator M_k

In this appendix, we present a decomposition of the Lindblad operator M_k, defined in (2.3.9), as a linear combination of unitaries. The construction holds for every $k \in [K]$,

and to simplify notation we suppress the index and write M, A, P, Q; the dependence on k is implicit throughout.

We now introduce the notation used below. Let A denote the register of the subsystem consisting of q qubits and $\bar{A}$ denote the complement of A, that is, the register consisting of the rest of the $(\log d - q)$ qubits. Let P and Q jointly denote the program registers, each consisting of q qubits. As stated previously in (2.3.8), the Lindbladian $\mathcal{M}$ acts jointly on the system and program registers and consists of a single Lindblad operator M, given by

$$M \;=\; \frac{1}{2^{q/2}}(I_A \otimes |\Gamma\rangle\langle\Gamma|_{PQ})(\mathrm{SWAP}_{AP} \otimes I_Q)\,. \tag{2.B.1}$$

Note that here the action of the identity on $\bar{A}$ is implicit. Now observe that the multi-qubit operators I_A, $|\Gamma\rangle\langle\Gamma|_{PQ}$, and SWAP_{AP} decompose as tensor products over single- or two-qubit factors:

$$I_A = I_{A_1} \otimes I_{A_2} \otimes \cdots \otimes I_{A_q}, \tag{2.B.2}$$

$$|\Gamma\rangle\langle\Gamma|_{PQ} = |\Gamma\rangle\langle\Gamma|_{P_1Q_1} \otimes |\Gamma\rangle\langle\Gamma|_{P_2Q_2} \otimes \cdots \otimes |\Gamma\rangle\langle\Gamma|_{P_qQ_q}, \tag{2.B.3}$$

$$\mathrm{SWAP}_{AP} = \mathrm{SWAP}_{A_1P_1} \otimes \mathrm{SWAP}_{A_2P_2} \otimes \cdots \otimes \mathrm{SWAP}_{A_qP_q}, \tag{2.B.4}$$

where $\{A_i\}_{i\in[q]}$, $\{P_i\}_{i\in[q]}$, and $\{Q_i\}_{i\in[q]}$ are single-qubit registers and $A := A_1 \otimes \cdots \otimes A_q$, $P := P_1 \otimes \cdots \otimes P_q$, and $Q := Q_1 \otimes \cdots \otimes Q_q$. Using these equalities, we decompose M as

$$\begin{aligned} M = \underbrace{\frac{1}{2^{1/2}}(I_{A_1} \otimes |\Gamma\rangle\langle\Gamma|_{P_1Q_1})(\mathrm{SWAP}_{A_1P_1} \otimes I_{Q_1})}_{=:M_1} \otimes \underbrace{\frac{1}{2^{1/2}}(I_{A_2} \otimes |\Gamma\rangle\langle\Gamma|_{P_2Q_2})(\mathrm{SWAP}_{A_2P_2} \otimes I_{Q_2})}_{=:M_2} \otimes \cdots \\ \otimes \underbrace{\frac{1}{2^{1/2}}\Big(I_{A_q} \otimes |\Gamma\rangle\langle\Gamma|_{P_qQ_q}\Big)\Big(\mathrm{SWAP}_{A_qP_q} \otimes I_{Q_q}\Big)}_{=:M_q} \;=\; M_1 \otimes M_2 \otimes \cdots \otimes M_q. \end{aligned} \tag{2.B.5}$$

It is straightforward to see that a linear-combination-of-unitaries (LCU) expression for M follows once we obtain such an expression for each M_i. For all $i \in [q]$, the two-qubit

operators $\mathrm{SWAP}_{A_iP_i}$ and $|\Gamma\rangle\langle\Gamma|_{P_iQ_i}$ admit Pauli expansions:

$$\mathrm{SWAP}_{A_iP_i} = \tfrac{1}{2}(I_{A_i}\otimes I_{P_i} + X_{A_i}\otimes X_{P_i} + Y_{A_i}\otimes Y_{P_i} + Z_{A_i}\otimes Z_{P_i})\,, \tag{2.B.6}$$

$$|\Gamma\rangle\langle\Gamma|_{P_iQ_i} = \tfrac{1}{2}(I_{P_i}\otimes I_{Q_i} + X_{P_i}\otimes X_{Q_i} - Y_{P_i}\otimes Y_{Q_i} + Z_{P_i}\otimes Z_{Q_i})\,. \tag{2.B.7}$$

Suppressing system labels for readability, each M_i can then be written as

$$\begin{aligned} M_i = \frac{1}{4\cdot 2^{1/2}}\Big(&I\otimes I\otimes I + X\otimes X\otimes I + Y\otimes Y\otimes I + Z\otimes Z\otimes I \\ &+ I\otimes X\otimes X + X\otimes I\otimes X + Y\otimes iZ\otimes X - Z\otimes iY\otimes X \\ &- I\otimes Y\otimes Y + X\otimes iZ\otimes Y - Y\otimes I\otimes Y - Z\otimes iX\otimes Y \\ &+ I\otimes Z\otimes Z + X\otimes iY\otimes Z - Y\otimes iX\otimes Z + Z\otimes I\otimes Z\Big). \end{aligned} \tag{2.B.8}$$

There are 16 terms in the above linear combination for each M_i, and hence $M = M_1\otimes\cdots\otimes M_q$ expands into $16^q = 2^{4q}$ terms.

2.C Error Analysis for Trotter-like Approach

In this section, we analyze the error of the Trotter-like WML approach introduced in Section 2.3.1, thereby establishing Theorem 9. We begin by introducing the definitions and notations that will be used throughout the analysis. We present the analysis in a self-contained manner and thus can be read independently of the error analysis of the sampling-based WML algorithm presented in Appendix 2.A.

The Lindbladian $\mathcal{L}$, as defined in (2.3.1), can be expressed as

$$\mathcal{L} = \sum_{j=1}^{J}\mathcal{H}_j + \sum_{k=1}^{K}\mathcal{L}_k, \tag{2.C.1}$$

where

$$\mathcal{H}_j(\cdot) \coloneqq -i\,[c_j\sigma_j, (\cdot)], \qquad \mathcal{L}_k(\cdot) \coloneqq L_k(\cdot)L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, (\cdot)\}. \tag{2.C.2}$$

Each Lindblad operator L_k acts non-trivially on at most q qubits, with the identity acting on the remainder of the system. From the definition of L_k in (2.3.3), it follows that $d_{A_k} = 2^q =: Q$. We keep q as a parameter in the analysis to maintain generality. At the end, we substitute $q = O(1)$, since in our setting the Lindblad operators act on only a constant number of qubits. Our goal is to evaluate how accurately the Trotter-like WML algorithm approximates the target quantum channel $e^{\mathcal{L}t}$.

Consider the following appending quantum channels:

$$\mathcal{P}_{1,j}(\rho) \coloneqq \rho \otimes \sigma_j, \qquad \mathcal{P}_{2,k}(\rho) \coloneqq \rho \otimes \psi_k. \tag{2.C.3}$$

Note that in each iteration of the Trotter-like WML algorithm, we either apply $e^{\mathcal{N}_j\Delta'_j}$ or $e^{\mathcal{M}_k\Delta_k}$ channel to the joint system and program registers, where

$$\mathcal{N}_j(\cdot) \coloneqq -i\left[\operatorname{sgn}(c_j)\,\mathrm{SWAP}, (\cdot)\right] \tag{2.C.4}$$

$$\mathcal{M}_k(\cdot) \coloneqq M_k(\cdot)M_k^\dagger - \frac{1}{2}\left\{M_k^\dagger M_k, \cdot\right\}, \tag{2.C.5}$$

where the Lindblad operator M_k is defined in (2.3.9).

With the above notions in place, we can now express the Trotter-like WML algorithm in its quantum channel form, denoted by $\mathcal{T}_{\mathrm{WML}}$:

$$\begin{aligned}\mathcal{T}_{\mathrm{WML}} \coloneqq \Bigg[\left(\prod_{k=1}^{K} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k\Delta_k} \circ \mathcal{P}_{2,k}\right) \circ \left(\prod_{j=1}^{J} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_j\Delta'_j} \circ \mathcal{P}_{1,j}\right) \\ \circ \left(\prod_{j=J}^{1} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{N}_j\Delta'_j} \circ \mathcal{P}_{1,j}\right) \circ \left(\prod_{k=K}^{1} \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k\Delta_k} \circ \mathcal{P}_{2,k}\right)\Bigg]^{\circ n}.\end{aligned} \tag{2.C.6}$$

That being said, we now proceed to the error analysis. We begin by considering

$$\frac{1}{2}\left\|e^{\mathcal{L}t} - \mathcal{T}_{\mathrm{WML}}\right\|_\diamond$$

$$
\begin{aligned}
&= \frac{1}{2}\left\| e^{\mathcal{L}t} - \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} \right. \\
&\qquad \left. + \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} - \mathcal{T}_{\mathrm{WML}} \right\|_\diamond \\
&\leq \frac{1}{2}\left\| e^{\mathcal{L}t} - \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} \right\|_\diamond \\
&\qquad + \frac{1}{2}\left\| \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} - \mathcal{T}_{\mathrm{WML}} \right\|_\diamond, \qquad (2.C.7)
\end{aligned}
$$

where the inequality follows directly from the triangle inequality.

To bound the first term on the right-hand side of (2.C.7), we invoke Proposition 2 of [CL17], specifically Eq. (36) in its proof. This yields

$$
\begin{aligned}
&\frac{1}{2}\left\| e^{\mathcal{L}t} - \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} \right\|_\diamond \\
&\qquad \leq \frac{1}{2} \cdot \frac{16(J+K)^3 L^3 t^3}{n^2}, \qquad (2.C.8)
\end{aligned}
$$

where

$$
L := \max_{j,k} \left\{ \left\| \mathcal{H}_j \right\|_\diamond, \left\| \mathcal{L}_k \right\|_\diamond \right\}. \qquad (2.C.9)
$$

If we instead want to express this bound in terms of $\|\mathcal{L}\|_{\max}$ (see definition in (2.3.15)), we can use the inequality $L \leq 2\|\mathcal{L}\|_{\max}$. Substituting this inequality gives

$$
\begin{aligned}
&\frac{1}{2}\left\| e^{\mathcal{L}t} - \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} \right\|_\diamond \\
&\qquad \leq \frac{64(J+K)^3 \|\mathcal{L}\|_{\max}^3 t^3}{n^2}. \qquad (2.C.10)
\end{aligned}
$$

We now turn to the second term on the right-hand side of (2.C.7). By substituting the definition of $\mathcal{T}_{\mathrm{WML}}$ from (2.C.6) into this term, and then applying the subadditivity of the

diamond distance under composition of channels, we obtain

$$\frac{1}{2}\left\|\left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n}\right)\circ\left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n}\right)\circ\left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n}\right)\circ\left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n}\right)\right]^{\circ n} - \mathcal{T}_{\mathrm{WML}}\right\|_{\diamond}$$
$$\leq n\left(\sum_{j=1}^{J}\left\|e^{\mathcal{H}_j t/2n} - \mathrm{Tr}_{\mathrm{ps}}\circ e^{\mathcal{N}_j\Delta_j'}\circ\mathcal{P}_{1,j}\right\|_{\diamond} + \sum_{k=1}^{K}\left\|e^{\mathcal{L}_k t/2n} - \mathrm{Tr}_{\mathrm{ps}}\circ e^{\mathcal{M}_k\Delta_k}\circ\mathcal{P}_{2,k}\right\|_{\diamond}\right). \tag{2.C.11}$$

We now bound both the terms from above one by one:

$$\left\|e^{\mathcal{H}_j t/2n} - \mathrm{Tr}_{\mathrm{ps}}\circ e^{\mathcal{N}_j\Delta_j'}\circ\mathcal{P}_{1,j}\right\|_{\diamond}$$
$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{H}_j + \sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\mathcal{H}_j^r - \mathrm{Tr}_{\mathrm{ps}}\circ\left(\mathcal{I} + \Delta_j'\mathcal{N}_j + \sum_{r=2}^{\infty}\frac{(\Delta_j')^r}{r!}\mathcal{N}_j^r\right)\circ\mathcal{P}_{1,j}\right\|_{\diamond} \tag{2.C.12}$$
$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{H}_j + \sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\mathcal{H}_j^r - \left(\mathcal{I} + \Delta_j'\,\mathrm{Tr}_{\mathrm{ps}}\circ\mathcal{N}_j\circ\mathcal{P}_{1,j} + \sum_{r=2}^{\infty}\frac{(\Delta_j')^r}{r!}\,\mathrm{Tr}_{\mathrm{ps}}\circ\mathcal{N}_j^r\circ\mathcal{P}_{1,j}\right)\right\|_{\diamond} \tag{2.C.13}$$
$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{H}_j + \sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\mathcal{H}_j^r - \left(\mathcal{I} + \frac{|c_j|\,t}{2n}\cdot\frac{\mathcal{H}_j}{|c_j|} + \sum_{r=2}^{\infty}\frac{|c_j|^r\,t^r}{r!(2n)^r}\,\mathrm{Tr}_{\mathrm{ps}}\circ\mathcal{N}_j^r\circ\mathcal{P}_{1,j}\right)\right\|_{\diamond} \tag{2.C.14}$$
$$= \left\|\sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\mathcal{H}_j^r - \sum_{r=2}^{\infty}\frac{|c_j|^r\,t^r}{r!(2n)^r}\,\mathrm{Tr}_{\mathrm{ps}}\circ\mathcal{N}_j^r\circ\mathcal{P}_{1,j}\right\|_{\diamond} \tag{2.C.15}$$
$$\leq \sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\left(\left\|\mathcal{H}_j\right\|_{\diamond}^r + |c_j|^r\left\|\mathrm{Tr}_{\mathrm{ps}}\right\|_{\diamond}\left\|\mathcal{N}_j\right\|_{\diamond}^r\left\|\mathcal{P}_{1,j}\right\|_{\diamond}\right) \tag{2.C.16}$$
$$\leq \sum_{r=2}^{\infty}\frac{t^r}{r!(2n)^r}\left(2^r|c_j|^r + |c_j|^r\cdot 1\cdot 2^r\cdot 1\right) \tag{2.C.17}$$
$$= 2\sum_{r=2}^{\infty}\frac{t^r\,|c_j|^r}{r!n^r} \tag{2.C.18}$$
$$\leq 2\sum_{r=2}^{\infty}\frac{t^r\,\|\mathcal{L}\|_{\max}^r}{r!n^r} \tag{2.C.19}$$
$$\leq \frac{3\,\|\mathcal{L}\|_{\max}^2\,t^2}{2n^2}. \tag{2.C.20}$$

The third equality follows directly from the fact that for all $j \in [J]$, we have $\mathrm{Tr}_{\mathrm{ps}}\circ\mathcal{N}_j\circ\mathcal{P}_{1,j} = \frac{\mathcal{H}_j}{|c_j|}$ (see the tensor-network diagrams in Figure 7 of [PW23a] for a visual representation of

this equality) and also using the fact that $\Delta'_j = \frac{|c_j|t}{2n}$. Then the first inequality follows by first using the triangle inequality and then using the submultiplicativity property of the diamond norm under composition. The second inequality follows due to the following:

$$\left\|\mathcal{H}_j\right\|_\diamond = \sup_\omega \left\|\mathcal{H}_j(\omega)\right\|_1 = \sup_\omega \left\|(-i)c_j[\sigma_j, \omega]\right\|_1 \leq 2\left|c_j\right| \tag{2.C.21}$$

$$\|\mathcal{N}_1\|_\diamond = \sup_\omega \|\mathcal{N}_1(\omega)\|_1 = \sup_\omega \|(-i)[\mathrm{SWAP}, \omega]\|_1 \tag{2.C.22}$$

$$= \sup_\omega \|(\mathrm{SWAP}\,\omega - \omega\,\mathrm{SWAP})\|_1 \leq 2\,\|\mathrm{SWAP}\| \leq 2, \tag{2.C.23}$$

where the optimization is over the set of density operators of appropriate dimension. Then the third inequality follows from the definition of $\|\mathcal{L}\|_{\max}$ (see (2.3.15)). The fourth inequality follows from the exponential tail bound, which is that for all $x \in [0, 1]$, we have $\sum_{r=2}^\infty \frac{x^r}{r!} \leq \frac{3}{4}x^2$. For our case, we can use this tail bound for $n \geq t\,\|\mathcal{L}\|_{\max}$, which as we will see later is automatically true for our case.

Next, consider that

$$\left\|e^{\mathcal{L}_k t/2n} - \mathrm{Tr}_{\mathrm{ps}} \circ e^{\mathcal{M}_k \Delta_k} \circ \mathcal{P}_{2,k}\right\|_\diamond$$

$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{L}_k + \sum_{r=2}^\infty \frac{t^r}{r!(2n)^r}\mathcal{L}_k^r - \mathrm{Tr}_{\mathrm{ps}} \circ \left(\mathcal{I} + \Delta_k \mathcal{M}_k + \sum_{r=2}^\infty \frac{\Delta_k^r}{r!}\mathcal{M}_k^r\right) \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.C.24}$$

$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{L}_k + \sum_{r=2}^\infty \frac{t^r}{r!(2n)^r}\mathcal{L}_k^r - \left(\mathcal{I} + \Delta_k\,\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k \circ \mathcal{P}_{2,k} + \sum_{r=2}^\infty \frac{\Delta_k^r}{r!}\,\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}\right)\right\|_\diamond \tag{2.C.25}$$

$$= \left\|\mathcal{I} + \frac{t}{2n}\mathcal{L}_k + \sum_{r=2}^\infty \frac{t^r}{r!(2n)^r}\mathcal{L}_k^r - \left(\mathcal{I} + \frac{\left\|L_{A_k}\right\|_2^2 t}{2n} \cdot \frac{\mathcal{L}_k}{\left\|L_{A_k}\right\|_2^2} + \sum_{r=2}^\infty \frac{\left\|L_{A_k}\right\|_2^{2r} t^r}{r!(2n)^r}\,\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}\right)\right\|_\diamond \tag{2.C.26}$$

$$= \left\|\sum_{r=2}^\infty \frac{t^r}{r!(2n)^r}\mathcal{L}_k^r - \sum_{r=2}^\infty \frac{\left\|L_{A_k}\right\|_2^{2r} t^r}{r!(2n)^r}\,\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k^r \circ \mathcal{P}_{2,k}\right\|_\diamond \tag{2.C.27}$$

$$\leq \sum_{r=2}^\infty \frac{t^r}{r!(2n)^r}\left(\|\mathcal{L}_k\|_\diamond^r + \left\|L_{A_k}\right\|_2^{2r} \left\|\mathrm{Tr}_{\mathrm{ps}}\right\|_\diamond \|\mathcal{M}_k\|_\diamond^r \left\|\mathcal{P}_{2,k}\right\|_\diamond\right) \tag{2.C.28}$$

$$\leq \sum_{r=2}^{\infty} \frac{t^r}{r!(2n)^r} \left(2^r \left\| L_{A_k} \right\|_2^{2r} + \left\| L_{A_k} \right\|_2^{2r} \cdot 1 \cdot 2^{(q+1)r} \cdot 1 \right) \tag{2.C.29}$$

$$\leq 2 \sum_{r=2}^{\infty} \frac{t^r \left\| L_{A_k} \right\|_2^{2r} 2^{qr}}{r! n^r} \tag{2.C.30}$$

$$\leq 2 \sum_{r=2}^{\infty} \frac{t^r \left\| \mathcal{L} \right\|_{\max}^r 2^{qr}}{r! n^r} \tag{2.C.31}$$

$$\leq \frac{3}{2} \cdot \frac{2^{2q} \left\| \mathcal{L} \right\|_{\max}^2 t^2}{n^2}. \tag{2.C.32}$$

The third equality follows directly from the fact that for all $k \in [K]$, we have $\mathrm{Tr}_{\mathrm{ps}} \circ \mathcal{M}_k \circ \mathcal{P}_{2,k} = \mathcal{L}_k / \|L_{A_k}\|_2^2$ (see the tensor-network diagrams in Figures 3, 5, and 6 of [PW23a] for a visual representation of this equality) and also using the fact that $\Delta_k = \|L_{A_k}\|_2^2 t / 2n$. The second inequality follows due to the following:

$$\left\| \mathcal{L}_k \right\|_\diamond = \sup_\omega \left\| \mathcal{L}_k(\omega) \right\|_1 = \sup_\omega \left\| L_k \omega L_k^\dagger - \frac{1}{2} \left\{ L_k^\dagger L_k, \omega \right\} \right\|_1 \leq 2 \left\| L_k \right\|^2 \leq 2 \left\| L_{A_k} \right\|_2^2 \tag{2.C.33}$$

$$\left\| \mathcal{M}_k \right\|_\diamond = \sup_\omega \left\| \mathcal{M}_k(\omega) \right\|_1 = \sup_\omega \left\| M \omega M^\dagger - \frac{1}{2} \{ M^\dagger M, \omega \} \right\|_1 \tag{2.C.34}$$

$$\leq 2 \left\| M \right\|^2 \leq 2^{q+1}. \tag{2.C.35}$$

Then the third inequality follows from the definition of $\|\mathcal{L}\|_{\max}$ (see (2.3.15)). The fourth inequality follows from the exponential tail bound, which is that for all $x \in [0, 1]$, we have $\sum_{r=2}^{\infty} \frac{x^r}{r!} \leq \frac{3}{4} x^2$. For our case, we can use this tail bound for $n \geq 2^q t \left\| \mathcal{L} \right\|_{\max}$, which as we will see later is automatically true for our case.

Now using the bounds in (2.C.20) and (2.C.32), in (2.C.11), we obtain

$$\frac{1}{2} \left\| \left[\left(\prod_{k=1}^{K} e^{\mathcal{L}_k t/2n} \right) \circ \left(\prod_{j=1}^{J} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{j=J}^{1} e^{\mathcal{H}_j t/2n} \right) \circ \left(\prod_{k=K}^{1} e^{\mathcal{L}_k t/2n} \right) \right]^{\circ n} - \mathcal{T}_{\mathrm{WML}} \right\|_\diamond$$

$$\leq n \left(\sum_{j=1}^{J} \frac{3 \left\| \mathcal{L} \right\|_{\max}^2 t^2}{2n^2} + \sum_{k=1}^{K} \frac{3}{2} \cdot \frac{2^{2q} \left\| \mathcal{L} \right\|_{\max}^2 t^2}{n^2} \right) \tag{2.C.36}$$

$$\leq \frac{3(J+K) 2^{2q-1} \left\| \mathcal{L} \right\|_{\max}^2 t^2}{n}. \tag{2.C.37}$$

This concludes the bound on the second term of (2.C.7). With this along with the bound on the first term, as given in (2.C.10), we can finally bound (2.C.7) from above:

$$\frac{1}{2}\left\|e^{\mathcal{L}t}-\mathcal{T}_{\mathrm{WML}}\right\|_{\diamond}\leq\frac{64(J+K)^{3}\left\|\mathcal{L}\right\|_{\max}^{3}t^{3}}{n^{2}}+\frac{3(J+K)2^{2q-1}\left\|\mathcal{L}\right\|_{\max}^{2}t^{2}}{n}. \tag{2.C.38}$$

If the desired final error is ε, then if we take

$$n\geq 8\sqrt{2}\cdot\frac{2^{2q}(J+K)^{3/2}\left\|\mathcal{L}\right\|_{\max}^{2}t^{2}}{\varepsilon}, \tag{2.C.39}$$

we attain the desired error. Asymptotically, since $q=O(1)$, we have

$$n=O\left(\frac{(J+K)^{3/2}\left\|\mathcal{L}\right\|_{\max}^{2}t^{2}}{\varepsilon}\right) \tag{2.C.40}$$

This concludes the proof of Theorem 9.

2.D Lemma 19 Statement and Proof

For the purposes of our development in this appendix, we suppose that the system of interest is labeled by 0, and the program systems are labeled by $1,\ldots,2D$, with system 1 entangled with D, system 2 entangled with $D+1$, ..., and system D entangled with $2D$. We also employ the shorthand $\Gamma_{m,n}\equiv|\Gamma\rangle\langle\Gamma|_{m,n}$ throughout to denote the maximally entangled vector for systems m and n.

Lemma 19 *For*

$$M\coloneqq\frac{1}{d^{D/2}}\left(I_{0}\otimes\bigotimes_{\ell=1}^{D}|\Gamma\rangle\langle\Gamma|_{\ell,D+\ell}\right)\left(\mathrm{CYCSWAP}_{0,1,\ldots,D}\otimes I_{D+1,\ldots,2D}\right), \tag{2.D.1}$$

the following equalities hold:

$$\mathrm{Tr}_{1,\ldots,2D}\Big[M^\dagger M\,(\rho_0 \otimes L_1\Gamma_{1,D+1}L_1' \otimes L_2\Gamma_{2,D+2}L_2' \otimes \cdots \otimes L_D\Gamma_{D,2D}L_D')\Big]$$
$$= (L_D'L_{D-1}'\cdots L_3'L_2'L_1'L_1L_2L_3\cdots L_{D-1}L_D\rho)_0 \ , \quad (2.D.2)$$

$$\mathrm{Tr}_{1,\ldots,2D}\Big[(\rho_0 \otimes L_1\Gamma_{1,D+1}L_1' \otimes L_2\Gamma_{2,D+2}L_2' \otimes \cdots \otimes L_D\Gamma_{D,2D}L_D')\,M^\dagger M\Big]$$
$$= (\rho L_D'L_{D-1}'\cdots L_3'L_2'L_1'L_1L_2L_3\cdots L_{D-1}L_D)_0 \ , \quad (2.D.3)$$

$$\mathrm{Tr}_{1,\ldots,2D}\Big[M\,(\rho_0 \otimes L_1\Gamma_{1,D+1}L_1' \otimes L_2\Gamma_{2,D+2}L_2' \otimes \cdots \otimes L_D\Gamma_{D,2D}L_D')\,M^\dagger\Big]$$
$$= (L_1L_2\cdots L_{D-1}L_D\rho L_D'L_{D-1}'\cdots L_2'L_1')_0 \, , \quad (2.D.4)$$

where ρ_0 is a density operator acting on system 0, the operators $L_1, \ldots, L_D$, $L_1', \ldots, L_D'$ are arbitrary square linear operators, and identity operators for systems $D+1, \ldots, 2D$ are implicit in each of the first lines above.

Proof. Let us begin with the following observation:

$$M = \frac{1}{d^{D/2}}\left(I_0 \otimes \bigotimes_{\ell=1}^{D} |\Gamma\rangle\langle\Gamma|_{\ell,D+\ell}\right)(\mathrm{CYCSWAP}_{0,1,\ldots,D} \otimes I_{D+1,\ldots,2D}) \quad (2.D.5)$$

$$= \frac{1}{d^{D/2}}\left(\begin{array}{c}\sum_k |k\rangle\langle k|_0 \otimes \sum_{\substack{i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}} |i_1\rangle\langle j_1|_1 \otimes |i_2\rangle\langle j_2|_2 \otimes \cdots \otimes |i_D\rangle\langle j_D|_D \otimes \\ |i_1\rangle\langle j_1|_{D+1} \otimes |i_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |i_D\rangle\langle j_D|_{2D}\end{array}\right)$$
$$\times (\mathrm{CYCSWAP}_{0,1,\ldots,D} \otimes I_{D+1,\ldots,2D}) \quad (2.D.6)$$

$$= \frac{1}{d^{D/2}}\left(\begin{array}{c}\sum_{k,i_1,j_1,\ \substack{i_2,j_2,\\ \ldots,i_D,j_D}} |k\rangle\langle j_D|_0 \otimes |i_1\rangle\langle k|_1 \otimes |i_2\rangle\langle j_1|_2 \otimes \cdots \otimes |i_{D-1}\rangle\langle j_{D-2}|_{D-1} \\ \otimes |i_D\rangle\langle j_{D-1}|_D \otimes |i_1\rangle\langle j_1|_{D+1} \otimes |i_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |i_D\rangle\langle j_D|_{2D}\end{array}\right). \quad (2.D.7)$$

Then we find that

$$M^\dagger M$$

$$= \frac{1}{d^{D/2}} \left(\begin{array}{c} \sum_{\substack{k',i'_1,j'_1,\\ i'_2,j'_2,\\ \ldots,i'_D,j'_D}} |j'_D\rangle\langle k'|_0 \otimes |k'\rangle\langle i'_1|_1 \otimes |j'_1\rangle\langle i'_2|_2 \otimes \cdots \otimes |j'_{D-2}\rangle\langle i'_{D-1}|_{D-1} \\ \otimes |j'_{D-1}\rangle\langle i'_D|_D \otimes |j'_1\rangle\langle i'_1|_{D+1} \otimes |j'_2\rangle\langle i'_2|_{D+2} \otimes \cdots \otimes |j'_D\rangle\langle i'_D|_{2D} \end{array} \right)$$

$$\times \frac{1}{d^{D/2}} \left(\begin{array}{c} \sum_{\substack{k,i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}} |k\rangle\langle j_D|_0 \otimes |i_1\rangle\langle k|_1 \otimes |i_2\rangle\langle j_1|_2 \otimes \cdots \otimes |i_{D-1}\rangle\langle j_{D-2}|_{D-1} \\ \otimes |i_D\rangle\langle j_{D-1}|_D \otimes |i_1\rangle\langle j_1|_{D+1} \otimes |i_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |i_D\rangle\langle j_D|_{2D} \end{array} \right) \tag{2.D.8}$$

$$= \frac{1}{d^D} \sum_{\substack{k',i'_1,j'_1,\\ i'_2,j'_2,\\ \ldots,i'_D,j'_D}} \sum_{\substack{k,i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}} |j'_D\rangle\langle k'|k\rangle\langle j_D|_0 \otimes |k'\rangle\langle i'_1|i_1\rangle\langle k|_1 \otimes |j'_1\rangle\langle i'_2|i_2\rangle\langle j_1|_2 \otimes \cdots$$

$$\otimes |j'_{D-2}\rangle\langle i'_{D-1}|i_{D-1}\rangle\langle j_{D-2}|_{D-1} \otimes |j'_{D-1}\rangle\langle i'_D|i_D\rangle\langle j_{D-1}|_D$$

$$\otimes |j'_1\rangle\langle i'_1|i_1\rangle\langle j_1|_{D+1} \otimes |j'_2\rangle\langle i'_2|i_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |j'_D\rangle\langle i'_D|i_D\rangle\langle j_D|_{2D} \tag{2.D.9}$$

$$= \sum_{\substack{j'_1,i'_2,j'_2,\\ \ldots,i'_D,j'_D}} \sum_{\substack{k,i_1,j_1,i_2,j_2,\\ \ldots,i_D,j_D}} |j'_D\rangle\langle j_D|_0 \otimes |k\rangle\langle k|_1 \otimes |j'_1\rangle\langle j_1|_2 \otimes \cdots \otimes |j'_{D-2}\rangle\langle j_{D-2}|_{D-1}$$

$$\otimes |j'_{D-1}\rangle\langle j_{D-1}|_D \otimes |j'_1\rangle\langle j_1|_{D+1} \otimes |j'_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |j'_D\rangle\langle j_D|_{2D} \tag{2.D.10}$$

$$= I_1 \otimes \Gamma_{0,2D} \otimes \Gamma_{2,D+1} \otimes \Gamma_{3,D+2} \otimes \cdots \otimes \Gamma_{D-1,2D-2} \otimes \Gamma_{D,2D-1}. \tag{2.D.11}$$

We first prove (2.D.2). To this end, consider that

$$M^\dagger M \left(\rho_0 \otimes L_1 \Gamma_{1,D+1} L'_1 \otimes L_2 \Gamma_{2,D+2} L'_2 \otimes \cdots \otimes L_D \Gamma_{D,2D} L'_D\right)$$

$$= \left[\begin{array}{c} (I_1 \otimes \Gamma_{0,2D} \otimes \Gamma_{2,D+1} \otimes \Gamma_{3,D+2} \otimes \cdots \otimes \Gamma_{D-1,2D-2} \otimes \Gamma_{D,2D-1}) \\ \left(\rho_0 \otimes L_1 \Gamma_{1,D+1} L'_1 \otimes L_2 \Gamma_{2,D+2} L'_2 \otimes \cdots \otimes L_D \Gamma_{D,2D} L'_D\right) \end{array} \right] \tag{2.D.12}$$

$$= \sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D,k_1,\ell_1,\\ k_2,\ell_2,\ldots,k_D,\ell_D}} \left[\begin{array}{c} \left(\begin{array}{c} I_1 \otimes |i_1\rangle\langle j_1|_0 \otimes |i_1\rangle\langle j_1|_{2D} \otimes |i_2\rangle\langle j_2|_2 \otimes |i_2\rangle\langle j_2|_{D+1} \\ \otimes |i_3\rangle\langle j_3|_3 \otimes |i_3\rangle\langle j_3|_{D+2} \otimes \cdots \otimes |i_{D-1}\rangle\langle j_{D-1}|_{D-1} \\ \otimes |i_{D-1}\rangle\langle j_{D-1}|_{2D-2} \otimes |i_D\rangle\langle j_D|_D \otimes |i_D\rangle\langle j_D|_{2D-1} \end{array} \right) \\ \left(\begin{array}{c} \rho_0 \otimes L_1 |k_1\rangle\langle \ell_1|_1 L'_1 \otimes |k_1\rangle\langle \ell_1|_{D+1} \otimes L_2 |k_2\rangle\langle \ell_2|_2 L'_2 \\ \otimes |k_2\rangle\langle \ell_2|_{D+2} \otimes \cdots \otimes L_D |k_D\rangle\langle \ell_D|_D L'_D \otimes |k_D\rangle\langle \ell_D|_{2D} \end{array} \right) \end{array} \right] \tag{2.D.13}$$

$$= \sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D,k_1,\ell_1,\\ k_2,\ell_2,k_3,\ell_3,\ldots,\\ k_{D-2},\ell_{D-2},k_{D-1},\\ \ell_{D-1},k_D,\ell_D}} \begin{pmatrix} L_1|k_1\rangle\langle\ell_1|_1 L_1' \otimes |i_1\rangle\langle j_1|_0 \rho_0 \otimes |i_1\rangle\langle j_1|_{2D} |k_D\rangle\langle\ell_D|_{2D} \\ \otimes |i_2\rangle\langle j_2|_2 L_2 |k_2\rangle\langle\ell_2|_2 L_2' \otimes |i_2\rangle\langle j_2|_{D+1} |k_1\rangle\langle\ell_1|_{D+1} \\ \otimes |i_3\rangle\langle j_3|_3 L_3 |k_3\rangle\langle\ell_3|_3 L_3' \otimes |i_3\rangle\langle j_3|_{D+2} |k_2\rangle\langle\ell_2|_{D+2} \otimes \cdots \\ \otimes |i_{D-1}\rangle\langle j_{D-1}|_{D-1} L_{D-1} |k_{D-1}\rangle\langle\ell_{D-1}|_D L_{D-1}' \\ \otimes |i_{D-1}\rangle\langle j_{D-1}|_{2D-2} |k_{D-2}\rangle\langle\ell_{D-2}|_{2D-2} \\ \otimes |i_D\rangle\langle j_D|_D L_D |k_D\rangle\langle\ell_D|_D L_D' \\ \otimes |i_D\rangle\langle j_D|_{2D-1} |k_{D-1}\rangle\langle\ell_{D-1}|_{2D-1} \end{pmatrix}. \tag{2.D.14}$$

Taking the partial trace over systems $1, \ldots, 2D$ of the last line above gives

$$\sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D,k_1,\ell_1,\\ k_2,\ell_2,k_3,\ell_3,\ldots,\\ k_{D-2},\ell_{D-2},k_{D-1},\\ \ell_{D-1},k_D,\ell_D}} \begin{pmatrix} |i_1\rangle\langle j_1|_0 \rho_0 \, \langle\ell_1|_1 L_1' L_1 |k_1\rangle \, \langle j_1|k_D\rangle\langle\ell_D|i_1\rangle \\ \langle j_2|_2 L_2 |k_2\rangle\langle\ell_2| L_2' |i_2\rangle \, \langle j_2|k_1\rangle\langle\ell_1|i_2\rangle \\ \langle j_3|_3 L_3 |k_3\rangle\langle\ell_3|_3 L_3' |i_3\rangle \, \langle j_3|k_2\rangle\langle\ell_2|i_3\rangle \cdots \\ \langle j_{D-1}|_{D-1} L_{D-1} |k_{D-1}\rangle\langle\ell_{D-1}|_D L_{D-1}' |i_{D-1}\rangle \\ \langle j_{D-1}|k_{D-2}\rangle\langle\ell_{D-2}|i_{D-1}\rangle \\ \langle j_D|_D L_D |k_D\rangle\langle\ell_D|_D L_D' |i_D\rangle \, \langle j_D|k_{D-1}\rangle\langle\ell_{D-1}|i_D\rangle \end{pmatrix}$$

$$= \sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D}} \begin{pmatrix} |i_1\rangle\langle j_1|_0 \rho_0 \, \langle i_2|_1 L_1' L_1 |j_2\rangle \\ \langle j_2|_2 L_2 |j_3\rangle\langle i_3| L_2' |i_2\rangle \\ \langle j_3|_3 L_3 |j_4\rangle\langle i_4|_3 L_3' |i_3\rangle \, \cdots \\ \langle j_{D-1}|_{D-1} L_{D-1} |j_D\rangle\langle i_D|_D L_{D-1}' |i_{D-1}\rangle \\ \langle j_D|_D L_D |j_1\rangle\langle i_1|_D L_D' |i_D\rangle \end{pmatrix} \tag{2.D.15}$$

$$= \sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D}} \begin{pmatrix} |i_1\rangle\langle j_1|_0 \rho_0 \, \langle i_2|_1 L_1' L_1 |j_2\rangle \, \langle j_2|_2 L_2 |j_3\rangle\langle i_3| L_2' |i_2\rangle \\ \langle j_3|_3 L_3 |j_4\rangle\langle i_4|_3 L_3' |i_3\rangle \, \cdots \langle j_{D-1}|_{D-1} L_{D-1} |j_D\rangle \\ \langle i_D|_D L_{D-1}' |i_{D-1}\rangle \, \langle j_D|_D L_D |j_1\rangle\langle i_1|_D L_D' |i_D\rangle \end{pmatrix} \tag{2.D.16}$$

$$= \sum_{\substack{i_1,j_1,i_2,j_2,\\ i_3,j_3,i_{D-1},j_{D-1},\\ i_D,j_D}} \begin{pmatrix} |i_1\rangle\langle i_1|_D L'_D |i_D\rangle\langle i_D|_D L'_{D-1} |i_{D-1}\rangle \cdots \langle i_4|_3 L'_3 |i_3\rangle \\ \langle i_3|L'_2|i_2\rangle\langle i_2|_1 L'_1 L_1 |j_2\rangle\langle j_2|_2 L_2 |j_3\rangle \\ \langle j_3|_3 L_3 |j_4\rangle \cdots \langle j_{D-1}|_{D-1} L_{D-1} |j_D\rangle \\ \langle j_D|_D L_D |j_1\rangle \langle j_1|_0 \rho_0 \end{pmatrix} \tag{2.D.17}$$

$$= \left(L'_D L'_{D-1} \cdots L'_3 L'_2 L'_1 L_1 L_2 L_3 \cdots L_{D-1} L_D \rho\right)_0 \ . \tag{2.D.18}$$

This completes the proof of (2.D.2).

Using the Hermitian conjugate of (2.D.2) along with some substitutions then gives (2.D.3).

Now we finally prove (2.D.4). To this end, consider that

$$\begin{aligned}
&M\left(\rho_0 \otimes L_1 \Gamma_{1,D+1} L'_1 \otimes L_2 \Gamma_{2,D+2} L'_2 \otimes \cdots \otimes L_D \Gamma_{D,2D} L'_D\right) M^\dagger \\
&= \frac{1}{d^D} \begin{pmatrix} \sum_{\substack{k,i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}} |k\rangle\langle j_D|_0 \otimes |i_1\rangle\langle k|_1 \otimes |i_2\rangle\langle j_1|_2 \otimes \cdots \otimes |i_{D-1}\rangle\langle j_{D-2}|_{D-1} \\ \otimes |i_D\rangle\langle j_{D-1}|_D \otimes |i_1\rangle\langle j_1|_{D+1} \otimes |i_2\rangle\langle j_2|_{D+2} \otimes \cdots \otimes |i_D\rangle\langle j_D|_{2D} \end{pmatrix} \\
&\quad \times \left(\rho_0 \otimes L_1 \Gamma_{1,D+1} L'_1 \otimes L_2 \Gamma_{2,D+2} L'_2 \otimes \cdots \otimes L_D \Gamma_{D,2D} L'_D\right) \\
&\quad \times \begin{pmatrix} \sum_{\substack{k',i'_1,j'_1,\\ i'_2,j'_2,\\ \ldots,i'_D,j'_D}} |j'_D\rangle\langle k'|_0 \otimes |k'\rangle\langle i'_1|_1 \otimes |j'_1\rangle\langle i'_2|_2 \otimes \cdots \otimes |j'_{D-2}\rangle\langle i'_{D-1}|_{D-1} \\ \otimes |j'_{D-1}\rangle\langle i'_D|_D \otimes |j'_1\rangle\langle i'_1|_{D+1} \otimes |j'_2\rangle\langle i'_2|_{D+2} \otimes \cdots \otimes |j'_D\rangle\langle i'_D|_{2D} \end{pmatrix} \\
&= \frac{1}{d^D} \sum_{\substack{k,i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}} \sum_{\substack{k',i'_1,j'_1,\\ i'_2,j'_2,\\ \ldots,i'_D,j'_D}} |k\rangle\langle j_D|_0 \rho_0 |j'_D\rangle\langle k'|_0 \\
&\quad \otimes \left(|i_1\rangle\langle k|_1 \otimes |i_1\rangle\langle j_1|_{D+1}\right) L_1 \Gamma_{1,D+1} L'_1 \left(|k'\rangle\langle i'_1|_1 \otimes |j'_1\rangle\langle i'_1|_{D+1}\right) \\
&\quad \otimes \left(|i_2\rangle\langle j_1|_2 \otimes |i_2\rangle\langle j_2|_{D+2}\right) L_2 \Gamma_{2,D+2} L'_2 \left(|j'_1\rangle\langle i'_2|_2 \otimes |j'_2\rangle\langle i'_2|_{D+2}\right) \\
&\quad \otimes \cdots \otimes \\
&\quad \left(|i_{D-1}\rangle\langle j_{D-2}|_{D-1} \otimes |i_{D-1}\rangle\langle j_{D-1}|_{2D-1}\right) L_{D-1} \Gamma_{D-1,2D-1} \times
\end{aligned} \tag{2.D.19}$$

$$L'_{D-1}\left(|j'_{D-2}\rangle\langle i'_{D-1}|_{D-1}\otimes|j'_{D-1}\rangle\langle i'_{D-1}|_{2D-1}\right)$$
$$\otimes\left(|i_D\rangle\langle j_{D-1}|_D\otimes|i_D\rangle\langle j_D|_{2D}\right)L_D\Gamma_{D,2D}L'_D\left(|j'_{D-1}\rangle\langle i'_D|_D\otimes|j'_D\rangle\langle i'_D|_{2D}\right). \tag{2.D.20}$$

Now taking a partial trace over systems $1,\ldots,2D$ in the last line gives the following:

$$\begin{aligned}
&\frac{1}{d^D}\sum_{\substack{k,i_1,j_1,\\ i_2,j_2,\\ \ldots,i_D,j_D}}\sum_{\substack{k',i'_1,j'_1,\\ i'_2,j'_2,\\ \ldots,i'_D,j'_D}}|k\rangle\langle j_D|_0\rho_0|j'_D\rangle\langle k'|_0\\
&\quad\times\left(\langle k|_1\otimes\langle j_1|_{D+1}\right)L_1\Gamma_{1,D+1}L'_1\left(|k'\rangle_1\langle i'_1|i_1\rangle\otimes|j'_1\rangle_{D+1}\langle i'_1|i_1\rangle\right)\\
&\quad\times\left(\langle j_1|_2\otimes\langle j_2|_{D+2}\right)L_2\Gamma_{2,D+2}L'_2\left(|j'_1\rangle_2\langle i'_2|i_2\rangle\otimes|j'_2\rangle_{D+2}\langle i'_2|i_2\rangle\right)\cdots\\
&\quad\times\left(\langle j_{D-2}|_{D-1}\otimes\langle j_{D-1}|_{2D-1}\right)L_{D-1}\Gamma_{D-1,2D-1}\\
&\quad\times L'_{D-1}\left(|j'_{D-2}\rangle_{D-1}\langle i'_{D-1}|i_{D-1}\rangle\otimes|j'_{D-1}\rangle_{2D-1}\langle i'_{D-1}|i_{D-1}\rangle\right)\\
&\quad\times\left(\langle j_{D-1}|_D\otimes\langle j_D|_{2D}\right)L_D\Gamma_{D,2D}L'_D\left(|j'_{D-1}\rangle_D\langle i'_D|i_D\rangle\otimes|j'_D\rangle_{2D}\langle i'_D|i_D\rangle\right)\\
&=\sum_{\substack{k,j_1,j_2,\\ \ldots,j_D}}\sum_{\substack{k',j'_1,j'_2,\\ \ldots,j'_D}}|k\rangle\langle j_D|_0\rho_0|j'_D\rangle\langle k'|_0\\
&\quad\times\left(\langle k|_1\otimes\langle j_1|_{D+1}\right)L_1\Gamma_{1,D+1}L'_1\left(|k'\rangle_1\otimes|j'_1\rangle_{D+1}\right)\\
&\quad\times\left(\langle j_1|_2\otimes\langle j_2|_{D+2}\right)L_2\Gamma_{2,D+2}L'_2\left(|j'_1\rangle_2\otimes|j'_2\rangle_{D+2}\right)\cdots\\
&\quad\times\left(\langle j_{D-2}|_{D-1}\otimes\langle j_{D-1}|_{2D-1}\right)L_{D-1}\Gamma_{D-1,2D-1}L'_{D-1}\left(|j'_{D-2}\rangle_{D-1}\otimes|j'_{D-1}\rangle_{2D-1}\right)\\
&\quad\times\left(\langle j_{D-1}|_D\otimes\langle j_D|_{2D}\right)L_D\Gamma_{D,2D}L'_D\left(|j'_{D-1}\rangle_D\otimes|j'_D\rangle_{2D}\right) \qquad (2.D.21)\\
&=\sum_{\substack{k,j_1,j_2,\\ \ldots,j_D}}\sum_{\substack{k',j'_1,j'_2,\\ \ldots,j'_D}}|k\rangle\langle j_D|\rho_0|j'_D\rangle\langle k'|_0\\
&\quad\times\langle k|L_1|j_1\rangle\,\langle j'_1|L'_1|k'\rangle\\
&\quad\times\langle j_1|L_2|j_2\rangle\,\langle j'_2|L'_2|j'_1\rangle\,\cdots\\
&\quad\times\langle j_{D-2}|L_{D-1}|j_{D-1}\rangle\,\langle j'_{D-1}|L'_{D-1}|j'_{D-2}\rangle\\
&\quad\times\langle j_{D-1}|L_D|j_D\rangle\,\langle j'_D|L'_D|j'_{D-1}\rangle \qquad (2.D.22)
\end{aligned}$$

$$= \sum_{\substack{k,j_1,j_2,\\ \ldots,j_D}} \sum_{\substack{k',j_1',j_2',\\ \ldots,j_D'}} |k\rangle\langle k|L_1|j_1\rangle\langle j_1|L_2|j_2\rangle \cdots$$

$$\times \langle j_{D-2}|L_{D-1}|j_{D-1}\rangle\langle j_{D-1}|L_D|j_D\rangle\langle j_D|\rho_0|j_D'\rangle$$

$$\times \langle j_D'|L_D'|j_{D-1}'\rangle\langle j_{D-1}'|L_{D-1}'|j_{D-2}'\rangle \cdots \langle j_2'|L_2'|j_1'\rangle\langle j_1'|L_1'|k'\rangle\langle k'| \tag{2.D.23}$$

$$= (L_1 L_2 \cdots L_{D-1} L_D \rho L_D' L_{D-1}' \cdots L_2' L_1')_0 \ . \tag{2.D.24}$$

This completes the proof of (2.D.4). ■

2.E Proof of Theorem 17

To begin with, we have

$$\left\| e^{\tilde{\mathcal{L}}t} - e^{\mathcal{L}t} \right\|_\diamond$$

$$= \lim_{r\to\infty} \left\| (e^{\tilde{\mathcal{L}}t/r})^r - (e^{\mathcal{L}t/r})^r \right\|_\diamond \tag{2.E.1}$$

$$\leq \lim_{r\to\infty} r \left\| e^{\tilde{\mathcal{L}}t/r} - e^{\mathcal{L}t/r} \right\|_\diamond \tag{2.E.2}$$

$$= \lim_{r\to\infty} r \sup_{\omega\in\mathcal{D}(\mathcal{H}_{RS})} \left\| e^{\tilde{\mathcal{L}}t/r}(\omega) - e^{\mathcal{L}t/r}(\omega) \right\|_1 \tag{2.E.3}$$

$$\leq \lim_{r\to\infty} r \left(\sup_{\omega\in\mathcal{D}(\mathcal{H}_{RS})} \left\| \omega + \mathcal{L}(\omega)t/r - \omega - \tilde{\mathcal{L}}(\omega)t/r \right\|_1 + O(t^2/r^2) \right) \tag{2.E.4}$$

$$= \lim_{r\to\infty} r \left(\sup_{\omega\in\mathcal{D}(\mathcal{H}_{RS})} \left\| \mathcal{L}(\omega) - \tilde{\mathcal{L}}(\omega) \right\|_1 t/r + O(t^2/r^2) \right) \tag{2.E.5}$$

$$= \lim_{r\to\infty} \sup_{\omega\in\mathcal{D}(\mathcal{H}_{RS})} \left\| \mathcal{L}(\omega) - \tilde{\mathcal{L}}(\omega) \right\|_1 t + O(t^2/r) \tag{2.E.6}$$

$$= \sup_{\omega\in\mathcal{D}(\mathcal{H}_{RS})} \left\| \mathcal{L}(\omega) - \tilde{\mathcal{L}}(\omega) \right\|_1 t \tag{2.E.7}$$

$$= \left\| \mathcal{L} - \tilde{\mathcal{L}} \right\|_\diamond t. \tag{2.E.8}$$

The first inequality employs the subadditivity of diamond distance under channel composition (the proof of this latter statement employs the triangle inequality, in a way similar

to the proof of [NC10, Eq. (4.63)]). For clarity, the identity channel's action is implicit in the second equality and thereafter. The second inequality applies a Taylor expansion and the triangle inequality.

Now, for bounding $\left\|\mathcal{L} - \tilde{\mathcal{L}}\right\|_\diamond$, let $\omega \in \mathcal{D}(\mathcal{H}_{RS})$ and, again with implicit identities acting on R, consider that

$$
\begin{aligned}
&\left\|\mathcal{L}(\omega) - \tilde{\mathcal{L}}(\omega)\right\|_1 \\
&= \left\| L\omega L^\dagger - \frac{1}{2}\left\{L^\dagger L, \omega\right\} - \tilde{L}\omega\tilde{L}^\dagger + \frac{1}{2}\left\{\tilde{L}^\dagger\tilde{L}, \omega\right\}\right\|_1 && \text{(2.E.9)} \\
&= \left\| L\omega L^\dagger - \frac{1}{2}\left(L^\dagger L\omega + \omega L^\dagger L\right) - \tilde{L}\omega\tilde{L}^\dagger + \frac{1}{2}\left(\tilde{L}^\dagger\tilde{L}\omega + \omega\tilde{L}^\dagger\tilde{L}\right)\right\|_1 && \text{(2.E.10)} \\
&= \left\| L\omega L^\dagger - \tilde{L}\omega\tilde{L}^\dagger - \frac{1}{2}\left(L^\dagger L\omega - \tilde{L}^\dagger\tilde{L}\omega\right) - \frac{1}{2}\left(\omega L^\dagger L - \omega\tilde{L}^\dagger\tilde{L}\right)\right\|_1 && \text{(2.E.11)} \\
&\leq \left\| L\omega L^\dagger - \tilde{L}\omega\tilde{L}^\dagger\right\|_1 + \frac{1}{2}\left\| L^\dagger L\omega - \tilde{L}^\dagger\tilde{L}\omega\right\|_1 + \frac{1}{2}\left\|\omega L^\dagger L - \omega\tilde{L}^\dagger\tilde{L}\right\|_1 && \text{(2.E.12)} \\
&= \left\| L\omega L^\dagger - \tilde{L}\omega L^\dagger + \tilde{L}\omega L^\dagger - \tilde{L}\omega\tilde{L}^\dagger\right\|_1 \\
&\qquad + \frac{1}{2}\left\| L^\dagger L\omega - \tilde{L}^\dagger L\omega + \tilde{L}^\dagger L\omega - \tilde{L}^\dagger\tilde{L}\omega\right\|_1 \\
&\qquad + \frac{1}{2}\left\|\omega L^\dagger L - \omega\tilde{L}^\dagger L + \omega\tilde{L}^\dagger L - \omega\tilde{L}^\dagger\tilde{L}\right\|_1 && \text{(2.E.13)} \\
&\leq \left\| L\omega L^\dagger - \tilde{L}\omega L^\dagger\right\|_1 + \left\|\tilde{L}\omega L^\dagger - \tilde{L}\omega\tilde{L}^\dagger\right\|_1 \\
&\qquad + \frac{1}{2}\left\| L^\dagger L\omega - \tilde{L}^\dagger L\omega\right\|_1 + \frac{1}{2}\left\|\tilde{L}^\dagger L\omega - \tilde{L}^\dagger\tilde{L}\omega\right\|_1 \\
&\qquad + \frac{1}{2}\left\|\omega L^\dagger L - \omega\tilde{L}^\dagger L\right\|_1 + \frac{1}{2}\left\|\omega\tilde{L}^\dagger L - \omega\tilde{L}^\dagger\tilde{L}\right\|_1 && \text{(2.E.14)} \\
&\leq \left\| L - \tilde{L}\right\|_2 \|\omega\| \left\| L^\dagger\right\|_2 + \left\|\tilde{L}\right\|_2 \|\omega\| \left\| L^\dagger - \tilde{L}^\dagger\right\|_2 \\
&\qquad + \|\omega\| \|L\|_2 \left\| L^\dagger - \tilde{L}^\dagger\right\|_2 + \|\omega\| \left\|\tilde{L}^\dagger\right\|_2 \left\| L - \tilde{L}\right\|_2 && \text{(2.E.15)} \\
&= O(\varepsilon/t). && \text{(2.E.16)}
\end{aligned}
$$

The first and second inequalities follow from the triangle inequality. The last inequality follows from the generalized Hölder's inequality (see, e.g., [Bei13, Eq. (8)]). The final

equality follows from the assumption in the theorem statement. We thus conclude that

$$\left\|\mathcal{L} - \tilde{\mathcal{L}}\right\|_\diamond = O(\varepsilon/t), \tag{2.E.17}$$

and finally, by combining with (2.E.1)–(2.E.8) that

$$\frac{1}{2}\left\|e^{\tilde{\mathcal{L}}t} - e^{\mathcal{L}t}\right\|_\diamond \leq \left\|\mathcal{L} - \tilde{\mathcal{L}}\right\|_\diamond t = O(\varepsilon/t)t = O(\varepsilon). \tag{2.E.18}$$

2.F Proof of Lemma 18

The steps are essentially the same as in the proof of [Dup10, Lemma I.4], but here we are considering bipartite vectors and their associated operators. We give the proof for convenience. Consider the following:

$$\begin{aligned}
&\frac{1}{2}\left\||\tilde{\psi}\rangle\langle\tilde{\psi}| - |\psi\rangle\langle\psi|\right\|_1 \\
&= \frac{1}{2}\left\|(\tilde{L}\otimes I)|\Gamma\rangle\langle\Gamma|(\tilde{L}^\dagger\otimes I) - (L\otimes I)|\Gamma\rangle\langle\Gamma|(L^\dagger\otimes I)\right\|_1 && \text{(2.F.1)}\\
&= \sqrt{1 - \left|\langle\Gamma|\left(\tilde{L}^\dagger L\otimes I\right)|\Gamma\rangle\right|^2} && \text{(2.F.2)}\\
&= \sqrt{1 - \left|\sum_{i,j}\langle i|\langle i|\left(\tilde{L}^\dagger L\otimes I\right)|j\rangle|j\rangle\right|^2} && \text{(2.F.3)}\\
&= \sqrt{1 - \left|\sum_{i,j}\langle i|\left(\tilde{L}^\dagger L\right)|j\rangle\otimes\langle i|j\rangle\right|^2} && \text{(2.F.4)}\\
&= \sqrt{1 - \left|\sum_{i}\langle i|\left(\tilde{L}^\dagger L\right)|i\rangle\right|^2} && \text{(2.F.5)}\\
&= \sqrt{1 - \left|\mathrm{Tr}\left[\tilde{L}^\dagger L\right]\right|^2} && \text{(2.F.6)}\\
&= \sqrt{1 + \left|\mathrm{Tr}\left[\tilde{L}^\dagger L\right]\right|}\sqrt{1 - \left|\mathrm{Tr}\left[\tilde{L}^\dagger L\right]\right|} && \text{(2.F.7)}\\
&\leq \sqrt{2\left(1 - \left|\mathrm{Tr}\left[\tilde{L}^\dagger L\right]\right|\right)} && \text{(2.F.8)}
\end{aligned}$$

$$\leq \sqrt{2\left(1 - \operatorname{Re}\left[\operatorname{Tr}\left[\tilde{L}^\dagger L\right]\right]\right)} \tag{2.F.9}$$

$$= \left\|\tilde{L} - L\right\|_2 \tag{2.F.10}$$

$$\leq \delta. \tag{2.F.11}$$

The second equality follows from the fact that the trace distance between two pure quantum states can be expressed in terms of their inner product (see proof of [Wil17, Theorem 9.3.1]); i.e., for quantum states $|\psi\rangle$ and $|\phi\rangle$, we have

$$\frac{1}{2}\left\||\phi\rangle\langle\phi| - |\psi\rangle\langle\psi|\right\|_1 = \sqrt{1 - |\langle\psi|\phi\rangle|^2}. \tag{2.F.12}$$

The third-to-last inequality follows because $\left|\operatorname{Tr}\left[\tilde{L}^\dagger L\right]\right| \leq 1$, which is a consequence of the Cauchy–Schwarz inequality for the Hilbert–Schmidt inner product and the assumption that $\|L\|_2 = \left\|\tilde{L}\right\|_2 = 1$. The last equality comes from the following chain of equalities:

$$\left\|\tilde{L} - L\right\|_2 = \sqrt{\operatorname{Tr}\left[\left(\tilde{L}^\dagger - L^\dagger\right)\left(\tilde{L} - L\right)\right]} \tag{2.F.13}$$

$$= \sqrt{\operatorname{Tr}\left[\tilde{L}^\dagger\tilde{L} - \tilde{L}^\dagger L - L^\dagger\tilde{L} + L^\dagger L\right]} \tag{2.F.14}$$

$$= \sqrt{\operatorname{Tr}\left[\tilde{L}^\dagger\tilde{L}\right] - \operatorname{Tr}\left[\tilde{L}^\dagger L\right] - \operatorname{Tr}\left[L^\dagger\tilde{L}\right] + \operatorname{Tr}[L^\dagger L]} \tag{2.F.15}$$

$$= \sqrt{\left\|\tilde{L}\right\|_2^2 - 2\operatorname{Re}\left[\operatorname{Tr}\left[\tilde{L}^\dagger L\right]\right] + \|L\|_2^2} \tag{2.F.16}$$

$$= \sqrt{2\left(1 - \operatorname{Re}\left[\operatorname{Tr}\left[\tilde{L}^\dagger L\right]\right]\right)}, \tag{2.F.17}$$

where the first equality follows from the definition of Hilbert–Schmidt norm (see (2.2.8)).

CHAPTER 3

QUANTUM BOLTZMANN MACHINE LEARNING OF GROUND-STATE ENERGIES[1]

3.1 Abstract

Estimating the ground-state energy of Hamiltonians is a fundamental task for which it is believed that quantum computers can be helpful. Several approaches have been proposed toward this goal, including algorithms based on quantum phase estimation and hybrid quantum-classical optimizers involving parameterized quantum circuits, the latter falling under the umbrella of the variational quantum eigensolver. Here, we analyze the performance of quantum Boltzmann machines for this task, which is a less explored ansatz based on parameterized thermal states and which is not known to suffer from the barren-plateau problem. We delineate a hybrid quantum-classical algorithm for this task and rigorously prove that it converges to an ε-approximate stationary point of the energy function optimized over parameter space, while using a number of parameterized-thermal-state samples that is polynomial in ε^{-1}, the number of parameters, and the norm of the Hamiltonian being optimized. Our algorithm estimates the gradient of the energy function efficiently by means of a quantum circuit construction that combines classical random sampling, Hamiltonian simulation, and the Hadamard test. Additionally, supporting our main claims are calculations of the gradient and Hessian of the energy function, as well as an upper bound on the matrix elements of the latter that is used in the convergence analysis.

[1]This chapter is based verbatim on the work [PKPW24], with typos corrected and some modifications, including the addition of an appendix section that directly addresses the open problem of QBM learning. The arXiv version will be updated soon.

3.2 Introduction

Calculating the ground-state energies of Hamiltonians is one of the chief goals of quantum physics [Lie05]. This is typically the first step employed in computing energetic properties of molecules and materials, and it thus has wide-ranging applications in materials science [SH09], condensed-matter physics [Con21], and quantum chemistry [DSL15].

Stemming from the exponential growth of the state space as the number of particles increases, calculating ground-state energies is generally a difficult problem, and in fact it has been rigorously proven that the worst-case complexity of doing so for physically relevant Hamiltonians is computationally difficult in principle, even for a quantum computer [SV09, CGW14, Hua21]. In spite of this complexity-theoretic barrier and due to the aforementioned applications, many approaches have emerged for calculating ground-state energies on classical computers. One of the oldest and most widely used approaches is based on the variational principle [GRS83], in which one reduces the search space by parameterizing a family of trial ground states and then searches over this reduced space by means of gradient-descent like algorithms. This has culminated in powerful methods like matrix product states [FNW92, VC06, PGVWC07], which perform well in practice.

In another direction, Ref. [AL99] has argued that quantum computers could be effective at calculating ground-state energies, due to their ability to simulate quantum mechanical processes faithfully and with reduced overhead, in principle, when compared to classical algorithms. Building upon [AL99], one of the first approaches proposed for doing so involves employing the quantum phase estimation algorithm for small molecules [AGDLHG05]. More recently, other phase-estimation-based algorithms for ground-state energy estimation have been proposed and analyzed [LT22, DLT22, WBC22, DL23, WFZ+23, WFRJ23], with the goal being to reduce the resources required, in a way

that is more amenable to "early fault-tolerant" quantum processors. All of these approaches assume the availability of an initial trial state that has non-trivial overlap with the true ground state.

Due to the approach of [AGDLHG05] requiring quantum circuits of large depth (i.e., a long sequence of consecutive quantum logic gates), researchers subsequently proposed the variational quantum eigensolver (VQE) as another approach for the ground-state energy estimation problem [PMS+14]. The VQE approach employs parameterized quantum circuits (PQCs) of shorter depth and involves a hybrid interaction between such shorter-depth quantum circuits and a classical optimizer. Interestingly, the VQE approach provides a quantum computational implementation of the aforementioned variational method. While the VQE approach at first seemed promising, later research pointed out a number of bottlenecks associated with it [TCC+22], which will likely preclude VQE from achieving practical quantum advantage in the near term. One of the primary bottlenecks is the barren-plateau problem [MBS+18, MKW21, AHCC22, HSCC22, FHC+24, RBS+24], in which the landscape of the objective function becomes extremely flat, so that a costly (exponential) number of measurements is required to determine which direction the optimizer should proceed to next, at any given iteration of the algorithm.

While the VQE approach is based on employing parameterized quantum circuits (a particular ansatz for generating trial states), an alternate ansatz involves using quantum Boltzmann machines (QBMs) [AAR+18, BRGBPO17, KW17], and this is the approach that we pursue and analyze here for ground-state energy estimation. Indeed, in the QBM approach to ground-state energy estimation, one substitutes parameterized quantum circuits with parameterized thermal states of a given Hamiltonian and performs the search over parameterized thermal states. Furthermore, the QBM approach appears to be viable, due to significant recent progress on the problem of preparing thermal states on quan-

tum computers [CKBG23, CKG23, BCL24, CLLY24, RW24, RFA24, BLMT24, DLLZ24], in spite of known worst-case complexity-theoretic barriers [BCGW22]. Hitherto, QBMs have been analyzed in the context of Hamiltonian learning [AAKS21, GPBB+24] and generative modeling [CB24], but, to the best of our knowledge, they have not been considered yet for ground-state energy estimation. Another significant and promising aspect of QBMs is that there is evidence that they do not suffer from the barren-plateau problem in certain contexts [CB24]. In this context, we should also note that [OMKW21] proved that QBMs with hidden units can suffer from the barren plateau problem, assuming a particular approach to generating parameterized thermal states randomly; however, this statement is not applicable to QBMs with visible units only, i.e., the model that we employ here (see [OMKW21] for definitions of hidden and visible units in QBMs). Indeed it was proven in [CB24] that QBMs with visible units do not suffer from the barren plateau problem when used in the context of generative modeling, and much more so, they provably converge in this setting, using a number of thermal state samples polynomial in the number of qubits.

3.3 Summary of Main Results

The main finding of this chapter is a rigorous mathematical proof that the QBM learning approach to approximating ground-state energies is *sample efficient*, in the sense that the number of samples of parameterized thermal states used by our algorithm is polynomial in several quantities of interest, the latter to be clarified later. In doing so, we also overcome a key obstacle to efficient training of QBMs, discussed in further detail in what follows.

In more detail, we adopt a hybrid quantum-classical approach, similar to what is used in VQE, but we instead replace PQCs with QBMs, as mentioned above. Let H denote the Hamiltonian of interest, which we assume can be efficiently measured on a quantum computer. We suppose that this Hamiltonian acts on n qubits, but let us note that all of the analysis and algorithms that follow apply also to qudit systems (d-dimensional systems). The Hamiltonian H can be efficiently measured when

$$H = \sum_{k=1}^{K} \alpha_k H_k, \tag{3.3.1}$$

where, for all $k \in [K]$, the coefficient $\alpha_k \in \mathbb{R}$ and H_k is a local Hamiltonian acting on a constant number of particles. Without loss of generality, we assume that $\|H_k\| \leq 1$ by absorbing the norm of H_k into α_k, so that

$$\|H\| \leq \sum_{k=1}^{K} |\alpha_k| \, \|H_k\| \leq \sum_{k=1}^{K} |\alpha_k| =: \|\alpha\|_1 , \tag{3.3.2}$$

and we also assume that $\alpha_k > 0$ for all $k \in [K]$, because any negative sign for α_k can be absorbed into H_k. Let

$$G(\theta) := \sum_{j=1}^{J} \theta_j G_j \tag{3.3.3}$$

be a trial Hamiltonian, with $J \in \mathbb{N}$, each $\theta_j \in \mathbb{R}$ a parameter,

$$\theta := (\theta_1, \ldots, \theta_J), \tag{3.3.4}$$

and each G_j a Hamiltonian that acts locally on a constant-sized set of qubits. We assume that we have some knowledge of the problem at hand and that therefore we can appropriately choose a subspace of parameters such that $J = \text{poly}(n)$. Furthermore, similar to [AAKS21, CB24], we assume that samples of the thermal state

$$\rho(\theta) := \frac{1}{Z(\theta)} e^{-G(\theta)}, \tag{3.3.5}$$

where

$$Z(\theta) := \text{Tr}[e^{-G(\theta)}], \tag{3.3.6}$$

are available, for every possible choice of $\theta \in \mathbb{R}^J$. As such, the QBM model that we employ here has only visible units and no hidden units, using the terminology of [AAR+18, KW17].

The following inequality is a basic consequence of the variational principle:

$$\inf_{\rho \in \mathcal{D}} \mathrm{Tr}[H\rho] \leq \inf_{\theta \in \mathbb{R}^J} f(\theta), \tag{3.3.7}$$

$$\text{where } f(\theta) := \mathrm{Tr}[H\rho(\theta)], \tag{3.3.8}$$

and $\mathcal{D}$ represents the set of all possible quantum states acting on the same Hilbert space on which H acts (i.e., $\mathcal{D}$ is the set of all such density operators, which are unit trace, positive semi-definite operators). The inequality in (3.3.7) indicates that the true ground-state energy is bounded from above by the minimal energy of the Hamiltonian H over every possible trial state $\rho(\theta)$.

With these notions in place, we can state our main claim: finding an ε-approximate stationary point of $f(\theta)$ is sample efficient, in the sense that our algorithm uses a number of parameterized-thermal-state samples that is polynomial in ε^{-1}, J, and $\|\alpha\|_1$. Since the function $\theta \mapsto f(\theta)$ is generally non-convex, finding an ε-stationary point (local minimum), rather than a global minimum, is essentially the best that one can hope for when using this approach. Indeed, we argue in Appendix 3.B.2 that even the following basic instance of $f(\theta)$ is non-convex:

$$(\theta_1, \theta_2) \mapsto \mathrm{Tr}\left[\sigma_Y \frac{e^{-G(\theta_1,\theta_2)}}{\mathrm{Tr}[e^{-G(\theta_1,\theta_2)}]}\right] \tag{3.3.9}$$

with $G(\theta_1, \theta_2) = \theta_1 \sigma_X + \theta_2 \sigma_Y$.

One of the essential steps in optimizing $f(\theta)$ in (3.3.8) is to determine its gradient. This is needed in any gradient-descent like algorithm, in order to determine which step to take next in an iterative search. An analytical form for the gradient $\nabla_\theta f(\theta)$ is based on an

analytical form for $\nabla_\theta \rho(\theta)$, the latter of which follows from the developments in [Has07], [AAKS21, Appendix B], and [CB24, Lemma 5] (see also [Kim12, Section III-C] and [KB19, Section IV-A]). In more detail, it follows from these works that

$$\partial_j f(\theta) = -\frac{1}{2}\left\langle \left\{ H, \Phi_\theta(G_j) \right\} \right\rangle + \langle H \rangle \left\langle G_j \right\rangle \tag{3.3.10}$$

$$= -\frac{1}{2}\left\langle \left\{ H - \langle H \rangle, \widetilde{G_j} - \left\langle \widetilde{G_j} \right\rangle \right\} \right\rangle, \tag{3.3.11}$$

where $\partial_j \equiv \frac{\partial}{\partial \theta_j}$,

$$\{A, B\} := AB + BA \tag{3.3.12}$$

denotes the anticommutator of operators A and B,

$$\widetilde{G_j} \equiv \Phi_\theta(G_j), \tag{3.3.13}$$

$$\langle C \rangle := \mathrm{Tr}[C\rho(\theta)], \tag{3.3.14}$$

for a Hermitian operator C, and Φ_θ is the following quantum channel:

$$\Phi_\theta(X) := \int_{\mathbb{R}} dt\ p(t)\ e^{-iG(\theta)t} X e^{iG(\theta)t}\ , \tag{3.3.15}$$

$$\text{with } p(t) := \frac{2}{\pi} \ln |\coth(\pi t/2)| \tag{3.3.16}$$

a probability density function on $t \in \mathbb{R}$ (we refer to $p(t)$ as the "high-peak-tent" probability density function, due to the form of its graph when plotted). We also used that $\left\langle \Phi_\theta(G_j) \right\rangle = \left\langle G_j \right\rangle$, which follows because $\Phi_\theta(\rho(\theta)) = \rho(\theta)$. Prior work [Has07, AAKS21, CB24] refers to the map Φ_θ as the quantum belief propagation superoperator. Here we observe that it is in fact a quantum channel (completely positive, trace-preserving map), due to the fact that $p(t)$ is a probability density function. Note that this is remarked upon (without proof) in [Kim12, Footnote 32].

Supporting our main finding are various contributions of our work, which we list now. Here we prove that the gradient $\nabla_\theta f(\theta)$ is Lipschitz continuous, which is needed

to make rigorous claims about the convergence of the stochastic gradient descent (SGD) algorithm. Moreover (and essential to our overall algorithm), we demonstrate how the gradient $\nabla_\theta f(\theta)$ can be efficiently estimated on a quantum computer, which is a consequence of the formula in (3.3.10) and the observation that $p(t)$ is a probability density function. That is, we provide an efficient quantum algorithm, called the *quantum Boltzmann gradient estimator*, that computes an unbiased estimate of $\nabla_\theta f(\theta)$. By doing so, we have thus overcome a key obstacle in QBM learning going back to [AAR+18, Section II], in which it was previously thought that estimating the gradient could not be done efficiently (see also [KW17, WW19, AC19, Kap20, ZLW21] for similar previous discussions on the perceived difficulty of training QBMs by directly estimating the gradient). We discuss this in more detail in Appendix 3.C. Having an unbiased estimator is also helpful in analyzing the convergence of SGD. With these analytical results in place, we then invoke known results [KR20, Corollary 1] on the convergence of SGD to conclude that the sample complexity of our algorithm for finding an ε-stationary point of $\theta \mapsto f(\theta)$ is polynomial in ε^{-1}, J, and $\|\alpha\|_1$ (recall that sample complexity here is the number of parameterized thermal states needed). This summarizes the main contributions of this chapter.

Our results reported here can be contrasted with an analytical study of VQE and PQCs [HN21]. Indeed, therein, the authors studied the convergence of VQE and PQCs when performing analytic measurements of the gradient of the cost function (a first-order method), as compared to a gradient-free method of measuring the cost function directly (a zeroth-order method). They found that, for certain Hamiltonians, VQE algorithms employing an analytic gradient measurement (first-order methods) are faster than zeroth-order methods. However, their analysis was restricted to non-interacting Hamiltonians, for which one can actually calculate the ground-state energy by hand; regardless, the authors suggested that their analytical finding should be indicative of what one might find

for more complex Hamiltonians. In contrast, our analysis applies to all Hamiltonians that are efficiently measurable on quantum computers, thus encompassing a significantly wider class of Hamiltonians.

In what follows, we provide further details of our results, while the appendices give complete proofs of all of our claims.

3.4 Calculation of Gradient and Hessian

Let us first briefly review the proof of the equality in (3.3.11), which begins with the following equality:

$$\partial_j \rho(\theta) = -\frac{1}{2}\left\{\Phi_\theta(G_j), \rho(\theta)\right\} + \rho(\theta)\left\langle G_j \right\rangle, \tag{3.4.1}$$

along with some further algebraic manipulations. To see (3.4.1), consider that

$$\partial_j \rho(\theta) = \partial_j \left[\frac{1}{Z(\theta)} e^{-G(\theta)}\right] \tag{3.4.2}$$

$$= -\frac{1}{Z(\theta)^2}\left[\partial_j Z(\theta)\right] e^{-G(\theta)} + \frac{1}{Z(\theta)} \partial_j e^{-G(\theta)} \tag{3.4.3}$$

$$= -\frac{\rho(\theta)}{Z(\theta)}\left[\partial_j \operatorname{Tr}[e^{-G(\theta)}]\right] + \frac{1}{Z(\theta)} \partial_j e^{-G(\theta)}. \tag{3.4.4}$$

Now recall [Has07, Eq. (9)] (however, for the precise statement that we use, see [AAKS21, Proposition 20] and [CB24, Lemma 5]):

$$\partial_j e^{-G(\theta)} = -\frac{1}{2}\left\{\Phi_\theta(G_j), e^{-G(\theta)}\right\}. \tag{3.4.5}$$

It was proven in [AAKS21, Appendix B] that $\frac{\tanh(\omega/2)}{\omega/2}$ is the Fourier transform of $p(t)$ in (3.3.16), i.e.,

$$\frac{\tanh(\omega/2)}{\omega/2} = \int_{-\infty}^{\infty} dt\ p(t) e^{-i\omega t}, \tag{3.4.6}$$

and that it has the explicit form given in (3.3.16). The latter implies that $p(t)$ is a probability density function and thus that Φ_θ is a quantum channel. As mentioned above, this observation is paramount later on for our quantum circuit construction that provides an unbiased estimate of $\partial_j f(\theta)$. Now plugging (3.4.5) into (3.4.4) and simplifying, we conclude (3.4.1). Plugging (3.4.1) into $\partial_j f(\theta)$ and simplifying, we arrive at (3.3.10). Further algebraic manipulations lead to (3.3.11). See Appendix 3.B.3.

We also compute the Hessian of $f(\theta)$. Due to the length of the expression, we only include it in Appendix 3.B.4, along with its derivation. While this quantity can also be efficiently estimated on a quantum computer (as argued in the supplementary material) and incorporated into a Newton method search (i.e., an extension of gradient descent that incorporates second-derivative information), we mainly use it to determine a Lipschitz constant for the gradient $\nabla_\theta f(\theta)$, which in turn implies rigorous statements about the convergence of SGD, as previously mentioned.

3.5 Lipschitz Constant for Gradient

By bounding the matrix elements of the Hessian of the objective function $f(\theta)$, we can use it to establish a Lipschitz constant for its gradient $\nabla_\theta f(\theta)$. Indeed, recalling that

$$\|A\| := \sup_{|\psi\rangle : \||\psi\rangle\| = 1} \|A|\psi\rangle\| \,, \tag{3.5.1}$$

we find that

$$\left|\partial_j \partial_k f(\theta)\right| \leq 8 \|H\| \left\|G_j\right\| \|G_k\| \,, \tag{3.5.2}$$

which we can substitute into [PCW24, Lemma 8], in order to conclude the following Lipschitz constant for the gradient $\nabla_\theta f(\theta)$:

$$8J \|H\| \max \left\{ \left\| G_j \right\|^2 \right\}_{j=1}^{J}. \tag{3.5.3}$$

If the Hamiltonian H is of the form in (3.3.1), then a Lipschitz constant ℓ for the gradient $\nabla_\theta \operatorname{Tr}[H\rho(\theta)]$ is as follows:

$$\ell := 8J \|\alpha\|_1 \max \left\{ \left\| G_j \right\|^2 \right\}_{j=1}^{J}. \tag{3.5.4}$$

As we will see, this Lipschitz constant for the gradient $\nabla_\theta f(\theta)$ implies that the sample complexity of SGD is polynomial in J and $\|\alpha\|_1$. Let us finally note that ℓ is also called a smoothness parameter for $f(\theta)$.

3.6 Quantum Algorithm for Gradient Estimation

In the mth step of the SGD algorithm (reviewed in Appendix 3.A.3), one updates the parameter vector θ according to the following rule:

$$\theta_{m+1} = \theta_m - \eta \overline{g}(\theta_m), \tag{3.6.1}$$

where $\eta > 0$ is the learning rate and $\overline{g}(\theta_m)$ is a stochastic gradient evaluated at θ_m. The stochastic gradient $\overline{g}(\theta)$ should be unbiased, in the sense that $\mathbb{E}[\overline{g}(\theta)] = \nabla_\theta f(\theta)$ for all $\theta \in \mathbb{R}^J$, where the expectation is over all the randomness associated with the generation of $\overline{g}(\theta)$. As such, it is necessary to have a method for generating the stochastic gradient $\overline{g}(\theta)$, and for this purpose, we prescribe a quantum algorithm based on (3.3.10), which we call the *quantum Boltzmann gradient estimator*.

Consider that (3.3.10) is a linear combination of two terms. As such, we delineate one procedure that estimates $\frac{1}{2}\left\langle \left\{ H, \Phi_\theta(G_j) \right\} \right\rangle$ and another that estimates $\langle H \rangle \left\langle G_j \right\rangle$. The second

term $\langle H \rangle \left\langle G_j \right\rangle$ is simpler: since it can be written as

$$\langle H \rangle \left\langle G_j \right\rangle = \mathrm{Tr}[\left(H \otimes G_j\right)(\rho(\theta) \otimes \rho(\theta))], \tag{3.6.2}$$

a procedure for estimating it is to generate the state $\rho(\theta) \otimes \rho(\theta)$ and then measure the observable $H \otimes G_j$ on these two copies. Through repetition, the estimate of $\langle H \rangle \left\langle G_j \right\rangle$ can be made as precise as desired. This procedure is described in detail in Appendix 3.B.6 as Algorithm 2, the result of which is that $O(\|\alpha\|_1^2 \, \varepsilon^{-2} \ln \delta^{-1})$ samples of $\rho(\theta)$ are required to have an accuracy of $\varepsilon > 0$ with a failure probability of $\delta \in (0, 1)$, when H is of the form in (3.3.1).

Our quantum algorithm for estimating the first term in (3.3.10) is more intricate. Under the assumption that H has the form in (3.3.1), it follows by direct substitution of (3.3.1) and (3.3.15), as well as the fact that $p(t)$ is an even function, that

$$\frac{1}{2}\left\langle \left\{H, \Phi_\theta(G_j)\right\} \right\rangle = \frac{1}{2}\mathrm{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)] = \sum_k \alpha_k \int_{-\infty}^{\infty} dt \; p(t) \; \mathrm{Re}[\mathrm{Tr}[U_{j,k}(\theta, t)\rho(\theta)]], \tag{3.6.3}$$

where

$$U_{j,k}(\theta, t) := H_k e^{-iG(\theta)t} G_j e^{iG(\theta)t}. \tag{3.6.4}$$

If H_k and G_j are also unitaries (which occurs in a rather standard case that they are Pauli strings, i.e., tensor products of Pauli matrices), then we can estimate the first term in (3.3.10) by a combination of classical sampling, Hamiltonian simulation [Llo96, CMN+18], and the Hadamard test [CEMM98]. This is the key insight behind the quantum Boltzmann gradient estimator. Indeed, the basic idea is to sample k with probability $\alpha_k / \|\alpha\|_1$ and t with probability density $p(t)$. Based on these choices, we then execute the quantum circuit in Figure 3.1, which outputs a Rademacher random variable Y, that has a realization $y = +1$ occurring with probability

$$\frac{1}{2}\left(1 + \mathrm{Re}[\mathrm{Tr}[U_{j,k}(\theta, t)\rho(\theta)]]\right) \tag{3.6.5}$$

and $y = -1$ occurring with probability

$$\frac{1}{2}\left(1 - \mathrm{Re}[\mathrm{Tr}[U_{j,k}(\theta,t)\rho(\theta)]]\right). \tag{3.6.6}$$

Thus, the expectation of a random variable $Z = (-1)^{Y+1}$, conditioned on k and t, is equal to $-\mathrm{Re}[\mathrm{Tr}[U_{j,k}(\theta,t)\rho(\theta)]]$ and including the further averaging over k and t, the expectation is equal to

$$-\frac{1}{2\left\|\alpha\right\|_1}\mathrm{Tr}[\left\{H,\Phi_\theta(G_j)\right\}\rho(\theta)]. \tag{3.6.7}$$

As such, we can sample k, t, and Z in this way, and averaging the outcomes and scaling by $\|\alpha\|_1$ gives an unbiased estimate of the first term in (3.3.10). This procedure is described in detail in Appendix 3.B.6 as Algorithm 1, the result of which is that $O(\|\alpha\|_1^2\,\varepsilon^{-2}\ln\delta^{-1})$ samples of $\rho(\theta)$ are required to have an accuracy of $\varepsilon > 0$ with a failure probability of $\delta \in (0,1)$, when H is of the form in (3.3.1). We finally note that this construction can straightforwardly be generalized beyond the case of H_k and G_j being Pauli strings, if they instead are block encoded into unitary circuits [LC19, GSLW19].

Thus, through this combination of classical random sampling and quantum circuitry, we can produce an unbiased estimate of the first term in (3.3.10). Adding this estimate and the one from the paragraph surrounding (3.6.2) then leads to the quantum Boltzmann gradient estimator, which realizes an unbiased estimate of (3.3.10).

3.7 Ground-State Energy Estimation Algorithm and its Performance

Finally, we assemble everything presented so far and describe our algorithm for ground-state energy estimation, along with guarantees on its performance.

Algorithm 1 (QBM-GSE) *Fix $\varepsilon \in (0,1)$. The algorithm for converging to an ε-stationary point*

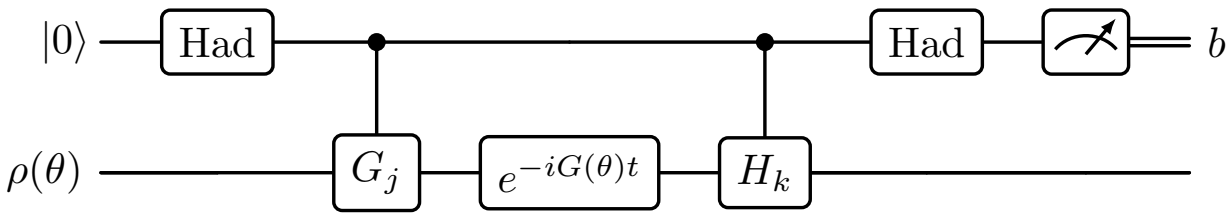


Figure 3.1: Quantum circuit that plays a role in realizing an unbiased estimate of $-\frac{1}{2}\left\langle\left\{H, \Phi_\theta(G_j)\right\}\right\rangle$. The Boltzmann gradient estimator combines this estimate with an unbiased estimate of $\langle H\rangle\left\langle G_j\right\rangle$, to realize an unbiased estimate of the gradient $\nabla_\theta f(\theta)$ in (3.3.10).

of $f(\theta)$ in (3.3.8) consists of the following steps:

1. *Initialize $\theta_0 \in \mathbb{R}^J$. Set the learning rate η and the number M of iterations as follows:*

$$\eta = \frac{1}{\ell}, \qquad M \geq \left\lceil \frac{12\Delta\ell}{\varepsilon^2} \right\rceil, \tag{3.7.1}$$

   where the smoothness parameter ℓ is defined in (3.5.4) and $\Delta := f(\theta_0) - \inf_{\theta\in\mathbb{R}^J} f(\theta)$. Set $m = 0$.

2. *Execute Algorithms 2 and 1 (detailed in Appendix 3.B.6) to calculate $\overline{g}(\theta_m)$, which is a stochastic gradient satisfying $\mathbb{E}[\overline{g}(\theta)] = \nabla_\theta f(\theta)$.*

3. *Apply the update: $\theta_{m+1} = \theta_m - \eta\overline{g}(\theta_m)$. Set $m = m + 1$.*

4. *Repeat steps 2-3 $M - 1$ more times and output an estimate of $\mathrm{Tr}[H\rho(\theta_M)]$ (the latter obtained by measuring H with respect to the state $\rho(\theta_M)$, i.e., through sampling and averaging).*

By invoking [KR20, Corollary 1] and further analysis from Appendix 3.B.8, we conclude the following convergence guarantee for the QBM-GSE algorithm:

Theorem 20 *The QBM-GSE algorithm converges to an ε-stationary point of $f(\theta)$ in (3.3.8), i.e., such that*

$$\min_{m\in\{1,\ldots,M\}} \mathbb{E}\,\|\nabla_\theta f(\theta_m)\| \leq \varepsilon. \tag{3.7.2}$$

Note that θ_m in Theorem 20 is a random variable, as given in the QBM-GSE algorithm, and the expectation in (3.7.2) is with respect to the randomness associated with generating θ_m.

The above statement in Theorem 20 implies that the number M of steps that the QBM-GSE algorithm requires to converge to an ε-stationary point of $f(\theta)$ is polynomial in ε^{-1}, J, and $\|\alpha\|_1$. The number M of steps is then directly related to the sample complexity of the algorithm. By combining Theorem 20 and the Hoeffding bound, the QBM-GSE algorithm uses at least the following number of parameterized-thermal-state samples:

$$2J\left\lceil \frac{12\ell\Delta}{\varepsilon^2}\right\rceil\left\lceil \frac{8J\,\|\alpha\|_1^2\,\ln(16J\,\|\alpha\|_1^2/\varepsilon^2)}{\varepsilon^2}\right\rceil, \tag{3.7.3}$$

where ℓ is defined in (3.5.4). As such, the sample complexity of our algorithm is polynomial in ε^{-1}, J, and $\|\alpha\|_1$, as claimed. Thus, if J and $\|\alpha\|_1$ are polynomial in n (the number of qubits), then the sample complexity of the QBM-GSE algorithm is also polynomial in n.

3.8 Discussion

Our algorithm is most pertinent for the situation in which low-temperature thermal states of H are computationally difficult to generate but thermal states of $G(\theta)$ are not. In such a situation, one can use our algorithm as a means for approximating the ground-state energy of H. This is similar to the scenario considered in VQE: there the VQE algorithm

is also most applicable when the ground state of H is computationally difficult to generate but states realized by PQCs are not, which is indeed the case for short-depth PQCs. However, as emphasized previously, in contrast to VQE, QBMs are not known to suffer from the barren-plateau problem, and there is evidence that they do not in certain contexts [CB24]. As such, they appear to offer a more promising route for ground-state energy estimation.

3.9 Conclusion and Outlook

In this chapter, we have analyzed quantum Boltzmann machine learning of ground-state energies. Our main result is an algorithm that converges to an ε-approximate stationary point of $f(\theta)$ in (3.3.8), along with a rigorous claim about its convergence. Namely, for Hamiltonians of the form in (3.3.1), we have proven that the sample complexity of the algorithm is polynomial in ε^{-1}, J, and $\|\alpha\|_1$, where $\|\alpha\|_1$ is defined in (3.3.2). Our algorithm welds together conventional stochastic gradient descent and a novel quantum circuit for estimating the gradient $\nabla_\theta f(\theta)$, the latter being the core component of the quantum Boltzmann gradient estimator. Supporting our main claims are calculations of the gradient, Hessian, and smoothness parameter of the objective function $f(\theta)$, along with various observations about the gradient that lead to our quantum algorithm for estimating it.

We believe that our results have far-reaching consequences for QBM learning. Indeed, given that the quantum Boltzmann gradient estimator efficiently estimates (3.3.10), this approach now opens the door to using QBMs for efficient learning and optimization in a much wider variety of contexts. For example, one can substitute QBMs for PQCs in recent works on semi-definite programming and constrained Hamiltonian optimiza-

tion [PCW24, CWH+23] and entropy estimation [GPSW24] and analyze the sample complexity and convergence when doing so. We suspect that claims similar to those made here, regarding polynomial sample complexity for convergence to an ε-stationary point, can be made in these contexts; however, it remains a topic for future investigation.

Going forward from here, it is a pressing open question to determine whether the landscape of the objective function $f(\theta)$ suffers from the barren-plateau problem. In a different optimization problem involving QBMs [CB24], evidence was given supporting the conclusion that barren plateaus do not occur there. If the landscape of the objective function $f(\theta)$ does not suffer from the barren-plateau problem, this, along with our algorithm and further progress on thermal state preparation, would imply that QBMs are a viable path toward ground-state energy estimation and other learning and optimization problems.

3.10 Acknowledgments

We thank Paul Alsing, Nana Liu, and Soorya Rethinasamy for helpful discussions. DP and MMW acknowledge support from AFRL under agreement no. FA8750-23-2-0031.

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-23-2-0031. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

BIBLIOGRAPHY

[AAKS21] Anurag Anshu, Srinivasan Arunachalam, Tomotaka Kuwahara, and Mehdi Soleimanifar. Sample-efficient learning of interacting quantum systems. *Nature Physics*, 17(8):931–935, 2021. doi:10.1038/s41567-021-01232-0.

[AAR+18] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum Boltzmann machine. *Physical Review X*, 8:021050, May 2018. doi:10.1103/PhysRevX.8.021050.

[AC19] Eric R. Anschuetz and Yudong Cao. Realizing quantum Boltzmann machines through eigenstate thermalization, 2019. arXiv:1903.01359.

[AGDLHG05] Alan Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, September 2005. doi:10.1126/science.1113479.

[AHCC22] 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. doi:10.1088/2058-9565/ac7d06.

[AL99] Daniel S. Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters*, 83:5162–5165, December 1999. doi:10.1103/PhysRevLett.83.5162.

[BCGW22] Sergey Bravyi, Anirban Chowdhury, David Gosset, and Pawel Wocjan. Quantum Hamiltonian complexity in thermal equilibrium. *Nature Physics*, 18(11):1367–1370, 2022. doi:10.1038/s41567-022-01742-5.

[BCL24] Thiago Bergamaschi, Chi-Fang Chen, and Yunchao Liu. Quantum computational advantage with constant-temperature Gibbs sampling, 2024. arXiv:2404.14639.

[BLMT24] Ainesh Bakshi, Allen Liu, Ankur Moitra, and Ewin Tang. High-temperature Gibbs states are unentangled and efficiently preparable, 2024. arXiv:2403.16850.

[BRGBPO17] Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz. Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Physical Review X*, 7:041052, November 2017. doi:10.1103/PhysRevX.7.041052.

[CB24] Luuk Coopmans and Marcello Benedetti. On the sample complexity of quantum Boltzmann machine learning. *Communications Physics*, 7(1):274, 2024. doi:10.1038/s42005-024-01763-x.

[CEMM98] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society A*, 454(1969):339–354, 1998. doi:10.1098/rspa.1998.0164.

[CGW14] Andrew M. Childs, David Gosset, and Zak Webb. The Bose-Hubbard model is QMA-complete. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming*, pages 308–319, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-43948-7_26.

[CKBG23] Chi-Fang Chen, Michael J. Kastoryano, Fernando G. S. L. Brandão, and András Gilyén. Quantum thermal state preparation, 2023. arXiv:2303.18224.

[CKG23] Chi-Fang Chen, Michael J. Kastoryano, and András Gilyén. An efficient and exact noncommutative quantum Gibbs sampler, 2023. arXiv:2311.09207.

[CLLY24] Hongrui Chen, Bowen Li, Jianfeng Lu, and Lexing Ying. A randomized method for simulating Lindblad equations and thermal state preparation, 2024. arXiv:2407.06594.

[CMN+18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018. doi:10.1073/pnas.1801723115.

[Con21] Mucio A. Continentino. *Key Methods and Concepts in Condensed Matter Physics*. 2053-2563. IOP Publishing, 2021. doi:10.1088/978-0-7503-3395-5.

[CWH+23] Jingxuan Chen, Hanna Westerheim, Zoë Holmes, Ivy Luo, Theshani Nu-

radha, Dhrumil Patel, Soorya Rethinasamy, Kathie Wang, and Mark M. Wilde. Qslack: A slack-variable approach for variational quantum semidefinite programming, 2023. arXiv:2312.03830.

[DDG+22] Marina Danilova, Pavel Dvurechensky, Alexander Gasnikov, Eduard Gorbunov, Sergey Guminov, Dmitry Kamzolov, and Innokentiy Shibaev. *Recent Theoretical Advances in Non-Convex Optimization*, volume 191, pages 79–163. Springer International Publishing, April 2022. arXiv:2012.06188, doi:10.1007/978-3-031-00832-0_3.

[DL23] Zhiyan Ding and Lin Lin. Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *PRX Quantum*, 4:020331, May 2023. doi:10.1103/PRXQuantum.4.020331.

[DLLZ24] Zhiyan Ding, Bowen Li, Lin Lin, and Ruizhe Zhang. Polynomial-time preparation of low-temperature Gibbs states for 2D toric code, 2024. arXiv:2410.01206.

[DLT22] Yulong Dong, Lin Lin, and Yu Tong. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum*, 3:040305, October 2022. doi:10.1103/PRXQuantum.3.040305.

[DSL15] Peter Deglmann, Ansgar Schäfer, and Christian Lennartz. Application of quantum calculations in the chemical industry—an overview. *International Journal of Quantum Chemistry*, 115(3):107–136, 2015. doi:10.1002/qua.24811.

[FHC+24] Enrico Fontana, Dylan Herman, Shouvanik Chakrabarti, Niraj Kumar, Romina Yalovetzky, Jamie Heredge, Shree Hari Sureshbabu, and Marco Pistoia. Characterizing barren plateaus in quantum ansätze with the adjoint representation. *Nature Communications*, 15(1):7171, 2024. doi:10.1038/s41467-024-49910-w.

[FNW92] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Communications in Mathematical Physics*, 144(3):443–490, 1992. doi:10.1007/BF02099178.

[GPBB+24] Luis Pedro García-Pintos, Kishor Bharti, Jacob Bringewatt, Hossein Dehghani, Adam Ehrenberg, Nicole Yunger Halpern, and Alexey V. Gor-

shkov. Estimation of Hamiltonian parameters from thermal states. *Physical Review Letters*, 133:040802, July 2024. doi:10.1103/PhysRevLett.133.040802.

[GPSW24] Ziv Goldfeld, Dhrumil Patel, Sreejith Sreekumar, and Mark M. Wilde. Quantum neural estimation of entropies. *Physical Review A*, 109:032431, March 2024. doi:10.1103/PhysRevA.109.032431.

[GRS83] E. Gerjuoy, A. R. P. Rau, and Larry Spruch. A unified formulation of the construction of variational principles. *Reviews of Modern Physics*, 55(3):725–774, July 1983. doi:10.1103/RevModPhys.55.725.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 193–204, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316366.

[Has07] M. B. Hastings. Quantum belief propagation: An algorithm for thermal quantum systems. *Physical Review B*, 76:201102, November 2007. doi:10.1103/PhysRevB.76.201102.

[HN21] Aram W. Harrow and John C. Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Physical Review Letters*, 126:140502, April 2021. doi:10.1103/PhysRevLett.126.140502.

[HSCC22] Zoë Holmes, Kunal Sharma, Marco Cerezo, and Patrick J. Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3(1):010313, 2022. doi:10.1103/PRXQuantum.3.010313.

[Hua21] Yichen Huang. Two-dimensional local Hamiltonian problem with area laws is QMA-complete. *Journal of Computational Physics*, 443:110534, 2021. doi:10.1016/j.jcp.2021.110534.

[Kap20] H. J. Kappen. Learning quantum models from quantum or classical data. *Journal of Physics A: Mathematical and Theoretical*, 53(21):214001, May 2020. doi:10.1088/1751-8121/ab7df6.

[KB19] Kohtaro Kato and Fernando G. S. L. Brandão. Quantum approximate

Markov chains are thermal. *Communications in Mathematical Physics*, 370(1):117–149, 2019. doi:10.1007/s00220-019-03485-6.

[Kim12] Isaac H. Kim. Perturbative analysis of topological entanglement entropy from conditional independence. *Physical Review B*, 86:245116, December 2012. doi:10.1103/PhysRevB.86.245116.

[KR20] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world, July 2020. doi:10.48550/ARXIV.2002.03329.

[KW17] Mária Kieferová and Nathan Wiebe. Tomography and generative training with quantum Boltzmann machines. *Physical Review A*, 96:062327, December 2017. doi:10.1103/PhysRevA.96.062327.

[LC19] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, July 2019. doi:10.22331/q-2019-07-12-163.

[Lie05] Elliott H. Lieb. *The Stability of Matter: From Atoms to Stars*. Springer, Springer Berlin, Heidelberg, 2005. doi:10.1007/b138553.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996. doi:10.1126/science.273.5278.1073.

[LT22] Lin Lin and Yu Tong. Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum*, 3:010318, February 2022. doi:10.1103/PRXQuantum.3.010318.

[MBS+18] 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, November 2018. doi:10.1038/s41467-018-07090-4.

[MKW21] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2(4):040316, 2021. doi:10.1103/PRXQuantum.2.040316.

[OMKW21] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2:040316, October 2021. doi:10.1103/PRXQuantum.2.040316.

[PCW24] Dhrumil Patel, Patrick J. Coles, and Mark M. Wilde. Variational quantum algorithms for semidefinite programming. *Quantum*, 8:1374, June 2024. doi:10.22331/q-2024-06-17-1374.

[PGVWC07] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Information and Computation*, 7(5):401–430, July 2007. doi:10.26421/QIC7.5-6-1.

[PKPW24] Dhrumil Patel, Daniel Koch, Saahil Patel, and Mark M. Wilde. Quantum boltzmann machine learning of ground-state energies, October 2024. doi:10.48550/ARXIV.2410.12935.

[PMS+14] 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. doi:10.1038/ncomms5213.

[RBS+24] Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martín Larocca, and M. Cerezo. A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits. *Nature Communications*, 15(1):7172, 2024. doi:10.1038/s41467-024-49909-3.

[RFA24] Cambyse Rouzé, Daniel Stilck Franca, and Álvaro M. Alhambra. Efficient thermalization and universal quantum computing with quantum Gibbs samplers, 2024. arXiv:2403.12691.

[RW24] Joel Rajakumar and James D. Watson. Gibbs sampling gives quantum advantage at constant temperatures with $O(1)$-local Hamiltonians, 2024. arXiv:2408.01516.

[SH09] Martin O. Steinhauser and Stefan Hiermaier. A review of computational methods in materials science: Examples from shock-wave and polymer physics. *International Journal of Molecular Sciences*, 10(12):5135–5216, 2009. doi:10.3390/ijms10125135.

[SV09] Norbert Schuch and Frank Verstraete. Computational complexity of interacting electrons and fundamental limitations of density functional theory. *Nature Physics*, 5(10):732–735, 2009. doi:10.1038/nphys1370.

[TCC+22] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia,

Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022. doi:10.1016/j.physrep.2022.08.003.

[VC06] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Physical Review B*, 73:094423, March 2006. doi:10.1103/PhysRevB.73.094423.

[WBC22] Kianna Wan, Mario Berta, and Earl T. Campbell. Randomized quantum algorithm for statistical phase estimation. *Physical Review Letters*, 129:030503, July 2022. doi:10.1103/PhysRevLett.129.030503.

[WFRJ23] Guoming Wang, Daniel Stilck França, Gumaro Rendon, and Peter D. Johnson. Faster ground state energy estimation on early fault-tolerant quantum computers via rejection sampling, 2023. arXiv:2304.09827.

[WFZ+23] Guoming Wang, Daniel Stilck França, Ruizhe Zhang, Shuchen Zhu, and Peter D. Johnson. Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision. *Quantum*, 7:1167, November 2023. doi:10.22331/q-2023-11-06-1167.

[Wik24] Wikipedia contributors. Fourier transform — Wikipedia, the free encyclopedia, 2024. [Online; accessed 5-October-2024] https://en.wikipedia.org/w/index.php?title=Fourier_transform&oldid=1249353484.

[WW19] Nathan Wiebe and Leonard Wossnig. Generative training of quantum Boltzmann machines with hidden units, 2019. arXiv:1905.09902.

[ZLW21] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Variational quantum Boltzmann machines. *Quantum Machine Intelligence*, 3(1):7, 2021. doi:10.1007/s42484-020-00033-7.

APPENDIX

3.A Preliminaries

In this section, we introduce our notation and present some definitions and standard results from the optimization literature, which we use later in this document. Specifically, we revisit one of the known convergence results [KR20, Corollary 1] associated with the stochastic gradient descent algorithm.

3.A.1 Notation

Let $\mathbb{R}, \mathbb{R}_{\geq 0}$, $\mathbb{N}$, and $\mathbb{C}$ denote the set of real, non-negative real, natural, and complex numbers, respectively. We use the notation $[M]$ to denote the set $\{1, 2, \ldots, M\}$. Let $\mathcal{H}$ denote a 2^n-dimensional Hilbert space associated with a quantum system of n qubits. We denote the set of quantum states acting on $\mathcal{H}$ by $\mathcal{D}$. Let $\mathrm{Tr}[X]$ denote the trace of a matrix X, i.e., the sum of its diagonal elements. Also, let $X^\dagger$ denote the Hermitian conjugate (or adjoint) of the matrix X. The Schatten p-norm of a matrix X is defined for $p \in [1, \infty)$ as follows:

$$\|X\|_p := \left(\mathrm{Tr}\left[\left(X^\dagger X\right)^{\frac{p}{2}}\right]\right)^{\frac{1}{p}}. \tag{3.A.1}$$

For our purposes, we use Schatten norms with $p = 1$ (also called trace norm), $p = 2$ (Hilbert–Schmidt norm), and $p = \infty$ (operator norm). Note that the operator norm of a matrix corresponds to its maximum singular value. For notational convenience, we omit the subscript '∞' when referring to the operator norm. Additionally, $\|x\|$ denotes the ℓ_2 norm of a vector x. Moreover, let $\{X, Y\} := XY + YX$ denote the anti-commutator of the matrices X and Y. For a multivariate function $f\colon \mathbb{R}^n \to \mathbb{R}$, we use ∇f and $\nabla^2 f$ to denote

its gradient and Hessian, respectively. Let $\partial f(\cdot)/\partial x_i$ denote the partial derivative of f with respect to the ith component of the vector x. For brevity, we use the notation $\partial_i f(\cdot) \equiv \partial f(\cdot)/\partial x_i$ throughout the appendices. The (i, j)-th element of the Hessian is then denoted by $\partial_i \partial_j f(\cdot)$. Finally, we use the notation $O(\cdot)$ for hiding constants that do not depend on any problem parameter.

3.A.2 Definitions and Standard Results

We now review some definitions and known results related to the Lipschitz continuity and smoothness of a function.

Definition 21 (Lipschitz Continuity) *A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is L-Lipschitz continuous if there exists a non-negative real number L such that, for all $x, x' \in \mathbb{R}^n$, the following holds:*

$$\|f(x) - f(x')\| \leq L \|x - x'\| . \tag{3.A.2}$$

We say that L is a Lipschitz constant for f.

Definition 22 (Smoothness) *A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is ℓ-smooth if its gradient is ℓ-Lipschitz continuous. In other words, there exists a non-negative real number ℓ such that, for all $x, x' \in \mathbb{R}^n$, the following holds:*

$$\|\nabla f(x) - \nabla f(x')\| \leq \ell \|x - x'\| . \tag{3.A.3}$$

We now recall some known results related to the Lipschitz continuity of a function. These results also directly apply to the smoothness of a function because, as per the definition above, if the gradient of the function is Lipschitz continuous, then the function is

smooth. Having said that, we begin with a simple case in which the function $f\colon \mathbb{R} \to \mathbb{R}$ is univariate and differentiable. Now, in order to prove that this function is Lipschitz continuous, one approach is to show that it satisfies the condition given by (3.A.2) for some non-negative constant L. This gives the Lipschitz constant L for this function. Alternatively, we can show Lipschitz continuity by bounding the gradient of f from above. This is a direct consequence of the mean value theorem. More formally, if there exists a non-negative real number L such that the absolute value of its gradient is bounded from above by L, that is, the following holds for all $x \in \mathbb{R}$

$$\left|\frac{df(x)}{dx}\right| \leq L, \tag{3.A.4}$$

then L is a Lipschitz constant for f. Using this simple case of a univariate function as a base case, the following lemmas provide Lipschitz constants for multivariate and multivariate vector-valued functions.

Lemma 23 (Lipschitz Constant for a Multivariate Function) *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a differentiable multivariate function with bounded partial derivatives. Then the value*

$$L = \sqrt{n} \max_{i\in[n]} \left\{ \sup_x \left| \frac{\partial f(x)}{\partial x_i} \right| \right\} \tag{3.A.5}$$

is a Lipschitz constant for f.

Proof. See [PCW24, Appendix A.1]. ■

Lemma 24 (Lipschitz Constant for a Multivariate Vector-Valued Function) *Let $f\colon \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable multivariate vector-valued function such that each of its components, f_i, is L_i-Lipschitz continuous. Then*

$$L = \left(\sum_{i=1}^{m} L_i^2 \right)^{\frac{1}{2}} = n^{\frac{1}{2}} \left(\sum_{i=1}^{m} \max_{j\in[n]} \left\{ \sup_x \left| \frac{\partial f_i(x)}{\partial x_j} \right|^2 \right\} \right)^{\frac{1}{2}} \tag{3.A.6}$$

is a Lipschitz constant for f.

Proof. For the proof of the first equality of the lemma statement, see [PCW24, Appendix A.2]. The second equality then directly follows from Lemma 23. ■

The objective function that we deal with in this chapter is non-convex in general. In optimization theory, it is well known that finding a globally optimal point of a non-convex function is generally intractable (NP-hard) [DDG+22, Section 2.1]. Therefore, an important question arises about the types of solutions that can be guaranteed in such a scenario. In optimization theory, when dealing with non-convex objective functions, the notion of ε-stationary points is often considered. Intuitively, a point is ε-stationary if the norm of the gradient of the function at that point is very small. Formally, we define an ε-stationary point as follows:

Definition 25 (ε-Stationary Point) *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function, and let $\varepsilon \geq 0$. A point $x \in \mathbb{R}^n$ is an ε-stationary point of f if $\|\nabla f(x)\| \leq \varepsilon$.*

We conclude by recalling Hoeffding's inequality, which we employ later to examine the sample complexity of our algorithm.

Lemma 26 (Hoeffding's Inequality) *Suppose that $X_1, \ldots, X_n$ are n independent random variables, and that there exist $a_i, b_i \in \mathbb{R}$ such that $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Then, for all $\varepsilon \geq 0$, we have that*

$$\Pr\left(\left|\overline{X} - \mathbb{E}[\overline{X}]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2n^2\varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right), \tag{3.A.7}$$

where $\overline{X} := \frac{1}{n}\sum_{i=1}^n X_i$ and $\mathbb{E}[\overline{X}]$ is the expected value of $\overline{X}$.

3.A.3 Stochastic Gradient Descent

Consider the following minimization problem:

$$f^* := \inf_{x \in \mathbb{R}^n} f(x), \tag{3.A.8}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is an ℓ-smooth function and f^* is the global minimum. The stochastic gradient descent (SGD) algorithm uses the following rule to update the iterate:

$$x_{m+1} = x_m - \eta \overline{g}(x_m), \tag{3.A.9}$$

where $\overline{g}(x)$ is a stochastic gradient, evaluated at some point x, and $\eta > 0$ is the learning rate parameter. Furthermore, the SGD algorithm requires the stochastic gradient $\overline{g}(x)$ to be unbiased, i.e., $\mathbb{E}[\overline{g}(x)] = \nabla f(x)$, for all $x \in \mathbb{R}^n$. Here, the expectation $\mathbb{E}[\cdot]$ is with respect to the randomness inherent in $\overline{g}(x)$. In addition, for all $x \in \mathbb{R}^n$, $\overline{g}(x)$ should also satisfy the following condition: there exist constants $A, B, C \geq 0$ such that

$$\mathbb{E}\left[\|\overline{g}(x)\|^2\right] \leq 2A(f(x) - f^*) + B\|\nabla f(x)\|^2 + C. \tag{3.A.10}$$

To this end, the following lemma demonstrates the rate at which the SGD algorithm converges to an ε-stationary point of f. This lemma is a restatement of [KR20, Corollary 1], and we include it here for completeness.

Lemma 27 (SGD Convergence) *Let M be the total number of iterations of the SGD algorithm with update rule given by (3.A.9). Also, let $\eta := \min\left\{\frac{1}{\sqrt{\ell A M}}, \frac{1}{\ell B}, \frac{\varepsilon}{2\ell C}\right\}$ and $\Delta := f(x_0) - f^*$. Then provided that*

$$M \geq \frac{12\ell\Delta}{\varepsilon^2} \max\left\{B, \frac{12A\Delta}{\varepsilon^2}, \frac{2C}{\varepsilon^2}\right\}, \tag{3.A.11}$$

the SGD algorithm converges in such a way that

$$\min_{1\leq m\leq M} \mathbb{E}[\|\nabla f(x_m)\|] \leq \varepsilon, \tag{3.A.12}$$

where the expectation $\mathbb{E}[\cdot]$ *is over the randomness of the SGD algorithm.*

3.B Justification for Main Results

3.B.1 Organization

The rest of the appendices are organized as follows. In Section 3.B.2, we begin by formally defining the quantum Boltzmann machine (QBM) learning problem of ground-state energies (Definition 28). The goal of this learning problem is to find an ε-stationary point of a particular optimization problem that we define later in (3.B.9). To accomplish this, we employ SGD; for more details on SGD, refer to Section 3.A.3. Motivated by the fact that SGD is a first-order optimization algorithm, in Section 3.B.3, we derive an analytical expression for the gradient of the objective function and further show that this gradient is bounded. Then, in Section 3.B.4, we derive analytical expressions for the matrix elements of the Hessian of our objective function and show that these elements are also bounded. This property of the Hessian is important for establishing that our objective function is smooth, which we analyze in Section 3.B.5. Furthermore, this smoothness property then ensures convergence of SGD to an ε-stationary point. In Section 3.B.6, we present a quantum algorithm that estimates the gradient by returning an unbiased estimator of it. We refer to this algorithm as the quantum Boltzmann gradient estimator (QBGE) in what follows. Then, in Algorithm 4, we present the full SGD-based algorithm for QBM learning of ground-state energies, which employs QBGE for computing the stochastic gradient at

each iteration. We refer to this algorithm as QBM-GSE. Finally, in Section 3.B.8, we analyze the sample complexity of the QBM-GSE algorithm.

3.B.2 Problem Setup

Here we formally present the QBM learning problem that we introduced in the main text. Let $H \in \mathbb{C}^{2^n \times 2^n}$ be the Hamiltonian of interest, acting on the space of n qubits, and suppose that it is given in the following form:

$$H := \sum_{k=1}^{K} \alpha_k H_k, \tag{3.B.1}$$

where, for all $k \in [K]$, the coefficient $\alpha_k \in \mathbb{R}$ and H_k is a local Hamiltonian that acts on a constant number of qubits. Note that, for all $k \in [K]$, the parameter α_k in general can be any real number, but we can absorb any negative sign of α_k into H_k. Therefore, without loss of generality, we assume that $\alpha_k > 0$, for all $k \in [K]$. Furthermore, we assume that $\|H_k\| \leq 1$, for all $k \in [K]$. We do this without loss of generality by absorbing the norm of H_k into α_k, so that

$$\|H\| \leq \sum_{k=1}^{K} |\alpha_k| \, \|H_k\| \leq \sum_{k=1}^{K} |\alpha_k| =: \|\alpha\|_1 . \tag{3.B.2}$$

For our purposes, we assume that $\|\alpha\|_1$ is polynomial in n, i.e., $\|\alpha\|_1 = O(\text{poly}(n))$. We also assume that H can be measured efficiently on a quantum computer. This assumption is reasonable because physically relevant Hamiltonians consist of only $O(\text{poly}(n))$ summands in their linear combinations (see (3.B.1)), and thus they can be efficiently measured on a quantum computer. Having said that, we now state the above assumptions more formally below.

Assumption 1 *The Hamiltonian H, defined in (3.B.1), can be efficiently measured on a quantum computer, and $\|\alpha\|_1 = O(\text{poly}(n))$.*

We now present the problem of determining the ground-state energy of H. This problem can be formulated as an optimization problem as follows:

$$\inf_{\rho \in \mathcal{D}} \text{Tr}[H\rho], \tag{3.B.3}$$

where $\mathcal{D}$ represents the set of all possible quantum states acting on the same Hilbert space on which H acts. As previously mentioned in the main text, the problem above is generally difficult to solve due to the large search space of size $2^n \times 2^n$, which is exponential in the number of qubits. Despite this computational difficulty, various approaches have been proposed (see main text for references). These approaches typically involve reducing the search space by making an informed guess and then parameterizing this reduced search space. Following this reduction, these approaches utilize classical optimization techniques, such as gradient descent, to find the optimal value within this reduced search space.

Here, we parameterize this search space in the following way. Consider a parameterized QBM Hamiltonian

$$G(\theta) \coloneqq \sum_{j=1}^{J} \theta_j G_j, \tag{3.B.4}$$

where, for all $j \in [J]$, $\theta_j \in \mathbb{R}$ is a tunable parameter and G_j is a local Hamiltonian that acts on a constant number of qubits. Additionally, $\theta \coloneqq (\theta_1, \ldots, \theta_J)$ is a parameter vector. This parameterized Hamiltonian $G(\theta)$ further defines the following parameterized thermal state:

$$\rho(\theta) \coloneqq \frac{e^{-G(\theta)}}{Z(\theta)}, \tag{3.B.5}$$

where $Z(\theta) := \mathrm{Tr}[e^{-G(\theta)}]$ is the partition function. Consequently, using this parameterization, we can rewrite the original optimization problem, given by (3.B.3), in the following way:

$$\inf_{\theta \in \mathbb{R}^J} \mathrm{Tr}[H\rho(\theta)]. \tag{3.B.6}$$

Notably, the original optimization problem (defined in (3.B.3)) and the above parameterized problem are equivalent because any quantum state can be expressed as a thermal state with a suitable Hamiltonian. This means that the dimension J of the parameterized search space is exponential in the number of qubits since the dimension of the original search space (the set of quantum states, $\mathcal{D}$) is also exponential in the number of qubits. This is evident from a simple counting argument. To address this computational difficulty, we assume some knowledge about the problem structure, allowing us to reduce the parameterized search space such that J is $O(\mathrm{poly}(n))$.

Due to this reduction in the search space, the following inequality is a basic consequence of the variational principle:

$$\inf_{\rho \in \mathcal{D}} \mathrm{Tr}[H\rho] \leq \inf_{\theta \in \mathbb{R}^J} \mathrm{Tr}[H\rho(\theta)], \tag{3.B.7}$$

The above inequality indicates that the true ground-state energy of H is bounded from above by the minimal energy of the Hamiltonian H over every possible trial state $\rho(\theta)$.

Moreover, due to this reduction, the function $f(\theta)$ is now a non-convex function. Indeed, one can check that even the following basic instance of $f(\theta)$ is non-convex (refer to Figure 3.2):

$$(\theta_1, \theta_2) \mapsto \mathrm{Tr}\left[\sigma_Y \frac{e^{-G(\theta_1,\theta_2)}}{\mathrm{Tr}[e^{-G(\theta_1,\theta_2)}]}\right] \tag{3.B.8}$$

with $G(\theta_1, \theta_2) = \theta_1\sigma_X + \theta_2\sigma_Y$. As mentioned previously in Section 3.A.2, it is well known in optimization theory that finding a globally optimal point of a non-convex function is generally NP-hard [DDG+22, Section 2.1]. Consequently, optimizing $\mathrm{Tr}[H\rho(\theta)]$ to find

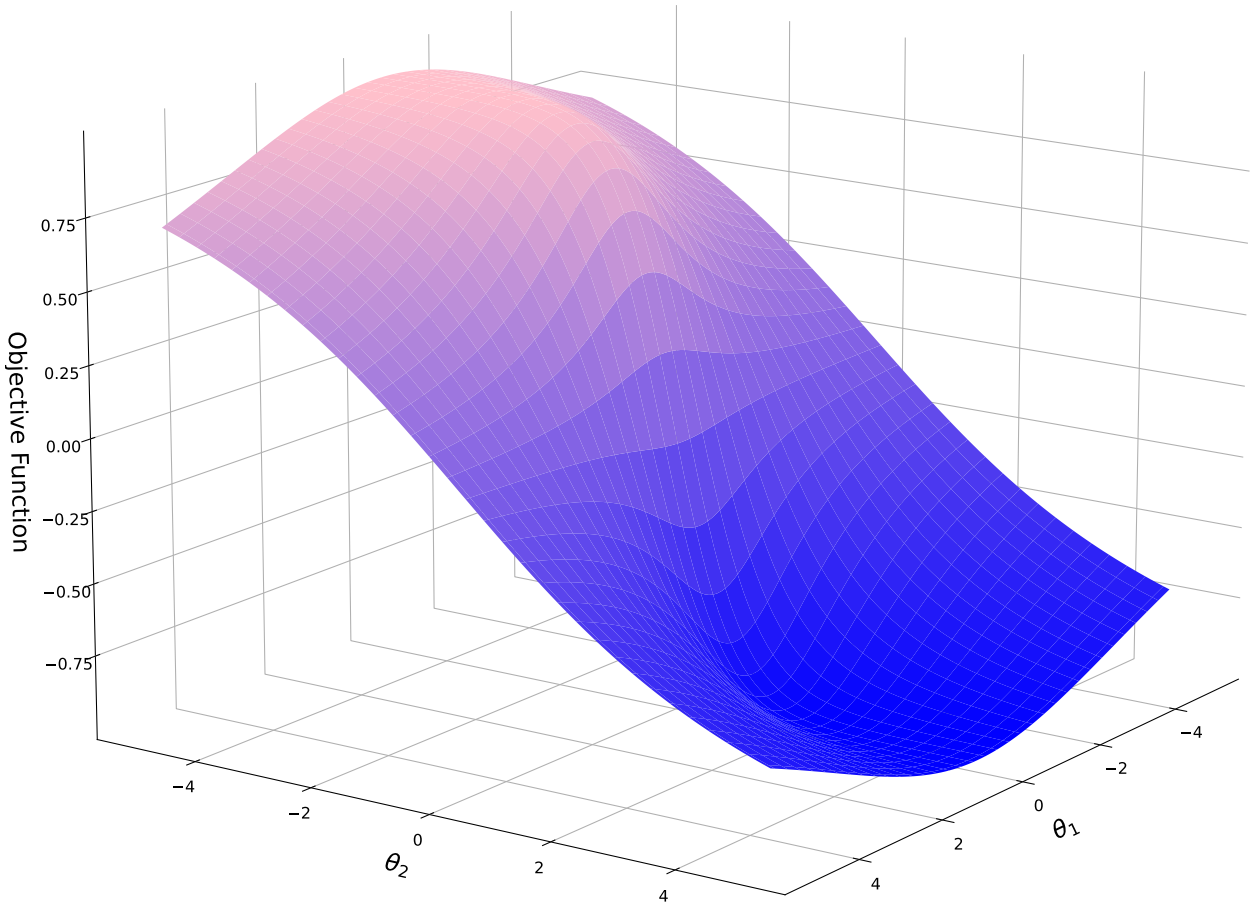


Figure 3.2: The non-convex landscape of the objective function given by (3.B.8).

its global optimal point is challenging. Therefore, an important question arises about the types of solutions that can be guaranteed in such a scenario. In optimization theory, when dealing with non-convex objective functions, the notion of ε-stationary points (Definition 25) is often considered. Given the difficulty of finding the globally optimal point, in this chapter, we focus instead on finding an ε-stationary point of $\mathrm{Tr}[H\rho(\theta)]$.

We measure the cost of our algorithm by the number of samples, N, of parameterized thermal states (as defined in (3.B.4)) required to find an ε-stationary point of $\mathrm{Tr}[H\rho(\theta)]$.

We refer to this metric as the sample complexity of our algorithm, which we analyze in more detail in Section 3.B.8.

With the above notions in place, we now formally define our problem as follows:

Definition 28 (QBM Learning of Ground-State Energy) *Let n be the number of qubits. Given a Hamiltonian $H \in \mathbb{C}^{2^n \times 2^n}$ as defined in (3.B.1) such that it satisfies Assumption 1 and a positive integer J such that $J \in O(\text{poly}(n))$, consider the following optimization problem:*

$$\inf_{\theta \in \mathbb{R}^J} \text{Tr}[H\rho(\theta)], \tag{3.B.9}$$

where $\rho(\theta)$ is a parameterized thermal state as defined in (3.B.4). Then the goal of QBM learning of ground-state energy is to find an ε-stationary point of the above optimization problem, given access to multiple copies of $\rho(\theta)$, for all $\theta \in \mathbb{R}^J$.

3.B.3 Gradient

We employ SGD (see Section 3.A.3 for more details on SGD) to find an ε-stationary point of $\text{Tr}[H\rho(\theta)]$ (refer to Definition 28 for a more formal definition of the problem). Therefore, our first step is to derive an analytical expression for the gradient, $\nabla_\theta \text{Tr}[H\rho(\theta)]$. Note that this gradient is a multivariate vector-valued function with partial derivatives as its components, i.e.,

$$\left\{ \frac{\partial \text{Tr}[H\rho(\theta)]}{\partial \theta_j} \right\}_{j=1}^{J}. \tag{3.B.10}$$

Recall that for brevity, we use the notation $\partial_j \equiv \frac{\partial}{\partial \theta_j}$. Since $\rho(\theta)$ is a thermal state, for all $j \in [J]$, its partial derivative, $\partial_j \rho(\theta)$, involves the partial derivative of the matrix exponential $e^{-G(\theta)}$:

$$\partial_j e^{-G(\theta)}. \tag{3.B.11}$$

Therefore, we first focus on deriving an explicit expression for the above quantity, which we do in the following lemma. Note that this derivation is not original, and it follows from the developments in [Has07], [AAKS21, Appendix B], and [CB24, Lemma 5] (see also [Kim12, Section III-C] and [KB19, Section IV-A]). We include it here for completeness.

Lemma 29 (Partial Derivatives of the Matrix Exponential [Has07, Kim12, KB19, AAKS21, CB24]) *Let $J \in \mathbb{N}$, and let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $G(\theta)$ be the corresponding parameterized QBM Hamiltonian as defined in (3.B.4). Then, the partial derivative of $e^{-G(\theta)}$ with respect to θ_j is given as follows:*

$$\partial_j e^{-G(\theta)} = -\frac{1}{2}\left\{\Phi_\theta(G_j), e^{-G(\theta)}\right\}, \tag{3.B.12}$$

where the quantum channel Φ_θ is defined as

$$\Phi_\theta(X) := \int_{\mathbb{R}} dt\; p(t)\, e^{-iG(\theta)t} X e^{iG(\theta)t}, \tag{3.B.13}$$

and $p(t)$ satisfies

$$\int_{\mathbb{R}} dt\; p(t) e^{-i\omega t} = \frac{\tanh \omega/2}{\omega/2}. \tag{3.B.14}$$

Proof. According to Duhamel's formula, the partial derivative of a matrix exponential $e^{A(x)}$ with respect to some parameter x is given as follows:

$$\partial_x e^{A(x)} = \int_0^1 e^{(1-u)A(x)} \left(\partial_x A(x)\right) e^{uA(x)}\, du. \tag{3.B.15}$$

Using this formula for $\partial_j e^{-G(\theta)}$, we obtain:

$$\partial_j e^{-G(\theta)} = \int_0^1 e^{(1-u)(-G(\theta))} \left(\partial_j(-G(\theta))\right) e^{u(-G(\theta))}\, du \tag{3.B.16}$$

$$= -\int_0^1 e^{(u-1)G(\theta)} G_j e^{-uG(\theta)}\, du. \tag{3.B.17}$$

Now, suppose that the spectral decomposition of $G(\theta)$ is as follows:

$$G(\theta) = \sum_k \lambda_k |k\rangle\langle k|, \tag{3.B.18}$$

where $\{\lambda_k\}_k$ are the eigenvalues and $\{|k\rangle\}_k$ are the corresponding eigenvectors. Substituting the above equation into (3.B.17), we find that

$$\begin{aligned}
\partial_j e^{-G(\theta)} &= -\int_0^1 \left(\sum_k e^{(u-1)\lambda_k} |k\rangle\langle k|\right) G_j \left(\sum_l e^{-u\lambda_l} |l\rangle\langle l|\right) du && (3.B.19)\\
&= -\int_0^1 \sum_{k,l} e^{(u-1)\lambda_k} |k\rangle\langle k| \left(G_j\right) e^{-u\lambda_l} |l\rangle\langle l| \, du && (3.B.20)\\
&= -\sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \left(\int_0^1 e^{(u-1)\lambda_k} e^{-u\lambda_l} \, du\right) && (3.B.21)\\
&= -\sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \left(e^{-\lambda_k} \int_0^1 e^{u(\lambda_k-\lambda_l)} \, du\right) && (3.B.22)\\
&= -\sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \left(e^{-\lambda_k} \frac{e^{\lambda_k-\lambda_l} - 1}{\lambda_k - \lambda_l}\right). && (3.B.23)
\end{aligned}$$

Now, consider the following:

$$e^{-\lambda_k} \frac{e^{\lambda_k-\lambda_l} - 1}{\lambda_k - \lambda_l} = e^{-\lambda_k} \frac{e^{\lambda_k-\lambda_l} - 1}{e^{\lambda_k-\lambda_l} + 1} \frac{e^{\lambda_k-\lambda_l} + 1}{\lambda_k - \lambda_l} = \frac{\tanh\left(\frac{\lambda_k-\lambda_l}{2}\right)}{\frac{\lambda_k-\lambda_l}{2}} \frac{e^{-\lambda_l} + e^{-\lambda_k}}{2}. \tag{3.B.24}$$

Let $p(t)$ be a function such that its Fourier transform is the following:

$$\int_{\mathbb{R}} dt \; p(t) e^{-i\omega t} = \frac{\tanh \omega/2}{\omega/2}. \tag{3.B.25}$$

Using this equation and (3.B.24), we rewrite (3.B.23) in the following way:

$$\begin{aligned}
&\partial_j e^{-G(\theta)} \\
&= -\sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \left(\left(\int_{\mathbb{R}} dt \; p(t) e^{-i(\lambda_k-\lambda_l)t}\right) \frac{e^{-\lambda_l} + e^{-\lambda_k}}{2}\right) && (3.B.26)\\
&= -\frac{1}{2} \sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \left(\int_{\mathbb{R}} dt \; p(t) \left(e^{-i\lambda_k t + i\lambda_l t - \lambda_l} + e^{-i\lambda_k t - \lambda_k + i\lambda_l t}\right)\right) && (3.B.27)\\
&= -\frac{1}{2} \left(\int_{\mathbb{R}} dt \; p(t) \left(\sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \; e^{-i\lambda_k t + i\lambda_l t - \lambda_l} + \sum_{k,l} |k\rangle\langle k| G_j |l\rangle\langle l| \; e^{-i\lambda_k t - \lambda_k + i\lambda_l t}\right)\right) && (3.B.28)
\end{aligned}$$

$$= -\frac{1}{2}\Bigg(\int_{\mathbb{R}} dt\ p(t) \Bigg(\sum_k e^{-i\lambda_k t} |k\rangle\langle k| G_j \sum_l e^{i\lambda_l t - \lambda_l} |l\rangle\langle l| + \sum_k e^{-i\lambda_k t - \lambda_k} |k\rangle\langle k| G_j \sum_l e^{i\lambda_l t} |l\rangle\langle l| \Bigg) \Bigg) \tag{3.B.29}$$

$$= -\frac{1}{2}\left(\int_{\mathbb{R}} dt\ p(t) \left(e^{-iG(\theta)t} G_j e^{iG(\theta)t} e^{-G(\theta)} + e^{-G(\theta)} e^{-iG(\theta)t} G_j e^{iG(\theta)t} \right) \right) \tag{3.B.30}$$

$$= -\frac{1}{2}\left(\left(\int_{\mathbb{R}} dt\ p(t) e^{-iG(\theta)t} G_j e^{iG(\theta)t} \right) e^{-G(\theta)} + e^{-G(\theta)} \left(\int_{\mathbb{R}} dt\ p(t) e^{-iG(\theta)t} G_j e^{iG(\theta)t} \right) \right) \tag{3.B.31}$$

$$= -\frac{1}{2}\left(\Phi_\theta(G_j) e^{-G(\theta)} + e^{-G(\theta)} \Phi_\theta(G_j) \right) \tag{3.B.32}$$

$$= -\frac{1}{2}\left\{ \Phi_\theta(G_j), e^{-G(\theta)} \right\}, \tag{3.B.33}$$

where, in the second last equality, we use the definition of the quantum channel Φ_θ introduced in the lemma statement (see (3.B.13)). ■

Remark 30 *The function $p(t)$ that satisfies (3.B.14) is the following "high-peak-tent" probability density:*

$$p(t) = \frac{2}{\pi} \ln \left| \coth\left(\frac{\pi t}{2} \right) \right|. \tag{3.B.34}$$

See Figure 3.3 for a plot of $p(t)$. For a proof of (3.B.34), refer to [AAKS21, Appendix B] or see Lemma 31 below. Furthermore, it is important to note that $p(t)$ is indeed a probability density function because $p(t) > 0$, for all $t \in \mathbb{R}$, and

$$\int_{\mathbb{R}} p(t)\, dt = 1. \tag{3.B.35}$$

The inequality $p(t) > 0$ follows because $|\coth(x)| > 1$ for all $x \in \mathbb{R}$ and (3.B.35) by plugging in $\omega = 0$ in (3.B.14) and noting that $\lim_{\omega \to 0} \frac{\tanh \omega/2}{\omega/2} = 1$.

Lemma 31 (Fourier transform of high-peak-tent probability density) *The following equality holds:*

$$\int_{\mathbb{R}} dt\ \frac{2}{\pi} \ln \left| \coth\left(\frac{\pi t}{2} \right) \right| e^{-i\omega t} = \frac{\tanh\left(\frac{\omega}{2} \right)}{\frac{\omega}{2}}. \tag{3.B.36}$$

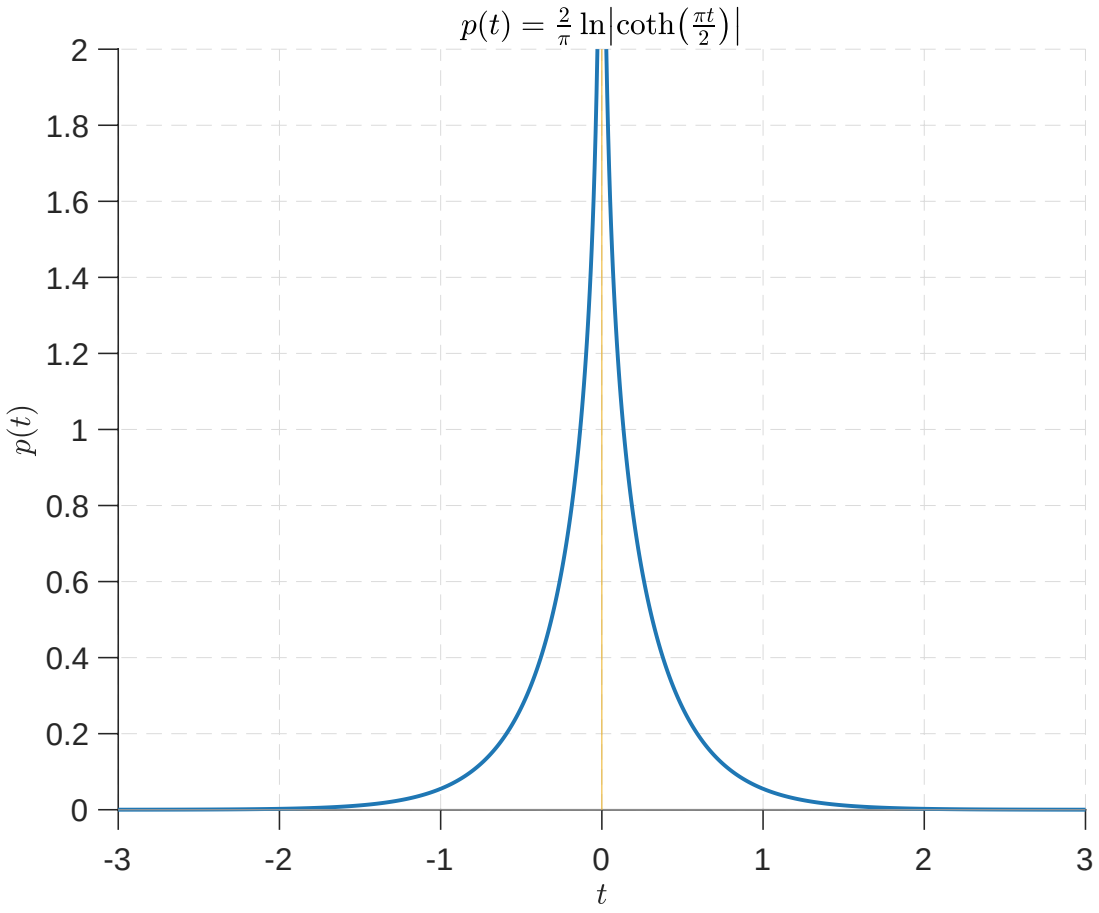


Figure 3.3: The high-peak-tent probability density function $p(t)$, defined in (3.B.34).

Proof. We provide two different proofs of this lemma. To begin with, let us define the Fourier operator $\mathcal{F}$, as applied to a function $f(t)$, as

$$\mathcal{F}\{f(t)\} := \int_{\mathbb{R}} dt\, f(t) e^{-i\omega t} =: F(\omega). \tag{3.B.37}$$

As such, our aim is to prove that

$$\mathcal{F}\{p(t)\} = \frac{\tanh\left(\frac{\omega}{2}\right)}{\frac{\omega}{2}}, \tag{3.B.38}$$

where

$$p(t) = \frac{2}{\pi} \ln \left| \coth\left(\frac{\pi t}{2}\right) \right|. \tag{3.B.39}$$

Consider that the derivative of $p(t)$ is as follows:

$$\frac{d}{dt} p(t) = \frac{d}{dt} \frac{2}{\pi} \ln \left| \coth\left(\frac{\pi t}{2}\right) \right| \tag{3.B.40}$$

$$= \frac{2}{\pi} \frac{1}{\coth\left(\frac{\pi t}{2}\right)} \frac{d}{dt} \left[\coth\left(\frac{\pi t}{2}\right) \right] \tag{3.B.41}$$

$$= \frac{2}{\pi} \frac{1}{\coth\left(\frac{\pi t}{2}\right)} \left[-\operatorname{csch}^2\left(\frac{\pi t}{2}\right) \right] \frac{\pi}{2} \tag{3.B.42}$$

$$= -\frac{2}{\sinh(\pi t)}. \tag{3.B.43}$$

Then, by the fundamental theorem of calculus, and the fact that $\lim_{t \to -\infty} p(t) = 0$, we find that

$$p(t) = -2 \int_{-\infty}^{t} d\tau \, \frac{1}{\sinh(\pi \tau)}. \tag{3.B.44}$$

It is then a well known fact that the following functions are Fourier transform pairs (see [Wik24]):

$$\int_{-\infty}^{t} d\tau \, f(\tau) \qquad \leftrightarrow \qquad \frac{F(\omega)}{i\omega} + 2\pi F(0) \delta(\omega), \tag{3.B.45}$$

where $F(\omega)$ is the Fourier transform of $f(t)$. Given that the following are also known Fourier transform pairs

$$\frac{1}{\sinh(\pi t)} \qquad \leftrightarrow \qquad -i \tanh\left(\frac{\omega}{2}\right), \tag{3.B.46}$$

and given that $\tanh(0) = 0$ and thus that $\tanh(0)\,\delta(\omega) = 0$, we conclude from (3.B.44), (3.B.45), and (3.B.46) that

$$\mathcal{F}\{p(t)\} = -2\mathcal{F}\left\{ \int_{-\infty}^{t} d\tau \, \frac{1}{\sinh(\pi \tau)} \right\} \tag{3.B.47}$$

$$= 2 \left[\frac{i \tanh\left(\frac{\omega}{2}\right)}{i\omega} + \pi i \tanh(0)\, \delta(\omega) \right] \tag{3.B.48}$$

$$= \frac{\tanh\left(\frac{\omega}{2}\right)}{\frac{\omega}{2}}, \tag{3.B.49}$$

as claimed.

Our alternate proof of Lemma 31 is as follows. Let us define the inverse Fourier operator $\mathcal{F}^{-1}$, as applied to a function $F(\omega)$ (defined in (3.B.37)), as

$$\mathcal{F}^{-1}\{F(\omega)\} \coloneqq \frac{1}{2\pi}\int_{\mathbb{R}} d\omega\, F(\omega)e^{i\omega t} = f(t). \tag{3.B.50}$$

Also, let the operator $f * g$ denote the convolution between two functions f and g:

$$(f * g)(t) \coloneqq \int_{\mathbb{R}} d\tau\, f(\tau)g(t-\tau) = f(t) * g(t). \tag{3.B.51}$$

Then from the convolution theorem for inverse Fourier transform, it follows that:

$$\mathcal{F}^{-1}\left(\frac{\tanh\left(\frac{\omega}{2}\right)}{\frac{\omega}{2}}\right) = \mathcal{F}^{-1}\left(\tanh\left(\frac{\omega}{2}\right)\right) * \mathcal{F}^{-1}\left(\frac{2}{\omega}\right) \tag{3.B.52}$$

$$= \frac{-i}{\sinh(\pi t)} * (-i\,\mathrm{sgn}(t)) \tag{3.B.53}$$

$$= \int_{\mathbb{R}} d\tau\, \frac{-i}{\sinh(\pi\tau)} \cdot (-i)\,\mathrm{sgn}(t-\tau) \tag{3.B.54}$$

$$= -\int_{\mathbb{R}} d\tau\, \frac{1}{\sinh(\pi\tau)} \cdot \mathrm{sgn}(t-\tau) \tag{3.B.55}$$

$$= -\left(\int_{-\infty}^{t} d\tau\, \frac{1}{\sinh(\pi\tau)} \cdot (1) + \int_{t}^{\infty} d\tau\, \frac{1}{\sinh(\pi\tau)} \cdot (-1)\right) \tag{3.B.56}$$

$$= -\int_{-\infty}^{t} d\tau\, \frac{1}{\sinh(\pi\tau)} + \int_{t}^{\infty} d\tau\, \frac{1}{\sinh(\pi\tau)} \tag{3.B.57}$$

$$= -\int_{-\infty}^{t} d\tau\, \frac{1}{\sinh(\pi\tau)} + \underbrace{\int_{-\infty}^{\infty} d\tau\, \frac{1}{\sinh(\pi\tau)}}_{=0} - \int_{-\infty}^{t} d\tau\, \frac{1}{\sinh(\pi\tau)} \tag{3.B.58}$$

$$= \int_{-\infty}^{t} d\tau\, \frac{-2}{\sinh(\pi\tau)} \tag{3.B.59}$$

$$= \frac{2}{\pi}\ln\left|\coth\left(\frac{\pi t}{2}\right)\right|, \tag{3.B.60}$$

where the last equality follows from (3.B.44). This concludes the alternate proof. ■

Using the development in Lemma 29 and Remark 30, we now direct our focus on deriving an analytical expression for the partial derivatives, that is, $\partial_j \mathrm{Tr}[H\rho(\theta)]$, for all $j \in [J]$.

Proposition 32 (Partial Derivatives) *Let H be a Hamiltonian as defined in (3.B.1). Let $J \in \mathbb{N}$, let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $\rho(\theta)$ be the corresponding parameterized thermal state as defined in (3.B.5). Then the partial derivative of the function $\mathrm{Tr}[H\rho(\theta)]$ with respect to the parameter θ_j can be expressed as follows:*

$$\partial_j \mathrm{Tr}[H\rho(\theta)] = -\frac{1}{2}\left\langle\left\{H, \Phi_\theta(G_j)\right\}\right\rangle_{\rho(\theta)} + \langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)} \tag{3.B.61}$$

$$= -\frac{1}{2}\mathrm{Tr}\left[\left\{H - \langle H\rangle_{\rho(\theta)}, \Phi_\theta(G_j) - \left\langle G_j\right\rangle_{\rho(\theta)}\right\}\rho(\theta)\right], \tag{3.B.62}$$

where the quantum channel Φ_θ is defined in (3.B.13) and $\langle C\rangle_\sigma \equiv \mathrm{Tr}[C\sigma]$.

Proof. Consider that

$$\partial_j \mathrm{Tr}[H\rho(\theta)] = \mathrm{Tr}[H\left(\partial_j\rho(\theta)\right)] = \mathrm{Tr}\left[H\partial_j\left(\frac{e^{-G(\theta)}}{Z(\theta)}\right)\right] \tag{3.B.63}$$

$$= \mathrm{Tr}\left[H\left(-\frac{e^{-G(\theta)}}{Z(\theta)^2}\partial_j Z(\theta) + \left(\partial_j e^{-G(\theta)}\right)\frac{1}{Z(\theta)}\right)\right], \tag{3.B.64}$$

where the last equality follows from the product rule. Now, to evaluate $\partial_j Z(\theta)$, consider the following:

$$\partial_j Z(\theta) = \partial_j \mathrm{Tr}\left[e^{-G(\theta)}\right] = \mathrm{Tr}\left[\partial_j e^{-G(\theta)}\right] \tag{3.B.65}$$

$$= -\frac{1}{2}\mathrm{Tr}\left[\left\{\Phi_\theta(G_j), e^{-G(\theta)}\right\}\right] = -\mathrm{Tr}\left[\Phi_\theta(G_j)e^{-G(\theta)}\right] \tag{3.B.66}$$

$$= -\mathrm{Tr}\left[\left(\int_{\mathbb{R}} dt\ p(t)e^{-iG(\theta)t}G_j e^{iG(\theta)t}\right)e^{-G(\theta)}\right] \tag{3.B.67}$$

$$= -\int_{\mathbb{R}} dt\ p(t)\,\mathrm{Tr}\left[e^{-iG(\theta)t}G_j e^{iG(\theta)t}e^{-G(\theta)}\right] \tag{3.B.68}$$

$$= -\int_{\mathbb{R}} dt\ p(t)\,\mathrm{Tr}\left[e^{-iG(\theta)t}G_j e^{-G(\theta)}e^{iG(\theta)t}\right] \tag{3.B.69}$$

$$= -\int_{\mathbb{R}} dt\ p(t)\,\mathrm{Tr}\left[e^{iG(\theta)t}e^{-iG(\theta)t}G_j e^{-G(\theta)}\right] \tag{3.B.70}$$

$$= -\int_{\mathbb{R}} dt\ p(t)\,\mathrm{Tr}\left[G_j e^{-G(\theta)}\right] \tag{3.B.71}$$

$$= -\mathrm{Tr}\left[G_j e^{-G(\theta)}\right]\int_{\mathbb{R}} dt\ p(t) \tag{3.B.72}$$

$$= -\operatorname{Tr}\left[G_j e^{-G(\theta)}\right]. \tag{3.B.73}$$

The third equality follows from Lemma 29. The seventh equality follows from the fact that $e^{-G(\theta)}$ commutes with $e^{-iG(\theta)t}$, and the eighth equality follows from the cyclicity property of trace. Finally, the penultimate equality follows from the fact that $p(t)$ is a probability density function (see (3.B.35)).

Now, substituting the last equality into (3.B.64) and using Lemma 29 for the second term of (3.B.64), we obtain:

$$\partial_j \operatorname{Tr}[H\rho(\theta)] = \operatorname{Tr}\left[H\left(-\frac{e^{-G(\theta)}}{Z(\theta)^2}\left(-\operatorname{Tr}[G_j e^{-G(\theta)}]\right) - \frac{1}{2}\left\{\Phi_\theta(G_j), e^{-G(\theta)}\right\}\frac{1}{Z(\theta)}\right)\right] \tag{3.B.74}$$

$$= \operatorname{Tr}\left[H\left(\frac{e^{-G(\theta)}}{Z(\theta)^2}\operatorname{Tr}[G_j e^{-G(\theta)}] - \frac{1}{2}\left\{\Phi_\theta(G_j), e^{-G(\theta)}\right\}\frac{1}{Z(\theta)}\right)\right] \tag{3.B.75}$$

$$= \operatorname{Tr}\left[H\frac{e^{-G(\theta)}}{Z(\theta)}\right]\operatorname{Tr}\left[G_j\frac{e^{-G(\theta)}}{Z(\theta)}\right] - \frac{1}{2}\operatorname{Tr}\left[H\left\{\Phi_\theta(G_j), \frac{e^{-G(\theta)}}{Z(\theta)}\right\}\right] \tag{3.B.76}$$

$$= -\frac{1}{2}\operatorname{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)] + \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}[G_j\rho(\theta)] \tag{3.B.77}$$

$$= -\frac{1}{2}\operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\right] + \langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)} \tag{3.B.78}$$

$$= -\frac{1}{2}\left\langle\left\{H, \Phi_\theta(G_j)\right\}\right\rangle_{\rho(\theta)} + \langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}. \tag{3.B.79}$$

This concludes the proof of the first equality claimed in the statement of the proposition.

Now, to show that the second equality in (3.B.62) also holds, consider the following steps:

$$\begin{aligned}
&\frac{1}{2}\operatorname{Tr}\left[\left\{H - \langle H\rangle_{\rho(\theta)}, \Phi_\theta(G_j) - \left\langle G_j\right\rangle_{\rho(\theta)}\right\}\rho(\theta)\right] \\
&= \frac{1}{2}\operatorname{Tr}\left[\left(H - \langle H\rangle_{\rho(\theta)}\right)\left(\Phi_\theta(G_j) - \left\langle G_j\right\rangle_{\rho(\theta)}\right)\rho(\theta)\right] \\
&\quad + \frac{1}{2}\operatorname{Tr}\left[\left(\Phi_\theta(G_j) - \left\langle G_j\right\rangle_{\rho(\theta)}\right)\left(H - \langle H\rangle_{\rho(\theta)}\right)\rho(\theta)\right] \\
&= \frac{1}{2}\operatorname{Tr}\left[H\Phi_\theta(G_j)\rho(\theta)\right] - \frac{1}{2}\operatorname{Tr}\left[H\left\langle G_j\right\rangle_{\rho(\theta)}\rho(\theta)\right] - \frac{1}{2}\operatorname{Tr}\left[\langle H\rangle_{\rho(\theta)}\Phi_\theta(G_j)\rho(\theta)\right] \\
&\quad + \frac{1}{2}\operatorname{Tr}\left[\langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}\rho(\theta)\right] + \frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)H\rho(\theta)\right] - \frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\langle H\rangle_{\rho(\theta)}\rho(\theta)\right]
\end{aligned} \tag{3.B.80}$$

$$- \frac{1}{2}\operatorname{Tr}\Big[\big\langle G_j\big\rangle_{\rho(\theta)} H\rho(\theta)\Big] + \frac{1}{2}\operatorname{Tr}\Big[\big\langle G_j\big\rangle_{\rho(\theta)} \langle H\rangle_{\rho(\theta)}\rho(\theta)\Big] \tag{3.B.81}$$

$$\begin{aligned}
&= \frac{1}{2}\operatorname{Tr}\Big[H\Phi_\theta(G_j)\rho(\theta)\Big] - \frac{1}{2}\langle H\rangle_{\rho(\theta)}\big\langle G_j\big\rangle_{\rho(\theta)} - \frac{1}{2}\langle H\rangle_{\rho(\theta)}\operatorname{Tr}\Big[\Phi_\theta(G_j)\rho(\theta)\Big] \\
&\quad + \frac{1}{2}\langle H\rangle_{\rho(\theta)}\big\langle G_j\big\rangle_{\rho(\theta)} + \frac{1}{2}\operatorname{Tr}\Big[\Phi_\theta(G_j)H\rho(\theta)\Big] - \frac{1}{2}\langle H\rangle_{\rho(\theta)}\operatorname{Tr}\Big[\Phi_\theta(G_j)\rho(\theta)\Big] \\
&\quad - \frac{1}{2}\big\langle G_j\big\rangle_{\rho(\theta)}\langle H\rangle_{\rho(\theta)} + \frac{1}{2}\big\langle G_j\big\rangle_{\rho(\theta)}\langle H\rangle_{\rho(\theta)}
\end{aligned} \tag{3.B.82}$$

$$= \frac{1}{2}\operatorname{Tr}\Big[\big\{H, \Phi_\theta(G_j)\big\}\rho(\theta)\Big] - \langle H\rangle_{\rho(\theta)}\big\langle G_j\big\rangle_{\rho(\theta)} \tag{3.B.83}$$

$$= \frac{1}{2}\big\langle\big\{H, \Phi_\theta(G_j)\big\}\big\rangle_{\rho(\theta)} - \langle H\rangle_{\rho(\theta)}\big\langle G_j\big\rangle_{\rho(\theta)}. \tag{3.B.84}$$

By multiplying both sides of the given equation by -1, we arrive at the second equality in (3.B.62). ■

Remark 33 *We employ the first equality stated in Proposition 32, i.e., (3.B.61), to propose a quantum algorithm that computes an unbiased estimator of the gradient. The purpose of stating the second equality is to illustrate that it resembles an entry in a covariance matrix.*

Having said that, we now show that for all $j \in [J]$, the absolute value of the partial derivative, i.e., $\big|\partial_j \operatorname{Tr}[H\rho(\theta)]\big|$, is bounded from above by a quantity that does not depend on the parameter vector θ. This demonstrates that the gradient is not arbitrarily large at any given point.

Proposition 34 (Bounds on Partial Derivatives) *Let H be a Hamiltonian as defined in (3.B.1). Let $J \in \mathbb{N}$, let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $\rho(\theta)$ be the corresponding parameterized thermal state as defined in (3.B.5). Then, for all $j \in [J]$, the following holds:*

$$\big|\partial_j \operatorname{Tr}[H\rho(\theta)]\big| \leq 2\,\|H\|\,\big\|G_j\big\|. \tag{3.B.85}$$

Proof. Consider that

$$\big|\partial_j \operatorname{Tr}[H\rho(\theta)]\big| = \Big|-\frac{1}{2}\operatorname{Tr}[\big\{H, \Phi_\theta(G_j)\big\}\rho(\theta)] + \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}[G_j\rho(\theta)]\Big| \tag{3.B.86}$$

$$\leq \frac{1}{2}\left|\mathrm{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)]\right| + |\mathrm{Tr}[H\rho(\theta)]|\left|\mathrm{Tr}[G_j\rho(\theta)]\right| \tag{3.B.87}$$

$$\leq \frac{1}{2}\left\|\left\{H, \Phi_\theta(G_j)\right\}\right\| + \|H\|\left\|G_j\right\| \tag{3.B.88}$$

$$\leq \|H\|\left\|G_j\right\| + \|H\|\left\|G_j\right\| \tag{3.B.89}$$

$$= 2\|H\|\left\|G_j\right\|. \tag{3.B.90}$$

The first equality follows from Proposition 32. The first inequality follows from the triangle inequality, and the second inequality follows from Hölder's inequality. The third inequality is a result of the anticommutator bound, which states that for any two matrices A and B, we have $\|\{A, B\}\| \leq 2\|A\|\|B\|$. Moreover, we also employed contractivity under a mixture-of-unitaries channel:

$$\|\Phi_\theta(X)\| = \left\|\int_{-\infty}^{\infty} dt\ p(t)e^{-iG(\theta)t}Xe^{iG(\theta)t}\right\| \tag{3.B.91}$$

$$\leq \int_{\mathbb{R}} dt\ p(t)\left\|e^{-iG(\theta)t}\right\|\|X\|\left\|e^{iG(\theta)t}\right\| \tag{3.B.92}$$

$$= \|X\|. \tag{3.B.93}$$

This concludes the proof. ■

3.B.4 Hessian

In this section, we focus on the matrix elements of the Hessian of the objective function $\mathrm{Tr}[H\rho(\theta)]$. We start by obtaining analytical expressions for these elements. Then, we demonstrate that these elements are bounded from above, ensuring that none of them can grow arbitrarily large at any given point. This property is crucial for establishing the smoothness of the objective function, which we will utilize later in our analysis.

Proposition 35 (Hessian) *Let H be a Hamiltonian as defined in (3.B.1). Let $J \in \mathbb{N}$, let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $\rho(\theta)$ be the corresponding parameterized thermal state as defined in (3.B.5). Then the matrix elements of the Hessian of $\mathrm{Tr}[H\rho(\theta)]$ are given by*

$$\begin{aligned}\partial_k \partial_j \mathrm{Tr}[H\rho(\theta)] = &-\frac{1}{2}\mathrm{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\rho(\theta)] + \frac{1}{4}\mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\{\rho(\theta), \Phi_\theta(G_k)\}\right] \\ &- \frac{1}{2}\mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\right]\langle G_k\rangle_{\rho(\theta)} - \frac{1}{2}\mathrm{Tr}\left[\{H, \Phi_\theta(G_k)\}\rho(\theta)\right]\left\langle G_j\right\rangle_{\rho(\theta)} \\ &- \frac{1}{2}\mathrm{Tr}\left[\left\{G_j, \Phi_\theta(G_k)\right\}\rho(\theta)\right]\langle H\rangle_{\rho(\theta)} + 2\langle H\rangle_{\rho(\theta)}\langle G_k\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}, \end{aligned} \tag{3.B.94}$$

where

$$\begin{aligned}\partial_k \Phi_\theta(G_j) = \int_{\mathbb{R}} dt \int_0^1 du\, it\, p(t) \times \\ \left(e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t} G_j e^{iG(\theta)t} - e^{-iG(\theta)t} G_j e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t}\right),\end{aligned} \tag{3.B.95}$$

with $p(t)$ the probability density function defined in (3.B.35).

Proof. From Proposition 32, we have that

$$\partial_k \partial_j \mathrm{Tr}[H\rho(\theta)] = \partial_k\left(-\frac{1}{2}\mathrm{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)] + \mathrm{Tr}[H\rho(\theta)]\,\mathrm{Tr}[G_j\rho(\theta)]\right), \tag{3.B.96}$$

Now, for the first term in the above equation, consider the following:

$$\begin{aligned}&\partial_k \mathrm{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)] \\ &= \mathrm{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\rho(\theta)] + \mathrm{Tr}[\left\{H, \Phi_\theta(G_j)\right\}\partial_k\rho(\theta)] &\text{(3.B.97)} \\ &= \mathrm{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\rho(\theta)] \\ &\quad + \mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\left(-\frac{1}{2}\{\Phi_\theta(G_k), \rho(\theta)\} + \rho(\theta)\langle G_k\rangle_{\rho(\theta)}\right)\right] &\text{(3.B.98)} \\ &= \mathrm{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\rho(\theta)] - \frac{1}{2}\mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\{\Phi_\theta(G_k), \rho(\theta)\}\right] \\ &\quad + \mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\langle G_k\rangle_{\rho(\theta)}\right] &\text{(3.B.99)} \\ &= \mathrm{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\rho(\theta)] - \frac{1}{2}\mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\{\rho(\theta), \Phi_\theta(G_k)\}\right]\end{aligned}$$

$$+ \operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\right]\langle G_k\rangle_{\rho(\theta)}\,, \tag{3.B.100}$$

where we again used Proposition 32 in the second equality. Now consider that

$$\partial_k \Phi_\theta(G_j) = \partial_k \int_{\mathbb{R}} dt\; p(t) e^{-iG(\theta)t} G_j e^{iG(\theta)t} \tag{3.B.101}$$

$$= \int_{\mathbb{R}} dt\; p(t)\left(\left[\partial_k e^{-iG(\theta)t}\right] G_j e^{iG(\theta)t} + e^{-iG(\theta)t} G_j \left[\partial_k e^{iG(\theta)t}\right]\right) \tag{3.B.102}$$

By applying Duhamel's formula, given by (3.B.15), we find that

$$\partial_k e^{iG(\theta)t} = it \int_0^1 du\; e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t}, \tag{3.B.103}$$

$$\partial_k e^{-iG(\theta)t} = -it \int_0^1 du\; e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t}, \tag{3.B.104}$$

so that

$$\partial_k \Phi_\theta(G_j) = \int_{\mathbb{R}} dt\; p(t) \begin{pmatrix} \left(it\int_0^1 du\; e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t}\right) G_j e^{iG(\theta)t} \\ + e^{-iG(\theta)t} G_j \left(-it\int_0^1 du\; e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t}\right) \end{pmatrix} \tag{3.B.105}$$

$$= \int_{\mathbb{R}} dt \int_0^1 du\; it\; p(t) \begin{pmatrix} e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t} G_j e^{iG(\theta)t} \\ - e^{-iG(\theta)t} G_j e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t} \end{pmatrix}. \tag{3.B.106}$$

Finally, for the second term in (3.B.96), consider that

$$\partial_k \left(\operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}[G_j\rho(\theta)]\right)$$

$$= \operatorname{Tr}[H\,(\partial_k\rho(\theta))]\operatorname{Tr}[G_j\rho(\theta)] + \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}[G_j\,(\partial_k\rho(\theta))] \tag{3.B.107}$$

$$= \operatorname{Tr}\left[H\left(-\frac{1}{2}\left\{\Phi_\theta(G_k), \rho(\theta)\right\} + \rho(\theta)\langle G_k\rangle_{\rho(\theta)}\right)\right]\operatorname{Tr}[G_j\rho(\theta)]$$
$$+ \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}\left[G_j\left(-\frac{1}{2}\left\{\Phi_\theta(G_k), \rho(\theta)\right\} + \rho(\theta)\langle G_k\rangle_{\rho(\theta)}\right)\right] \tag{3.B.108}$$

$$= -\frac{1}{2}\operatorname{Tr}\left[H\left\{\Phi_\theta(G_k), \rho(\theta)\right\}\right]\operatorname{Tr}[G_j\rho(\theta)] + \operatorname{Tr}[H\rho(\theta)]\langle G_k\rangle_{\rho(\theta)}\operatorname{Tr}[G_j\rho(\theta)]$$
$$- \frac{1}{2}\operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}\left[G_j\left\{\Phi_\theta(G_k), \rho(\theta)\right\}\right] + \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}\left[G_j\rho(\theta)\right]\langle G_k\rangle_{\rho(\theta)} \tag{3.B.109}$$

$$= -\frac{1}{2}\operatorname{Tr}\left[\left\{H, \Phi_\theta(G_k)\right\}\rho(\theta)\right]\left\langle G_j\right\rangle_{\rho(\theta)} + 2\langle H\rangle_{\rho(\theta)}\langle G_k\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}$$

$$- \frac{1}{2} \langle H \rangle_{\rho(\theta)} \operatorname{Tr}\left[\left\{G_j, \Phi_\theta(G_k)\right\} \rho(\theta)\right]. \tag{3.B.110}$$

Then, we finally see that

$$\begin{aligned}
\partial_k \partial_j \operatorname{Tr}[H\rho(\theta)] = &-\frac{1}{2} \operatorname{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\} \rho(\theta)] + \frac{1}{4} \operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\} \{\rho(\theta), \Phi_\theta(G_k)\}\right] \\
&- \frac{1}{2} \operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\} \rho(\theta)\right] \langle G_k \rangle_{\rho(\theta)} - \frac{1}{2} \operatorname{Tr}\left[\{H, \Phi_\theta(G_k)\} \rho(\theta)\right] \left\langle G_j \right\rangle_{\rho(\theta)} \\
&- \frac{1}{2} \operatorname{Tr}\left[\left\{G_j, \Phi_\theta(G_k)\right\} \rho(\theta)\right] \langle H \rangle_{\rho(\theta)} + 2 \langle H \rangle_{\rho(\theta)} \langle G_k \rangle_{\rho(\theta)} \left\langle G_j \right\rangle_{\rho(\theta)},
\end{aligned} \tag{3.B.111}$$

thus concluding the proof. ■

Proposition 36 (Bounds on the Hessian Elements) *Let H be a Hamiltonian as defined in (3.B.1). Let $J \in \mathbb{N}$, let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $\rho(\theta)$ be the corresponding parameterized thermal state as defined in (3.B.5). Then, for all $j, k \in [J]$, the following holds:*

$$\left|\partial_k \partial_j \operatorname{Tr}[H\rho(\theta)]\right| \leq 8 \|H\| \left\|G_j\right\| \|G_k\|. \tag{3.B.112}$$

Proof. Consider that

$$\begin{aligned}
&\left|\partial_k \partial_j \operatorname{Tr}[H\rho(\theta)]\right| \\
&\leq \frac{1}{2} \left|\operatorname{Tr}[\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\} \rho(\theta)]\right| \\
&\quad + \frac{1}{4} \left|\operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\} \{\rho(\theta), \Phi_\theta(G_k)\}\right]\right| + \frac{1}{2} \left|\operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\} \rho(\theta)\right] \langle G_k \rangle_{\rho(\theta)}\right| \\
&\quad + \frac{1}{2} \left|\operatorname{Tr}\left[\{H, \Phi_\theta(G_k)\} \rho(\theta)\right] \left\langle G_j \right\rangle_{\rho(\theta)}\right| + \frac{1}{2} \left|\operatorname{Tr}\left[\left\{G_j, \Phi_\theta(G_k)\right\} \rho(\theta)\right] \langle H \rangle_{\rho(\theta)}\right| \\
&\quad + 2 \left|\langle H \rangle_{\rho(\theta)}\right| \left|\langle G_k \rangle_{\rho(\theta)}\right| \left|\left\langle G_j \right\rangle_{\rho(\theta)}\right| \qquad (3.B.113) \\
&\leq \frac{1}{2} \left\|\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\right\| + \frac{1}{2} \left\|\left\{H, \Phi_\theta(G_j)\right\} \Phi_\theta(G_k)\right\| + \frac{1}{2} \left\|\left\{H, \Phi_\theta(G_j)\right\}\right\| \|G_k\| \\
&\quad + \frac{1}{2} \|\{H, \Phi_\theta(G_k)\}\| \left\|G_j\right\| + \frac{1}{2} \left\|\left\{G_j, \Phi_\theta(G_k)\right\}\right\| \|H\| + 2 \|H\| \|G_k\| \left\|G_j\right\| \qquad (3.B.114) \\
&\leq \frac{1}{2} \left\|\left\{H, \left[\partial_k \Phi_\theta(G_j)\right]\right\}\right\| + \|H\| \left\|G_j\right\| \|G_k\| + \|H\| \left\|G_j\right\| \|G_k\|
\end{aligned}$$

$$+ \|H\| \|G_k\| \left\|G_j\right\| + \left\|G_j\right\| \|G_k\| \|H\| + 2\|H\| \|G_k\| \left\|G_j\right\| \tag{3.B.115}$$

$$= \frac{1}{2} \left\| \left\{ H, \left[\partial_k \Phi_\theta(G_j) \right] \right\} \right\| + 6\|H\| \left\|G_j\right\| \|G_k\|$$

$$\leq \|H\| \left\|\partial_k \Phi_\theta(G_j)\right\| + 6\|H\| \left\|G_j\right\| \|G_k\| . \tag{3.B.116}$$

For the first inequality, we first obtain the matrix elements of the Hessian from Proposition 35. Then, this inequality directly follows from the triangle inequality. The second inequality follows from Hölder's inequality and submultiplicativity of the spectral norm. The third inequality follows from the anticommutator bound and contracitivity under a mixture-of-unitaries channel, both of which we mention now. The anticommutator bound is the following: given two matrices A and B, we have $\|\{A, B\}\| \leq 2\|A\| \|B\|$. Now consider that

$$\left\|\partial_k \Phi_\theta(G_j)\right\|$$

$$\leq \left\| \int_{\mathbb{R}} dt \int_0^1 du\, it\, p(t) \begin{pmatrix} e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t} G_j e^{iG(\theta)t} \\ -e^{-iG(\theta)t} G_j e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t} \end{pmatrix} \right\| \tag{3.B.117}$$

$$\leq \int_{\mathbb{R}} dt \int_0^1 du\, |t|\, p(t) \begin{pmatrix} \left\| e^{(1-u)iG(\theta)t} G_k e^{uiG(\theta)t} G_j e^{iG(\theta)t} \right\| \\ + \left\| e^{-iG(\theta)t} G_j e^{-(1-u)iG(\theta)t} G_k e^{-uiG(\theta)t} \right\| \end{pmatrix} \tag{3.B.118}$$

$$\leq \int_{\mathbb{R}} dt \int_0^1 du\, |t|\, p(t) \begin{pmatrix} \left\| e^{(1-u)iG(\theta)t} \right\| \|G_k\| \left\| e^{uiG(\theta)t} \right\| \left\|G_j\right\| \left\| e^{iG(\theta)t} \right\| \\ + \left\| e^{-iG(\theta)t} \right\| \left\|G_j\right\| \left\| e^{-(1-u)iG(\theta)t} \right\| \|G_k\| \left\| e^{-uiG(\theta)t} \right\| \end{pmatrix} \tag{3.B.119}$$

$$= 2 \int_{\mathbb{R}} dt \int_0^1 du\, |t|\, p(t) \left\|G_j\right\| \|G_k\| \tag{3.B.120}$$

$$= 2 \left\|G_j\right\| \|G_k\| \int_{-\infty}^{\infty} dt\, |t|\, p(t) \tag{3.B.121}$$

$$\leq 2 \left\|G_j\right\| \|G_k\| . \tag{3.B.122}$$

Applying this to (3.B.116), we finally conclude that

$$\left|\partial_k \partial_j \operatorname{Tr}[H\rho(\theta)]\right| \leq \|H\| \left\|\partial_k \Phi_\theta(G_j)\right\| + 6\|H\| \left\|G_j\right\| \|G_k\| \tag{3.B.123}$$

$$\leq 8 \|H\| \left\|G_j\right\| \|G_k\|, \tag{3.B.124}$$

thus completing the proof. ■

3.B.5 Smoothness

From the development above, we are now in a position to prove that the objective function $\mathrm{Tr}[H\rho(\theta)]$ is smooth. We do so in the proof of the following proposition.

Proposition 37 (Smoothness) *Let H be a Hamiltonian as defined in (3.B.1). Let $J \in \mathbb{N}$, let $\theta \in \mathbb{R}^J$ be a parameter vector, and let $\rho(\theta)$ be the corresponding parameterized thermal state as defined in (3.B.5). Then, the objective function $\mathrm{Tr}[H\rho(\theta)]$ is ℓ-smooth, where*

$$\ell = 8J \|H\| \max\left\{\|G_k\|^2\right\}_k. \tag{3.B.125}$$

Proof. To prove that the objective function $\mathrm{Tr}[H\rho(\theta)]$ is ℓ-smooth, we need to show that its gradient $\nabla_\theta \mathrm{Tr}[H\rho(\theta)]$ is ℓ-Lipschitz continuous (see Definition 22). To this end, note that this gradient is a multivariate vector-valued function. Therefore, it is ℓ-Lipschitz continuous if all its components, i.e., $\{\partial_j \mathrm{Tr}[H\rho(\theta)]\}_j$, are Lipschitz continuous. Let ℓ_j be a Lipschitz constant of the function $\partial_j \mathrm{Tr}[H\rho(\theta)]$. Then, from Lemma 24, it directly follows that a choice for ℓ is

$$\ell = \left(\sum_{j=1}^{J} \ell_j^2\right)^{\frac{1}{2}}. \tag{3.B.126}$$

Next, we get the Lipschitz constant ℓ_j by using Lemma 23 and the bounds that we obtained on the elements of the Hessian in Proposition 36:

$$\ell_j = \sqrt{J} \max_k \left\{\sup_\theta \left|\partial_k \partial_j \mathrm{Tr}[H\rho(\theta)]\right|\right\}_k \tag{3.B.127}$$

$$\leq 8\sqrt{J}\max_k \left\{ \|H\| \left\|G_j\right\| \|G_k\| \right\}_k \tag{3.B.128}$$

$$\leq 8\sqrt{J}\,\|H\| \left\|G_j\right\| \max_k \left\{ \|G_k\| \right\}_k . \tag{3.B.129}$$

Substituting the above equation into (3.B.126), we find that

$$\ell \leq \left(\sum_{j=1}^{J} \left[8\sqrt{J}\,\|H\| \left\|G_j\right\| \max_k \left\{ \|G_k\| \right\}_k \right]^2 \right)^{\frac{1}{2}} \tag{3.B.130}$$

$$= 8J^{\frac{1}{2}} \Bigg(\|H\|^2 \max_k \left\{ \|G_k\|^2 \right\}_k \underbrace{\sum_{j=1}^{J} \left\|G_j\right\|^2}_{\leq J \max_j \left\{ \|G_j\|^2 \right\}_j} \Bigg)^{\frac{1}{2}} \tag{3.B.131}$$

$$\leq 8J\,\|H\| \max_k \left\{ \|G_k\|^2 \right\}_k . \tag{3.B.132}$$

Finally, we take the right-hand side of the above inequality as the Lipschitz constant ℓ. ■

3.B.6 Quantum Boltzmann Gradient Estimator

In this section, we present a quantum algorithm for estimating the gradient $\nabla_\theta \operatorname{Tr}[H\rho(\theta)]$. As mentioned in the main text and previously in Section 3.B.1, we refer to this algorithm as the quantum Boltzmann gradient estimator (QBGE). For the sake of simplicity, we assume in what follows that for all $k \in [K]$ and for all $j \in [J]$, the Hamiltonians H_k and G_j are local unitaries. However, let us note that our algorithm can straightforwardly be generalized beyond the case of H_k and G_j being local unitaries, if they instead are block encoded into unitary circuits [LC19, GSLW19].

The gradient is a multivariate vector-valued function with components corresponding to the partial derivatives $\{\partial_j \operatorname{Tr}[H\rho(\theta)]\}_j$. Therefore, QBGE estimates the gradient by estimating these partial derivatives individually using Algorithms 1 and 2 as subroutines.

We begin by presenting these subroutines first and then provide pseudocode for QBGE at the end of this section, showcasing how QBGE employs these subroutines. To this end, let us recall the explicit form of $\partial_j \operatorname{Tr}[H\rho(\theta)]$ from Proposition 32:

$$\partial_j \operatorname{Tr}[H\rho(\theta)] = -\frac{1}{2}\left\langle \left\{H, \Phi_\theta(G_j)\right\}\right\rangle_{\rho(\theta)} + \langle H\rangle_{\rho(\theta)} \left\langle G_j\right\rangle_{\rho(\theta)}. \tag{3.B.133}$$

Observe that the above equation is a linear combination of two terms. Therefore, we present algorithms (Algorithms 1 and 2) for estimating these terms separately.

Estimating the first term

Expanding the first term of (3.B.133) yields:

$$\begin{aligned}
&-\frac{1}{2}\operatorname{Tr}\Big[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\Big] \\
&= -\frac{1}{2}\operatorname{Tr}\Big[\Big(H\Phi_\theta(G_j) + \Phi_\theta(G_j)H\Big)\rho(\theta)\Big] && (3.B.134)\\
&= -\frac{1}{2}\operatorname{Tr}\Bigg[\Bigg(\Bigg(\sum_k \alpha_k H_k\Bigg)\Bigg(\int_{\mathbb{R}} dt\, p(t) e^{iG(\theta)t} G_j e^{-iG(\theta)t}\Bigg) \\
&\qquad\qquad + \Bigg(\int_{\mathbb{R}} dt\, p(t) e^{iG(\theta)t} G_j e^{-iG(\theta)t}\Bigg)\Bigg(\sum_k \alpha_k H_k\Bigg)\Bigg)\rho(\theta)\Bigg] && (3.B.135)\\
&= -\sum_k \alpha_k \int_{\mathbb{R}} dt\, p(t) \Bigg(\frac{1}{2}\operatorname{Tr}\Bigg[\Bigg(\underbrace{H_k e^{iG(\theta)t} G_j e^{-iG(\theta)t}}_{-:U_{jk}(\theta,t)} + \underbrace{e^{iG(\theta)t} G_j e^{-iG(\theta)t} H_k}_{=U_{jk}^\dagger(\theta,t)}\Bigg)\rho(\theta)\Bigg]\Bigg) && (3.B.136)\\
&= -\sum_k \alpha_k \int_{\mathbb{R}} dt\, p(t) \Bigg(\frac{1}{2}\operatorname{Tr}\Big[\Big(U_{jk}(\theta,t) + U_{jk}^\dagger(\theta,t)\Big)\rho(\theta)\Big]\Bigg). && (3.B.137)
\end{aligned}$$

Note that, in the third equality, $H_k e^{iG(\theta)t} G_j e^{-iG(\theta)t}$ is a unitary because we are assuming that $\{H_k\}_k$ and $\{G_j\}_j$ are local unitaries.

The key idea here is to estimate the term on the right-hand side of (3.B.137). However, before presenting a quantum algorithm to accomplish this, let us first recall a fundamental

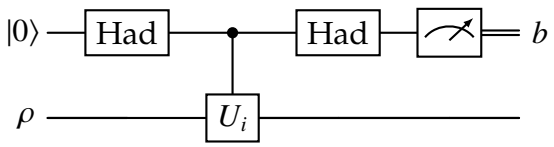


Figure 3.4: Quantum primitive for estimating $\frac{1}{2}\operatorname{Tr}\left[\left(U_1^\dagger U_0 + U_0^\dagger U_1\right)\rho\right]$. Note that the "Had" gate denotes the Hadamard gate.

primitive on which this algorithm is based. This primitive estimates a quantity of the form

$$\frac{1}{2}\operatorname{Tr}\left[\left(U_1^\dagger U_0 + U_0^\dagger U_1\right)\rho\right], \tag{3.B.138}$$

where U_0 and U_1 are unitaries and ρ is a quantum state. Observe that the expression above is similar to the expression on the right-hand side of (3.B.137), so understanding how this primitive works is crucial. The quantum circuit for this primitive is depicted in Figure 3.4, where the controlled gate is given by $|0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1$. Moreover, this circuit consists of the following two quantum registers: 1) a control register, initialized in the state $|0\rangle$, and 2) a system register, which is in the state ρ. After executing this circuit and obtaining a measurement outcome b in the control register, the final state $\sigma_{\text{sub}}^{(b)}$ (sub-normalized) of the system register is as follows, where $b \in \{0, 1\}$:

$$\sigma_{\text{sub}}^{(b)} = \left(\frac{U_0 + (-1)^b U_1}{2}\right)\rho\left(\frac{U_0^\dagger + (-1)^b U_1^\dagger}{2}\right) \tag{3.B.139}$$

$$= \frac{1}{4}\left(U_0\rho U_0^\dagger + (-1)^b U_0\rho U_1^\dagger + (-1)^b U_1\rho U_0^\dagger + U_1\rho U_1^\dagger\right). \tag{3.B.140}$$

The probability p_b of obtaining the measurement outcome b is then

$$p_b = \operatorname{Tr}\left[\sigma_{\text{sub}}^{(b)}\right] = \frac{2 + (-1)^b \operatorname{Tr}\left[U_0\rho U_1^\dagger\right] + (-1)^b \operatorname{Tr}\left[U_1\rho U_0^\dagger\right]}{4} \tag{3.B.141}$$

$$= \frac{2 + (-1)^b \operatorname{Tr}\left[\left(U_1^\dagger U_0 + U_0^\dagger U_1\right)\rho\right]}{4}. \tag{3.B.142}$$

In order to estimate the quantity, given by (3.B.138), we use the following approach. Let $b_1, \ldots, b_N$ represent the measurement results obtained from N independent executions

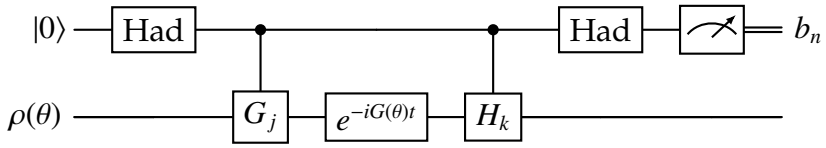


Figure 3.5: Quantum circuit corresponding to the n^{th} iteration of Algorithm 1.

of the aforementioned quantum circuit. We define a random variable X as $X \coloneqq (-1)^b$. Then the sample mean

$$\overline{X} \coloneqq \frac{1}{N}\sum_{i=1}^{N} X_i \tag{3.B.143}$$

serves as an unbiased estimator for the quantity of interest, because

$$\mathbb{E}\Big[\overline{X}\Big] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} X_i\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[X_i] = \frac{1}{N}\sum_{i=1}^{N}\sum_{b_i\in\{0,1\}} p_{b_i}(-1)^{b_i} \tag{3.B.144}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{2+\operatorname{Tr}\Big[\Big(U_1^\dagger U_0 + U_0^\dagger U_1\Big)\rho\Big]}{4} - \frac{2-\operatorname{Tr}\Big[\Big(U_1^\dagger U_0 + U_0^\dagger U_1\Big)\rho\Big]}{4}\right) \tag{3.B.145}$$

$$= \frac{\operatorname{Tr}\Big[\Big(U_1^\dagger U_0 + U_0^\dagger U_1\Big)\rho\Big]}{2}. \tag{3.B.146}$$

With the above primitive in mind, we are now in a position to present an algorithm (Algorithm 1) to estimate the first term of (3.B.133), i.e., $-\frac{1}{2}\operatorname{Tr}\Big[\Big\{H, \Phi_\theta(G_j)\Big\}\rho(\theta)\Big]$, using its equivalent form given by (3.B.137). Let $\alpha \coloneqq (\alpha_1, \ldots, \alpha_K)^\mathsf{T} \in \mathbb{R}^K_{\geq 0}$. For this algorithm, we assume that we have access to an oracle that samples an index $k \in [K]$ and time $t \in \mathbb{R}$ with probabilities ${}^{\alpha_k}/_{\|\alpha\|_1}$ and $p(t)$, respectively. Additionally, we assume that we have access to multiple copies of $\rho(\theta)$. At the core of our algorithm lies the aforementioned primitive with $U_0 = e^{-iG(\theta)t}$ and $U_1 = H_k e^{-iG(\theta)t} G_j$ (see Figure 3.5), so that $U_1^\dagger U_0 = G_j e^{iG(\theta)t} H_k e^{-iG(\theta)t}$ and

$$\frac{1}{2}\operatorname{Tr}\Big[\Big(U_1^\dagger U_0 + U_0^\dagger U_1\Big)\rho(\theta)\Big] = \frac{1}{2}\operatorname{Tr}\Big[\Big(G_j e^{iG(\theta)t} H_k e^{-iG(\theta)t} + e^{iG(\theta)t} H_k e^{-iG(\theta)t} G_j\Big)\rho(\theta)\Big] \tag{3.B.147}$$

$$= \frac{1}{2}\operatorname{Tr}\Big[\Big(e^{-iG(\theta)t} G_j e^{iG(\theta)t} H_k + H_k e^{-iG(\theta)t} G_j e^{iG(\theta)t}\Big)\rho(\theta)\Big] \tag{3.B.148}$$

Algorithm 1: `estimate_first_term`$(\alpha, \{H_k\}_{k=1}^K, \theta, \{G_\ell\}_{\ell=1}^J, j, \varepsilon_1, \delta_1)$

Input: Hamiltonian coefficients vector $\alpha = (\alpha_1, \ldots, \alpha_K)^\mathsf{T} \in \mathbb{R}^K_{\geq 0}$, local Hamiltonians $\{H_k\}_{k=1}^K$, parameter vector $\theta = (\theta_1, \ldots, \theta_J)^\mathsf{T} \in \mathbb{R}^J$, Gibbs local Hamiltonians $\{G_\ell\}_{\ell=1}^J$, index $j \in [J]$, precision $\varepsilon_1 > 0$, error probability $\delta_1 > 0$

$N_1 \leftarrow \lceil 2 \|\alpha\|_1^2 \ln(2/\delta_1) / \varepsilon_1^2 \rceil$

for $n = 0$ to $N_1 - 1$ **do**

  Initialize the control register to $|0\rangle$

  Prepare the system register in the state $\rho(\theta)$

  Sample k and t with probabilities $\alpha_k / \|\alpha\|_1$ and $p(t)$, respectively

  Apply the Hadamard gate to the control register

  Apply the following unitaries to the control and system registers:

    • Controlled-G_j: G_j is a local unitary with the control on the control register

    • $e^{-iG(\theta)t}$: Hamiltonian simulation for time t on the system register

    • Controlled-H_k: H_k is a local unitary with the control on the control register

  Apply the Hadamard gate to the control register

  Measure the control register in the computational basis and store the measurement outcome b_n

  $Y_n^{(1)} \leftarrow \|\alpha\|_1 (-1)^{b_n + 1}$

end for

return $\overline{Y}^{(1)} \leftarrow \frac{1}{N_1} \sum_{n=0}^{N_1 - 1} Y_n^{(1)}$

$$= \frac{1}{2}\operatorname{Tr}\Big[\Big\{H_k, e^{-iG(\theta)t}G_j e^{iG(\theta)t}\Big\}\rho(\theta)\Big], \tag{3.B.149}$$

where used the fact that $[G(\theta), \rho(\theta)] = 0$ and cyclicity of trace to obtain the second equality.

The output of our algorithm is a random variable $\overline{Y}^{(1)}$, which we show is an unbiased estimator of $-\frac{1}{2}\operatorname{Tr}[\{H, \Phi_\theta(G_j)\}\rho(\theta)]$:

$$\mathbb{E}\Big[\overline{Y}^{(1)}\Big] = \mathbb{E}\left[\frac{1}{N_1}\sum_{n=0}^{N_1-1} Y_n^{(1)}\right] = \mathbb{E}\left[\frac{1}{N_1}\sum_{n=0}^{N_1-1} \|\alpha\|_1 (-1)^{b_n+1}\right] \tag{3.B.150}$$

$$= -\frac{\|\alpha\|_1}{N_1}\sum_{n=0}^{N_1-1}\mathbb{E}\Big[(-1)^{b_n}\Big] = -\frac{\|\alpha\|_1}{N_1}\sum_{n=0}^{N_1-1}\sum_{b_n\in\{0,1\}} p_{b_n}\Big[(-1)^{b_n}\Big], \tag{3.B.151}$$

where

$$p_{b_n} \coloneqq \sum_k \frac{\alpha_k}{\|\alpha\|_1}\int_{\mathbb{R}} dt\, p(t)\left(\frac{2 + (-1)^{b_n}\operatorname{Tr}\Big[\Big(U_{jk}(\theta,t) + U_{jk}^\dagger(\theta,t)\Big)\rho(\theta)\Big]}{4}\right). \tag{3.B.152}$$

The above expression for p_{b_n} follows from the fact that we first sample an index k and time t with probabilities $\alpha_k/\|\alpha\|_1$ and $p(t)$, respectively, and then we apply the primitive introduced before with $U_0 = e^{-iG(\theta)t}$ and $U_1 = H_k e^{-iG(\theta)t}G_j$, whose probability of outputting a bit b is given by (3.B.142). Plugging the expression above for p_{b_n} into (3.B.151) and simplifying, we finally get

$$\mathbb{E}\Big[\overline{Y}^{(1)}\Big] = -\sum_k \alpha_k \int_{\mathbb{R}} dt\, p(t)\left(\frac{1}{2}\operatorname{Tr}\Big[\Big(U_{jk}(\theta,t) + U_{jk}^\dagger(\theta,t)\Big)\rho(\theta)\Big]\right) \tag{3.B.153}$$

$$= -\frac{1}{2}\operatorname{Tr}\Big[\Big\{H, \Phi_\theta(G_j)\Big\}\rho(\theta)\Big], \tag{3.B.154}$$

where the second equality follows due to (3.B.137).

Estimating the second term

We now expand the second term on the right-hand side of (3.B.133):

$$\langle H\rangle_{\rho(\theta)}\Big\langle G_j\Big\rangle_{\rho(\theta)} = \operatorname{Tr}[H\rho(\theta)]\operatorname{Tr}[G_j\rho(\theta)] \tag{3.B.155}$$

$$= \operatorname{Tr}\left[\left(\sum_k \alpha_k H_k\right)\rho(\theta)\right]\operatorname{Tr}\left[G_j\rho(\theta)\right] \tag{3.B.156}$$

$$= \sum_k \alpha_k \operatorname{Tr}[H_k\rho(\theta)]\operatorname{Tr}\left[G_j\rho(\theta)\right]. \tag{3.B.157}$$

The second equality follows directly from the definition of H, given by (3.B.1).

Algorithm 2: `estimate_second_term`$(\alpha, \{H_k\}_{k=1}^K, G_j, \varepsilon_2, \delta_2)$

1: **Input:** Hamiltonian coefficients vector $\alpha = (\alpha_1, \ldots, \alpha_K)^\mathsf{T} \in \mathbb{R}^K_{\geq 0}$, local Hamiltonians $\{H_k\}_{k=1}^K$, parameter vector $\theta = (\theta_1, \ldots, \theta_J)^\mathsf{T} \in \mathbb{R}^J$, Gibbs local Hamiltonian G_j, precision $\varepsilon_2 > 0$, error probability $\delta_2 > 0$

2: $N_2 \leftarrow {}^{2\|\alpha\|_1^2 \ln(2/\delta_2)}/_{\varepsilon_2^2}$

3: **for** $n = 0$ to $N_2 - 1$ **do**

4: Sample k with probability ${}^{\alpha_k}/_{\|\alpha\|_1}$

5: Prepare a register in the state ρ_θ, measure H_k, and store the measurement outcome h_n

6: Prepare a register in the state ρ_θ, measure G_j, and store the measurement outcome g_n

7: Set $Y_n^{(2)} \leftarrow \|\alpha\|_1 (-1)^{h_n + g_n}$

8: **end for**

9: **return** $\overline{Y}^{(2)} \leftarrow \frac{1}{N_2}\sum_{n=0}^{N_2-1} Y_n^{(2)}$

Then Algorithm 2 estimates the above quantity. As we did previously for Algorithm 1, we assume that we have access to an oracle that samples an index $k \in [K]$ with probability ${}^{\alpha_k}/_{\|\alpha\|_1}$ and that we have access to multiple copies of $\rho(\theta)$. The output of this algorithm is a random variable $\overline{Y}^{(2)}$, which can be easily shown to be an unbiased estimator of our quantity of interest, i.e., $\langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}$.

Estimating the full gradient using QBGE

With the algorithms (Algorithms 1 and 2) for estimating the first and second terms of the partial derivatives in place, we now provide pseudocode for QBGE in Algorithm 3. As previously mentioned, this algorithm estimates the full gradient $\nabla_\theta \operatorname{Tr}[H\rho(\theta)]$ and outputs an estimator $\overline{g}$, the latter of which can be easily verified to be an unbiased estimator of $\nabla_\theta \operatorname{Tr}[H\rho(\theta)]$.

Algorithm 3: $\texttt{QBGE}(\alpha, \{H_k\}_{k=1}^K, \theta, \{G_\ell\}_{\ell=1}^J, \varepsilon_1, \varepsilon_2, \delta_1, \delta_2)$

1: **Input:** Hamiltonian coefficients vector $\alpha = (\alpha_1, \ldots, \alpha_K)^\mathsf{T} \in \mathbb{R}^K_{\geq 0}$, local Hamiltonians $\{H_k\}_{k=1}^K$, parameter vector $\theta = (\theta_1, \ldots, \theta_J)^\mathsf{T} \in \mathbb{R}^J$, Gibbs local Hamiltonians $\{G_\ell\}_{\ell=1}^J$, precisions $\varepsilon_1, \varepsilon_1 > 0$, error probabilities $\delta_1, \delta_2 > 0$

2: **for** $j = 1$ to J **do**

3: $\quad \overline{Y}_j^{(1)} \leftarrow \texttt{estimate\_first\_term}(\alpha, \{H_k\}_{k=1}^K, \theta, \{G_\ell\}_{\ell=1}^J, j, \varepsilon_1, \delta_1)$

4: $\quad \overline{Y}_j^{(2)} \leftarrow \texttt{estimate\_second\_term}(\alpha, \{H_k\}_{k=1}^K, G_j, \varepsilon_2, \delta_2)$

5: $\quad \overline{g}_j \leftarrow \overline{Y}_j^{(1)} + \overline{Y}_j^{(2)}$

6: **end for**

7: **return** $\overline{g} \leftarrow (\overline{g}_1, \ldots, \overline{g}_J)^\mathsf{T}$

Remark 38 *Given the form of the Hessian in Proposition 35, along with the fact that the function $|t|\, p(t)$ is normalizable as a probability density function (i.e., $\int_{\mathbb{R}} dt\, |t|\, p(t) \approx 0.2714$), we can devise a quantum algorithm for estimating the matrix elements of the Hessian. The algorithm employs ideas similar to those used in this section to derive the algorithm for estimating the elements of the gradient. This quantum algorithm for estimating the matrix elements of the Hessian can be used in a second-order stochastic Newton search method, thus extending the SGD algorithm used in this chapter. We leave the detailed exploration of this approach for future work.*

3.B.7 SGD for QBM Learning of Ground-State Energies

In the previous section, we introduced QBGE (see Algorithm 3), an algorithm for estimating the gradient $\nabla_\theta \operatorname{Tr}[H\rho(\theta)]$. This algorithm outputs an unbiased estimator $\overline{g}(\theta)$ of the gradient at a given point $\theta \in \mathbb{R}^J$. Note that from now on, we use the notation $\overline{g}(\theta)$ instead of simply using $\overline{g}$ (as in Algorithm 3) to emphasize the explicit dependence on θ. Having said that, in this section, we present an algorithm that uses SGD for QBM learning of ground-state energies (Definition 28).

Algorithm 4: $\texttt{QBM\_GSE}(\alpha, \{H_k\}_{k=1}^K \{G_\ell\}_{\ell=1}^J, \varepsilon)$

Input: Hamiltonian coefficients vector $\alpha = (\alpha_1, \ldots, \alpha_K)^\intercal \in \mathbb{R}^K_{\geq 0}$, local Hamiltonians $\{H_k\}_{k=1}^K$, Gibbs local Hamiltonians $\{G_\ell\}_{\ell=1}^J$, precision $\varepsilon > 0$

$\theta \leftarrow$ Random Initialization

$M \leftarrow \left\lceil \frac{12\ell\Delta}{\varepsilon^2} \right\rceil$

$\varepsilon_1, \varepsilon_2 \leftarrow \frac{\varepsilon}{2\sqrt{2J}}$

$\delta_1, \delta_2 \leftarrow \frac{\varepsilon^2}{8J\|\alpha\|_1^2}$

for $m = 1$ to M **do**

  $\overline{g}(\theta_m) \leftarrow \texttt{QBGE}(\alpha, \{H_k\}_{k=1}^K, \theta_m, \{G_\ell\}_{\ell=1}^J, \varepsilon_1, \varepsilon_2, \delta_1, \delta_2)$

  $\theta_{m+1} \leftarrow \theta_{m+1} - \eta \overline{g}(\theta_m)$

end for

return $\min_{m \in [M]} \operatorname{Tr}[H\rho(\theta_m)]$

3.B.8 Sample Complexity

In this section, we investigate the sample complexity – the number of samples of the thermal state $\rho(\theta)$ – required by the QBM-GSE algorithm (Algorithm 4) to reach an ε-stationary point of the optimization problem defined in (3.B.9). To simplify the discussion, we divide the analysis into two parts. First, we investigate the sample complexities of Algorithms 1 and 2 and then investigate the sample complexity of the QBM-GSE algorithm itself. This is because the QBM-GSE algorithm employs QBGE for gradient estimation, which, in turn, employs Algorithms 1 and 2 for estimating partial derivatives.

Sample complexities of Algorithms 1 and 2

Recall that Algorithms 1 and 2 output estimates of the first and second terms of the partial derivative $\partial_j \mathrm{Tr}[H\rho(\theta)]$, where

$$\partial_j \mathrm{Tr}[H\rho(\theta)] = -\frac{1}{2}\left\langle \left\{H, \Phi_\theta(G_j)\right\}\right\rangle_{\rho(\theta)} + \langle H\rangle_{\rho(\theta)} \left\langle G_j\right\rangle_{\rho(\theta)}. \tag{3.B.158}$$

We demonstrated in Section 3.B.6 that these estimators are unbiased. The next step is to investigate how fast these estimators converge to their respective expected values, which we do formally in the proofs of the following two lemmas.

Lemma 39 (Sample Complexity of Algorithm 1) *Let $\varepsilon_1 > 0$, $J, K \in \mathbb{N}$, $\delta_1 \in (0,1)$, $\theta \in \mathbb{R}^J$, $j \in [J]$, and $\alpha \in \mathbb{R}^K_{\geq 0}$. Then, the number of samples, N_1, of $\rho(\theta)$ used by Algorithm 1 to produce an ε_1-close estimate of $-\frac{1}{2}\mathrm{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\right]$ with a success probability not less than $1-\delta_1$ is*

$$N_1 = \left\lceil \frac{2\left\|\alpha\right\|_1^2 \ln(2/\delta_1)}{\varepsilon_1^2} \right\rceil. \tag{3.B.159}$$

Proof. Recall that Algorithm 1 outputs an unbiased estimator $\overline{Y}^{(1)}$ of $-\frac{1}{2}\operatorname{Tr}\left[\left\{H, \Phi_\theta(G_j)\right\}\rho(\theta)\right]$:

$$\overline{Y}^{(1)} = \frac{1}{N_1}\sum_{n=0}^{N_1-1} Y_n^{(1)}, \tag{3.B.160}$$

where $Y_n^{(1)} = \|\alpha\|_1 (-1)^{b_n+1}$ and $b_n \in \{0,1\}$, for all $n \in \{0,\ldots,N_1-1\}$. This implies that $Y_n^{(1)}$ lies in the range

$$-\|\alpha\|_1 \leq Y_n^{(1)} \leq \|\alpha\|_1 . \tag{3.B.161}$$

Now, using the Hoeffding inequality (Lemma 26), we can say that for $\varepsilon_1 > 0$, we have

$$\Pr\left(\left|\overline{Y}^{(1)} - \mathbb{E}\left[\overline{Y}^{(1)}\right]\right| \geq \varepsilon_1\right) \leq \exp\left(\frac{-N_1\varepsilon_1^2}{2\|\alpha\|_1^2}\right). \tag{3.B.162}$$

This implies that for

$$N_1 \geq \frac{2\|\alpha\|_1^2 \ln(2/\delta_1)}{\varepsilon_1^2}, \tag{3.B.163}$$

we have

$$\Pr\left(\left|\overline{Y}^{(1)} - \mathbb{E}\left[\overline{Y}^{(1)}\right]\right| \leq \varepsilon_1\right) \geq 1-\delta_1, \tag{3.B.164}$$

thus concluding the proof. ■

Lemma 40 (Sample Complexity of Algorithm 2) *Let $\varepsilon_2 > 0$, $J, K \in \mathbb{N}$, $\delta_2 \in (0,1)$, $\theta \in \mathbb{R}^J$, $j \in [J]$, and $\alpha \in \mathbb{R}^K_{\geq 0}$. Then, the number of samples, N_2, of $\rho(\theta)$ used by Algorithm 2 to produce an ε_2-close estimate of $\langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}$ with a success probability not less than $1-\delta_2$ is*

$$N_2 = \left\lceil \frac{2\|\alpha\|_1^2 \ln(2/\delta_2)}{\varepsilon_2^2} \right\rceil. \tag{3.B.165}$$

Proof. The proof follows a similar line of reasoning as that of Lemma 39, and we provide it here for completeness. Recall that Algorithm 2 outputs an unbiased estimator $\overline{Y}^{(2)}$ of $\langle H\rangle_{\rho(\theta)}\left\langle G_j\right\rangle_{\rho(\theta)}$:

$$\overline{Y}^{(2)} = \frac{1}{N_2}\sum_{n=0}^{N_2-1} Y_n^{(2)}, \tag{3.B.166}$$

where $Y_n^{(2)} = \|\alpha\|_1 (-1)^{h_n+g_n}$ and $h_n, g_n \in \{0,1\}$, for all $n \in \{0,\dots,N_2-1\}$. This implies that $Y_n^{(2)}$ lies in the range

$$-\|\alpha\|_1 \leq Y_n^{(2)} \leq \|\alpha\|_1 . \tag{3.B.167}$$

Now, using the Hoeffding inequality (Lemma 26), we can say that for $\varepsilon_2 > 0$, we have

$$\Pr\left(\left|\overline{Y}^{(2)} - \mathbb{E}\left[\overline{Y}^{(2)}\right]\right| \geq \varepsilon_2\right) \leq \exp\left(\frac{-N_2\varepsilon_2^2}{2\|\alpha\|_1^2}\right). \tag{3.B.168}$$

This implies that for

$$N_2 \geq \frac{2\|\alpha\|_1^2 \ln(2/\delta_2)}{\varepsilon_2^2}, \tag{3.B.169}$$

we have

$$\Pr\left(\left|\overline{Y}^{(2)} - \mathbb{E}\left[\overline{Y}^{(2)}\right]\right| \leq \varepsilon_2\right) \geq 1-\delta_2, \tag{3.B.170}$$

thus concluding the proof. ■

Sample complexity of QBM-GSE

Using the development above, in the proof of the following theorem, we analyze the sample complexity of the QBM-GSE algorithm (Algorithm 4).

Theorem 41 (Sample Complexity of QBM-GSE) *Let H be a Hamiltonian as defined in (3.B.1), and let $\alpha \in \mathbb{R}^K_{\geq 0}$ be the coefficients vector of H. Let $\varepsilon > 0$ and $J \in \mathbb{N}$. Then the sample complexity, N, of the QBM-GSE algorithm (Algorithm 4) to reach an ε-stationary point of the optimization problem (3.B.9) is given by*

$$N = 2J\left\lceil\frac{12\ell\Delta}{\varepsilon^2}\right\rceil\left\lceil\frac{8J\|\alpha\|_1^2 \ln(16J\|\alpha\|_1^2/\varepsilon^2)}{\varepsilon^2}\right\rceil, \tag{3.B.171}$$

where the smoothness parameter ℓ is defined in (3.B.125), $\Delta := \mathrm{Tr}[H\rho(\theta_0)] - \inf_{\theta\in\mathbb{R}^J}\mathrm{Tr}[H\rho(\theta)]$, and $\theta_0 \in \mathbb{R}^J$ is a randomly chosen initial point.

Proof. Note that QBM-GSE is an SGD algorithm, where the stochastic gradients $\overline{g}(\theta)$, at any given point θ, are estimated using QBGE (Algorithm 3):

$$\overline{g}(\theta) = (\overline{g}_1(\theta), \ldots, \overline{g}_J(\theta))^\mathsf{T}, \tag{3.B.172}$$

where $\overline{g}_j(\theta)$ is the stochastic partial derivative given as

$$\overline{g}_j(\theta) = \overline{Y}_j^{(1)}(\theta) + \overline{Y}_j^{(2)}(\theta). \tag{3.B.173}$$

Here, QBGE evaluates $\overline{Y}_j^{(1)}(\theta)$ and $\overline{Y}_j^{(2)}(\theta)$ using Algorithms 1 and 2, respectively.

From Section 3.A.3, we know that in order to use SGD for optimization, the stochastic gradient should be unbiased. This is true for our case; i.e., for all $\theta \in \mathbb{R}^J$, we have $\mathbb{E}[\overline{g}(\theta)] = \nabla_\theta \operatorname{Tr}[H\rho(\theta)]$, and we showed this previously in Section 3.B.6.

Another requirement for SGD is that the variance of the stochastic gradient should be bounded from above. Specifically, the stochastic gradient should satisfy the condition given by (3.A.10) for some constants A, B, and C. Therefore, we now proceed to obtain these constants for our case. Consider that

$$\begin{aligned}
&\mathbb{E}\Big[\|\overline{g}(\theta)\|^2\Big] \\
&= \mathbb{E}\Big[\|\overline{g}(\theta) - \nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2\Big] + \|\nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2 && (3.B.174)\\
&= \mathbb{E}\left[\sum_{j=1}^{J} \Big(\overline{g}_j(\theta) - \partial_j \operatorname{Tr}[H\rho(\theta)]\Big)^2\right] + \|\nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2 && (3.B.175)\\
&= \sum_{j=1}^{J} \mathbb{E}\Big[\Big(\overline{g}_j(\theta) - \partial_j \operatorname{Tr}[H\rho(\theta)]\Big)^2\Big] + \|\nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2 && (3.B.176)\\
&= \sum_{j=1}^{J} \mathbb{E}\left[\left(\overline{Y}_j^{(1)}(\theta) + \overline{Y}_j^{(2)}(\theta) - \left[-\frac{1}{2}\operatorname{Tr}\Big[\Big\{H, \Phi_\theta(G_j)\Big\}\rho(\theta)\Big] + \langle H\rangle_{\rho(\theta)} \Big\langle G_j\Big\rangle_{\rho(\theta)}\right]\right)^2\right] \\
&\qquad + \|\nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2 && (3.B.177)\\
&= \sum_{j=1}^{J} \operatorname{Var}\Big[\overline{Y}_j^{(1)}(\theta) + \overline{Y}_j^{(2)}(\theta)\Big] + \|\nabla_\theta \operatorname{Tr}[H\rho(\theta)]\|^2 && (3.B.178)
\end{aligned}$$

$$= \sum_{j=1}^{J} \mathrm{Var}\left[\overline{Y}_j^{(1)}(\theta)\right] + \mathrm{Var}\left[\overline{Y}_j^{(2)}(\theta)\right] + \|\nabla_\theta \mathrm{Tr}[H\rho(\theta)]\|^2 \tag{3.B.179}$$

$$\leq \sum_{j=1}^{J} (\varepsilon_1^2 + \delta_1 \|\alpha\|_1^2) + (\varepsilon_2^2 + \delta_2 \|\alpha\|_1^2) + \|\nabla_\theta \mathrm{Tr}[H\rho(\theta)]\|^2 \tag{3.B.180}$$

$$\leq J\left(\varepsilon_1^2 + \varepsilon_2^2 + (\delta_1 + \delta_2)\|\alpha\|_1^2\right) + \|\nabla_\theta \mathrm{Tr}[H\rho(\theta)]\|^2 . \tag{3.B.181}$$

The fourth equality follows from (3.B.173) and (3.B.158). The sixth equality follows from the fact that the variance of the sum of two independent random variables, X and Y, is equal to the sum of their individual variances, i.e., $\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$. The first inequality follows directly from the variance bounds of the sample means $\overline{Y}_j^{(1)}(\theta)$ and $\overline{Y}_j^{(2)}(\theta)$. Indeed, these are a consequence of the following reasoning. Letting $Y \in [-C, C]$ be a random variable such that $\Pr(|Y - \mathbb{E}[Y]| \leq \varepsilon') \geq 1 - \delta'$, for $\varepsilon' > 0$ and $\delta' \in (0, 1)$, and defining the set $\mathcal{S} := \{y : |y - \mathbb{E}[Y]| \leq \varepsilon'\}$, we find that

$$\mathrm{Var}[Y] = \sum_y p(y)|y - \mathbb{E}[Y]|^2 \tag{3.B.182}$$

$$= \sum_{y\in\mathcal{S}} p(y)|y - \mathbb{E}[Y]|^2 + \sum_{y\in\mathcal{S}^c} p(y)|y - \mathbb{E}[Y]|^2 \tag{3.B.183}$$

$$\leq \sum_{y\in\mathcal{S}} p(y)\varepsilon'^2 + \sum_{y\in\mathcal{S}^c} p(y)C^2 \tag{3.B.184}$$

$$\leq \varepsilon'^2 + \delta' C^2. \tag{3.B.185}$$

Applying this inequality and the Hoeffding bounds in (3.B.164) and (3.B.170), we conclude the first inequality. Now comparing the second inequality with the condition given by (3.A.10), we obtain the constants for our case: $A = 0, B = 1$, and $C = J\left(\varepsilon_1^2 + \varepsilon_2^2 + (\delta_1 + \delta_2)\|\alpha\|_1^2\right)$.

Recall from the convergence result of SGD (Lemma 27) that the total number of iterations, M, is bounded from below as follows:

$$M \geq \frac{12\ell\Delta}{\varepsilon^2} \max\left\{B, \frac{12A\Delta}{\varepsilon^2}, \frac{2C}{\varepsilon^2}\right\}. \tag{3.B.186}$$

Now, if we choose the algorithm parameters ε_1, ε_2, δ_1, and δ_2 such that the following holds:

$$2C = 2J\left(\varepsilon_1^2 + \varepsilon_2^2 + (\delta_1 + \delta_2)\,\|\alpha\|_1^2\right) \leq \varepsilon^2, \tag{3.B.187}$$

then the bound in (3.B.186) can be written as follows:

$$M \geq \frac{12\ell\Delta}{\varepsilon^2}. \tag{3.B.188}$$

This resolves the minimum number of iterations needed by SGD to reach an ε-stationary point, given that (3.B.187) holds.

Similarly, we can evaluate the step size η for SGD, which we do in the following way. Again recall from Lemma 27 that the step size η is given as follows:

$$\eta = \min\left\{\frac{1}{\sqrt{\ell AM}}, \frac{1}{\ell B}, \frac{\varepsilon}{2\ell C}\right\}. \tag{3.B.189}$$

Now, if we have that $0 < \varepsilon < 1$, then this condition along with the condition given by (3.B.187) implies the following:

$$2C \leq \varepsilon. \tag{3.B.190}$$

Using this inequality in (3.B.189), we finally obtain:

$$\eta = \frac{1}{\ell}. \tag{3.B.191}$$

That being said, the question is how to choose the algorithm parameters ε_1, ε_2, δ_1, and δ_2 such that the condition given by (3.B.187) holds. One way to do that is to choose $\varepsilon_1 = \varepsilon_2 = \varepsilon/2\sqrt{2J}$ and $\delta_1 = \delta_2 = \varepsilon^2/8J\|\alpha\|_1^2$. This now resolves the sample complexities N_1 and N_2 of Algorithms 1 and 2, respectively:

$$N_1 = N_2 = \left\lceil \frac{8J\,\|\alpha\|_1^2 \ln(16J\|\alpha\|_1^2/\varepsilon^2)}{\varepsilon^2} \right\rceil. \tag{3.B.192}$$

From this, we get the total sample complexity of the QBM-GSE algorithm:

$$N = M \cdot J(N_1 + N_2) = 2J \left\lceil \frac{12\ell\Delta}{\varepsilon^2} \right\rceil \left\lceil \frac{8J \|\alpha\|_1^2 \ln(16J\|\alpha\|_1^2/\varepsilon^2)}{\varepsilon^2} \right\rceil. \tag{3.B.193}$$

This concludes the proof. ■

3.C Addressing the Open Problem Regarding QBM Learning

We begin by recalling the problem at hand, originally put forward in [AAR+18]. Consider a probability distribution $P_{\mathbf{v}}^{\text{data}}$ defined by classical training data over a random variable $\mathbf{v}$. The goal is to learn this distribution using a parameterized model $P_{\mathbf{v}}(\theta)$, where θ represents a vector of parameters. More concretely, we aim to minimize the average negative log-likelihood $\mathcal{L}(\theta)$ between the target distribution $P_{\mathbf{v}}^{\text{data}}$ and the model distribution $P_{\mathbf{v}}(\theta)$:

$$\mathcal{L}(\theta) \coloneqq -\sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \log P_{\mathbf{v}}(\theta). \tag{3.C.1}$$

One way to realize the model distribution $P_{\mathbf{v}}(\theta)$ is via a QBM with some parameterized Hamiltonian $G(\theta)$ as defined in (3.B.4). More formally, we define:

$$P_{\mathbf{v}}(\theta) \coloneqq \text{Tr}[\Lambda_{\mathbf{v}}\rho(\theta)], \tag{3.C.2}$$

where $(\Lambda_{\mathbf{v}})_{\mathbf{v}}$ is an efficiently implementable measurement and $\rho(\theta)$ is the parameterized thermal state corresponding to $G(\theta)$ as defined in (3.B.5). Using the definition above, we can rewrite $\mathcal{L}(\theta)$ as follows:

$$\mathcal{L}(\theta) \coloneqq -\sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \log \text{Tr}[\Lambda_{\mathbf{v}}\rho(\theta)]. \tag{3.C.3}$$

The partial derivative of $\mathcal{L}(\theta)$ with respect to the parameter θ_j is given by

$$\partial_j \mathcal{L}(\theta) = -\sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \left(\frac{\text{Tr}[\Lambda_{\mathbf{v}} \partial_j \rho(\theta)]}{\text{Tr}[\Lambda_{\mathbf{v}}\rho(\theta)]} \right) \tag{3.C.4}$$

$$= -\sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \left(\frac{\text{Tr}[\Lambda_{\mathbf{v}} \left(-\frac{1}{2} \left\{ \Phi_\theta(G_j), \rho(\theta) \right\} + \rho(\theta) \left\langle G_j \right\rangle \right)]}{\text{Tr}[\Lambda_{\mathbf{v}} \rho(\theta)]} \right) \tag{3.C.5}$$

$$= \sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \frac{\frac{1}{2} \text{Tr}[\Lambda_{\mathbf{v}} \left\{ \Phi_\theta(G_j), \rho(\theta) \right\}]}{\text{Tr}[\Lambda_{\mathbf{v}} \rho(\theta)]} - \left\langle G_j \right\rangle \tag{3.C.6}$$

$$= \sum_{\mathbf{v}} P_{\mathbf{v}}^{\text{data}} \frac{\frac{1}{2} \text{Tr}[\left\{ \Lambda_{\mathbf{v}}, \Phi_\theta(G_j) \right\} \rho(\theta)]}{\text{Tr}[\Lambda_{\mathbf{v}} \rho(\theta)]} - \left\langle G_j \right\rangle, \tag{3.C.7}$$

where the second equality follows from (3.4.1). Previous works suggested that efficient gradient estimation was infeasible due to the apparent computational difficulty in evaluating the numerator of the first term. However, we demonstrate that this limitation can be overcome—Algorithm 1 provides an efficient method to estimate this term to arbitrary precision, thereby enabling efficient gradient computation.

We now analyze how errors in estimating both the numerator and denominator propagate to affect the final precision of their ratio. Let p and q be unbiased estimates of $\frac{1}{2} \text{Tr}[\left\{ \Lambda_{\mathbf{v}}, \Phi_\theta(G_j) \right\} \rho(\theta)]$ and $\text{Tr}[\Lambda_{\mathbf{v}} \rho(\theta)]$, respectively, such that the following holds for some $\varepsilon_1, \varepsilon_2 \geq 0$:

$$\left| p - \frac{1}{2} \text{Tr}[\left\{ \Lambda_{\mathbf{v}}, \Phi_\theta(G_j) \right\} \rho(\theta)] \right| \leq \varepsilon_1, \tag{3.C.8}$$

$$|q - \text{Tr}[\Lambda_{\mathbf{v}} \rho(\theta)]| \leq \varepsilon_2. \tag{3.C.9}$$

We also assume that there exists a scalar $r > 0$, such that $\text{Tr}\left[\Lambda_{\mathbf{v}} \rho(\theta)\right] \geq r$. This assumption is justified because the thermal state $\rho(\theta)$ is generically full-rank for typical parameter values. Finally, we assume that each G_j is a Pauli string, and that $\varepsilon_2 < r/2$. Note that the assumptions $\varepsilon_2 < r/2$ and $\text{Tr}\left[\Lambda_{\mathbf{v}} \rho(\theta)\right] \geq r$ imply that $\text{Tr}\left[\Lambda_{\mathbf{v}} \rho(\theta)\right] - \varepsilon_2 > 0$.

Proposition 42 *The following inequality holds:*

$$\left| \frac{p}{q} - \frac{\frac{1}{2} \text{Tr}\left[\Phi_\theta(G_j) \{ \Lambda_{\mathbf{v}}, \rho(\theta) \} \right]}{\text{Tr}\left[\Lambda_{\mathbf{v}} \rho(\theta) \right]} \right| \leq \frac{8\varepsilon_2^3}{r^4} + \frac{2\varepsilon_2}{r^2} + \frac{\varepsilon_1}{r}. \tag{3.C.10}$$

Proof. Consider that

$$\left|\frac{p}{q} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right|$$

$$= \left|\frac{p}{q} - \frac{p}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]} + \frac{p}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| \tag{3.C.11}$$

$$\leq \left|\frac{p}{q} - \frac{p}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| + \left|\frac{p}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| \tag{3.C.12}$$

$$\leq |p|\left|\frac{1}{q} - \frac{1}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| + \frac{1}{\left|\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]\right|}\left|p - \frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]\right|. \tag{3.C.13}$$

Using the facts that $|\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]| \geq r$, and $p \in [-1, 1]$, which holds because

$$\left|\frac{1}{2}\operatorname{Tr}[\left\{\Lambda_{\mathbf{v}}, \Phi_\theta(G_j)\right\}\rho(\theta)]\right| \leq \left\|\frac{1}{2}\left\{\Lambda_{\mathbf{v}}, \Phi_\theta(G_j)\right\}\right\| \|\rho(\theta)\|_1 \tag{3.C.14}$$

$$\leq \|\Lambda_{\mathbf{v}}\| \left\|\Phi_\theta(G_j)\right\| \tag{3.C.15}$$

$$\leq \left\|G_j\right\| \tag{3.C.16}$$

$$\leq 1, \tag{3.C.17}$$

and the circuit estimating p never returns a value outside of the interval $[-1, 1]$, we have that

$$\left|\frac{p}{q} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right|$$

$$\leq \left|\frac{1}{q} - \frac{1}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| + \frac{1}{r}\left|p - \frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]\right| \tag{3.C.18}$$

$$\leq \left|\frac{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right] - q}{q\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| + \frac{\varepsilon_1}{r} \tag{3.C.19}$$

$$\leq \frac{1}{|q|}\frac{\varepsilon_2}{r} + \frac{\varepsilon_1}{r}. \tag{3.C.20}$$

Now from (3.C.9), we have the following:

$$\frac{1}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right] + \varepsilon_2} \leq \frac{1}{q} \leq \frac{1}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right] - \varepsilon_2} \leq \frac{1}{r - \varepsilon_2}. \tag{3.C.21}$$

Using the above inequality, it follows that

$$\left|\frac{p}{q} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| \leq \frac{1}{(r-\varepsilon_2)}\frac{\varepsilon_2}{r} + \frac{\varepsilon_1}{r} \tag{3.C.22}$$

$$= \frac{1}{\frac{r}{2}\left(2 - \frac{2\varepsilon_2}{r}\right)}\frac{\varepsilon_2}{r} + \frac{\varepsilon_1}{r} \tag{3.C.23}$$

$$= \frac{1}{\left(2 - \frac{2\varepsilon_2}{r}\right)}\frac{2\varepsilon_2}{r^2} + \frac{\varepsilon_1}{r}. \tag{3.C.24}$$

Using the inequality $\frac{1}{2-x} \leq x^2 - x + 1$, which holds for all $x \in (0, 1)$, and applying the assumption that $2\varepsilon_2 < r$, we have

$$\left|\frac{p}{q} - \frac{\frac{1}{2}\operatorname{Tr}\left[\Phi_\theta(G_j)\{\Lambda_{\mathbf{v}}, \rho(\theta)\}\right]}{\operatorname{Tr}\left[\Lambda_{\mathbf{v}}\rho(\theta)\right]}\right| \leq \left(\left(\frac{2\varepsilon_2}{r}\right)^2 - \left(\frac{2\varepsilon_2}{r}\right) + 1\right)\frac{2\varepsilon_2}{r^2} + \frac{\varepsilon_1}{r} \tag{3.C.25}$$

$$\leq \frac{8\varepsilon_2^3}{r^4} + \frac{2\varepsilon_2}{r^2} + \frac{\varepsilon_1}{r}. \tag{3.C.26}$$

This concludes the proof. ■

CHAPTER 4

VARIATIONAL QUANTUM ALGORITHMS FOR SEMIDEFINITE PROGRAMMING[1]

4.1 Abstract

A semidefinite program (SDP) is a particular kind of convex optimization problem with applications in operations research, combinatorial optimization, quantum information science, and beyond. In this work, we propose variational quantum algorithms for approximately solving SDPs. For one class of SDPs, we provide a rigorous analysis of their convergence to approximate locally optimal solutions, under the assumption that they are weakly constrained (i.e., $N \gg M$, where N is the dimension of the input matrices and M is the number of constraints). We also provide algorithms for a more general class of SDPs that requires fewer assumptions. Finally, we numerically simulate our quantum algorithms for applications such as MaxCut, and the results of these simulations provide evidence that convergence still occurs in noisy settings.

4.2 Introduction

Semidefinite programming (SDP) is one of the most important tools in optimization developed over the past few decades. One can see SDPs as a natural extension of the better known linear programs (LPs), in which the vector inequalities of LPs are replaced by matrix inequalities. One of the reasons underlying the importance of SDPs is their applicability to a broad range of problems, including approximation algorithms for combi-

[1] This chapter is based verbatim on the work [PCW24], with some typos corrected.

natorial optimization [LS06], control theory [MHA20], and sum-of-squares [Par03]. Additionally, a variety of quantum information problems can be formulated as SDPs, including state discrimination [YKL75, Eld03], upper bounds on quantum channel capacity [WXD18, Wan18], and self-testing [SB20].

The power of SDPs lies with the fact that they can be solved efficiently in polynomial time using classical algorithms such as the celebrated interior-point method [PW00]. Although SDPs can be solved efficiently using classical techniques, as the size of the input matrices increases, many first-order and second-order algorithms incur significant computational overhead due to the expensive gradient computation at each iteration. For this reason, it is imperative to design more efficient algorithms for solving SDPs.

Given the speed-ups of quantum algorithms over classical algorithms for a variety of problems [Sho94, HHL09, Gro97, DH96, HV03], it is natural to ask if there exists a quantum algorithm that can solve SDPs efficiently. This question was positively answered in [BS17], wherein a quantum algorithm was proposed and proven to have a quadratic speedup over the classical Arora–Kale algorithm [AK16]. Following this initial result, more efficient quantum algorithms for solving SDPs were later developed [BKL+19, vAGGdW20].

Although quantum algorithms have been theoretically proven to outperform known classical algorithms for many applications, a fault-tolerant quantum computer is required to reap their benefits. Currently, fault-tolerant quantum computers are not available, and we are instead in the Noisy Intermediate-Scale Quantum (NISQ) era [Pre18]. Google constructed a noisy quantum computer with just 54 qubits [AAB+19], but it remains an open challenge to design a fault-tolerant quantum computer that requires millions of qubits for successful operation. Some of the quantum information science re-

search community is focused on determining the power of noisy quantum computers and is designing quantum algorithms that acknowledge the limitations of such devices [CAB+21, BCLK+22, CCHL23, Ang23], such as a finite number of gates and qubits, noisy gate execution, and rapid decoherence of qubits.

Variational quantum algorithms (VQAs) constitute an important class of NISQ-friendly algorithms [CAB+21, BCLK+22]. VQAs can be seen as hybrid quantum-classical algorithms that have a classical computer available for optimization, only calling a quantum subroutine for tasks that are not efficiently solvable by it. Hitherto, VQAs have been proposed for numerous computational tasks, for which there are also known quantum and classical algorithms [CAB+21]. Some well studied VQAs include the Variational Quantum Eigensolver (VQE) [PMS+14] and the Quantum Approximate Optimization Algorithm (QAOA) [FGG14].

This brings us to the main motivation of our work, where we reformulate three different kinds of SDPs and develop variational quantum algorithms for solving them. In particular, the contributions of our paper are as follows:

1. We present unconstrained reformulations of the constrained general and standard forms of an SDP, by employing a series of reductions (Sections 4.4.1, 4.4.2, and 4.4.3). For the standard form, we consider the case in which the SDP is weakly constrained (i.e., when $N \gg M$, where N is the dimension of the input matrices and M is the number of constraints). On the contrary, we do not require such assumptions when considering the general form, making it applicable to a wide range of problems.
2. We propose variational quantum algorithms for obtaining approximate stationary points of such unconstrained problems, which in turn are approximate stationary points of their respective constrained problems due to the equivalence between

them (Sections 4.4.1, 4.4.2, and 4.4.3). Note that stationary points include globally optimal points.

3. We analyze the convergence rate of an algorithm corresponding to the equality constrained standard formulation of an SDP (Section 4.4.2).

4. We provide numerical evidence that showcases how our algorithms work in practice for applications such as MaxCut (Section 4.5). Specifically, we analyze the convergence of the proposed algorithms by assessing how close the final value is to the actual optimal value evaluated using exact classical solvers. We perform such experiments on a noisy quantum simulator from Qiskit [T+23] (i.e., QASM simulator) and then compare these results with those on a noiseless simulator. Note that the QASM simulator mimics IBM's quantum computer, which is actually noisy in nature due to gate errors and decoherence.

Additionally, in Section 4.3, we introduce some notations and definitions, the basics of SDPs, and a brief overview of VQAs, as well as the associated concept of computing partial derivatives on a quantum circuit, i.e., the parameter-shift rule.

4.2.1 Main Idea and Setup

In this paper, we consider the following three different kinds of SDPs, and we present variational quantum algorithms for all three of them:

- *General Form (GF)*: The general form of an SDP can be concisely written as follows:

$$p^* := \sup_{X \succcurlyeq 0} \{\mathrm{Tr}[CX] : \Phi(X) \preccurlyeq B\}, \tag{4.2.1}$$

where $C \in \mathcal{S}^N$, $B \in \mathcal{S}^M$, and the map Φ is Hermiticity-preserving (Definition 45). Here, the notation $\mathcal{S}^N$ denotes the set of $N \times N$ Hermitian operators. For this case, we do not assume that the SDPs are weakly constrained; i.e., we do not assume that the dimension M of the constraint variable B is much smaller than the dimension N of the objective variable C. Additionally, to generalize it further, we consider an inequality-constrained problem. For solving these general SDPs, we propose a variational quantum algorithm (Algorithm 5).

- *Standard Form (SF)*: Here, we consider the Hermiticity-preserving map Φ and the Hermitian operator B to have a diagonal form, as given in (4.3.9). Specifically, in this case, we set

$$\begin{aligned} \Phi(X) &= \operatorname{diag}\left(\operatorname{Tr}[A_1 X], \ldots, \operatorname{Tr}[A_M X]\right), \\ B &= \operatorname{diag}\left(b_1, \ldots, b_M\right), \end{aligned} \tag{4.2.2}$$

  where $A_1, \ldots, A_M \in \mathcal{S}^N$ and $b_1, \ldots, b_M \in \mathbb{R}$. This form of an SDP is well known in the convex optimization literature [BV04], and most exact or approximation algorithms designed for combinatorial optimization problems are based on this form. For this case, we assume that the SDPs are weakly constrained, i.e., $N \gg M$. We further categorize them based on the nature of the constraints as follows:

  - *Equality Constrained Standard Form (ECSF)*: Here, we consider equality constraints, i.e., $\Phi(X) = B$. For solving such SDPs, we propose a variational quantum algorithm (Algorithm 6) and establish its convergence rate and total iteration complexity.
  - *Inequality Constrained Standard Form (ICSF)*: Here, we consider inequality constraints, i.e., $\Phi(X) \preccurlyeq B$, and we propose a variational quantum algorithm for this case (Algorithm 7).

In our methods, we first reduce these constrained optimization problems to their un-

constrained forms by employing a series of identities. Second, we express the final unconstrained form as a function of expectation values of the input Hermitian operators, i.e., $C, A_1, \ldots, A_M \in \mathcal{S}^N$.

For solving these final unconstrained formulations using a gradient-based method, we need access to the full gradient of the objective function at each iteration. However, when the dimension N of the input Hermitian operators is large, the evaluation of the gradient of the objective function using a classical computer becomes computationally expensive. Therefore, we delegate this gradient computation to parameterized quantum circuits, and we use a technique called the parameter-shift rule [LYPS17, MNKF18, SBG+19] to evaluate the partial derivatives of the objective function with respect to circuit parameters. Please refer to Section 4.3.2 for more details on the parameter-shift rule.

We design variational quantum algorithms for solving these unconstrained optimization problems. Our methods provide bounds on the optimal values, due to the reduction in the search space, as well as the non-convex nature of the objective function landscape in terms of quantum circuit parameters. We do not assume that the final objective function is convex with respect to these parameters, as generally it is non-convex [HD21]. In general, finding a globally optimal point for a non-convex function is known to be NP-hard [DDG+22, Section 2.1], and so an important question regards the type of solutions that we can guarantee in such a scenario. In the classical optimization literature, the notion of approximate stationary points (Definition 52) is considered when the objective function is non-convex. Therefore, in this paper, we focus on proving the convergence of our algorithms to approximate stationary points, as proving the same for a global optimal point is quite difficult. We would also like to emphasize that VQAs that have been proposed prior to our paper have not considered this notion, which is quite natural when considering non-convex objective functions.

Intuitively, stationary points are those for which the gradient of the function under consideration is equal to zero. Therefore, a stationary point can be a local maximum (including the global maximum), a local minimum (including the global minimum), or a saddle point. Consequently, when using a first-order solver such as gradient descent, one may get stuck at one of these points. However, it is often desirable to escape unwanted stationary points and move to a more favorable one, depending on the type of optimization problem (maximization or minimization). This can be achieved either through the use of higher-order solvers or by employing a noisy first-order solver [JGN+17]. We focus on the latter approach in this paper, as computing higher-order derivatives can be computationally expensive compared to calculating the first-order gradient. Moreover, due to the inherent stochastic nature of variational quantum algorithms (VQAs), a first-order solver can inherently be noisy. This noise can aid in escaping unwanted stationary points and converging to a better stationary point.

The primary reason behind proposing different variational quantum algorithms for three different kinds of SDPs is the nature of the final unconstrained forms of these SDPs. In the general form of an SDP, as we do not consider any assumptions on the input matrices, the final unconstrained optimization problem turns out to be a non-convex–non-concave optimization problem (see (4.2.3)). Proving the convergence of the proposed algorithm to an approximate stationary point of this problem requires sophisticated analysis, which we leave for future work. In contrast, although the standard form of SDPs is a special case of the general form, we consider cases where these SDPs are weakly constrained. Studying the standard form of SDPs with this assumption is important because many SDPs of interest are actually large and weakly constrained. Due to this assumption, we observe that the final unconstrained forms turn out to be non-convex–concave optimization problems (see (4.2.4) and (4.2.6)). Non-convex–concave optimiza-

tion [DISZ18, HRU+17, MLZ+19, MJS19, RLLY22] has a rich literature when compared to that of non-convex–non-concave optimization, as the latter is a harder problem than the former. Therefore, the design and convergence analysis of more sophisticated variational quantum algorithms for solving non-convex–concave optimization problems is possible.

Reformulation of General Form (GF) of SDPs—For the SDPs written as (4.2.1), we reduce this form to the following final unconstrained form expressed in terms of quantum circuit parameters $\boldsymbol{\theta}_1 \in [0, 2\pi]^{r_1}$ and $\boldsymbol{\theta}_2 \in [0, 2\pi]^{r_2}$:

$$p^* := \sup_{\substack{\boldsymbol{\theta}_1 \in [0,2\pi]^{r_1}, \\ \lambda \geq 0}} \inf_{\substack{\boldsymbol{\theta}_2 \in [0,2\pi]^{r_2}, \\ \mu \geq 0}} \left\{ \lambda \langle I \otimes C^\top \rangle_{\boldsymbol{\theta}_1} + \mu \langle I \otimes B \rangle_{\boldsymbol{\theta}_2} - \lambda\mu \langle I \otimes I \otimes \Gamma^\Phi \rangle_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2} \right\}, \tag{4.2.3}$$

where Γ^Φ is the Choi operator (Definition 46) of the linear map Φ. See Section 4.4.1 for more details. For solving the above optimization problem, we propose a VQA called *inexact Variational Quantum Algorithm for General Form* (iVQAGF), in which we have two parameterized quantum circuits competing against each other to maximize/minimize the objective function, and then there is a classical optimizer that updates the parameters of these quantum circuits.

Reformulation of Equality Constrained Standard Form (ECSF) of SDPs—As stated before, for this type of SDP, we make an assumption on it being weakly-constrained, i.e., $N \gg M$. By exploiting this assumption, we design more sophisticated variational quantum algorithms, in which we need just one parameterized quantum circuit. For such SDPs, we arrive at the following final unconstrained form expressed in terms of quantum circuit parameters $\boldsymbol{\theta} \in [0, 2\pi]^r$:

$$p^* := \sup_{\boldsymbol{\theta} \in [0,2\pi]^r} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \left\{ \lambda \langle I \otimes C \rangle_{\boldsymbol{\theta}} + \boldsymbol{y}^\top (\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\boldsymbol{\theta})) + \frac{c}{2} \| \boldsymbol{b} - \lambda \boldsymbol{\Phi}(\boldsymbol{\theta}) \|^2 \right\}. \tag{4.2.4}$$

See Section 4.4.2 for more details. Throughout this paper, we use the notation

$$\langle H \rangle_{\boldsymbol{\theta}} \equiv \langle \phi(\boldsymbol{\theta}) | H | \phi(\boldsymbol{\theta}) \rangle \tag{4.2.5}$$

to represent the expectation value of a Hermitian operator H with respect to $|\phi(\boldsymbol{\theta})\rangle$. For solving the above optimization problem, we propose a VQA called *inexact Variational Quantum Algorithm for Equality Constrained standard form* (iVQAEC). We run iVQAEC on a classical computer, and at any step of the algorithm, the expectation value of a Hermitian operator is evaluated using a parameterized quantum circuit. One of our main results establishes the convergence rate and total iteration complexity of iVQAEC under the assumption of the SDP being weakly-constrained.

Reformulation of Inequality Constrained Standard Form (ICSF) of SDPs—For this case also we make the weakly-constrained assumption on SDPs. Here, we solve the dual problem instead of the primal problem. For this type of SDP, we arrive at the following final unconstrained form expressed in terms of quantum circuit parameters $\boldsymbol{\theta} \in [0, 2\pi]^r$:

$$d' := \sup_{\boldsymbol{\theta}\in[0,2\pi]^r} \inf_{\bar{\boldsymbol{y}}\geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + \frac{b_M}{\gamma} \ln\left(e^{\gamma\langle I\otimes H(\bar{\boldsymbol{y}})\rangle_{\boldsymbol{\theta}}} + 1\right) \right\}. \tag{4.2.6}$$

See Section 4.4.3 for more details. For solving this problem, we propose a VQA called *inexact Variational Quantum Algorithm for Inequality Constrained standard form* (iVQAIC). For this algorithm, we do not establish its convergence rate, which we leave for future work; instead, we prove a property of the objective function that is necessary for providing such a convergence analysis, i.e., smoothness of the objective function for a fixed $\bar{\boldsymbol{y}}$.

4.2.2 Related Work

Recently, an approach to semidefinite programming on NISQ devices was proposed in [BHVK22], which is non-variational and called NISQ SDP solver (NSS) therein. We should note that this approach does not provide an efficient solution for a general SDP problem. Like our approach, the NSS approach also optimizes over a subset of the

positive-semidefinite operator space. The NSS approach assumes the ability to prepare pure states in a set $\{|\psi_i\rangle\}_i$. Using these states, the NSS approach then constructs a hybrid density matrix ansatz: $X_\beta = \sum_{i,j} \beta_{ij} |\psi_i\rangle\langle\psi_j|$, where the entries $\{\beta_{ij}\}_{ij}$ are stored on a classical computer. The NSS approach then transforms the original SDP into a low-dimensional SDP, which is solved by optimizing over these classical entries. Note that the NSS approach does not change the set of pure states for each iteration. As one can see, the NSS approach does not encompass the entire space of positive semidefinite operators; therefore, it is heuristic in nature, similar to ours. This means that there is no guarantee of convergence to the global optimal point.

Additionally, another technique was proposed in [MA21] where the authors "quantized" the classical randomized cutting plane method for solving semidefinite programs. They used a quantum eigensolver subroutine in order to speedup the classical method. Their results indicate that the robustness of their approach against noise may be useful in implementing their method on NISQ devices.

Our approach here is complementary to both of these approaches because our algorithm is a variational quantum algorithm. Another important point to note here is that, unlike prior works that focus solely on solving SDPs in the standard form, we propose algorithms for solving SDPs in both the standard form and a form considered in [Wat18, KW20]. The latter form is prevalent in quantum information theory, as it is used to compute various relevant quantities like fidelity and trace distance. While one can convert this form to the standard form, working in the original form is more convenient, avoiding unnecessary conversion overhead. Comparing the aforementioned approaches to ours is an interesting direction for future work.

4.3 Preliminaries

In this section, we introduce some notations and definitions.

Notations: We denote the set of real and complex numbers by $\mathbb{R}$ and $\mathbb{C}$, respectively. For a positive integer m, the notation $[m]$ denotes the set $\{1, \ldots, m\}$. We use upper-case letters to denote matrices and bold lower-case letters to denote vectors (e.g., A is a matrix, and $\boldsymbol{x}$ is a vector). Let $\mathcal{H}$ denote a finite-dimensional Hilbert space of dimension N. This Hilbert space represents a system of n qubits, where $n = \lceil \log_2 N \rceil$. The notation $L(\mathcal{H})$ denotes the set of linear operators acting on $\mathcal{H}$. The set of $N \times N$ Hermitian or self-adjoint matrices is denoted by $\mathcal{S}^N \subset L(\mathcal{H})$. The notation $\mathcal{S}^N_+ \subset \mathcal{S}^N$ denotes the set of $N \times N$ positive semidefinite (PSD) matrices, and $\mathcal{D}^N \subset \mathcal{S}^N_+$ denotes the set of $N \times N$ density matrices. Additionally, we use $\|\cdot\|$ to represent the ℓ_2 norm of a vector, as well as the spectral or operator norm of a matrix (its largest singular value), and it should be clear from the context which is being used. Let $\mathrm{Tr}[X]$ denote the trace of a matrix X, i.e., the sum of its diagonal terms. Let $X^\top$ and $X^\dagger$ denote the transpose and Hermitian conjugate (or adjoint) of the matrix X, respectively. The notation $A \succcurlyeq B$ or $A - B \succcurlyeq 0$ indicates that $A - B \in \mathcal{S}^N_+$. For a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, we use ∇f and $\nabla^2 f$ to denote its gradient and Hessian, respectively. Let $\frac{\partial f(\cdot)}{\partial x_i}$ denote the partial derivative of f with respect to i^{th} component of the vector $\boldsymbol{x}$. For a multivariate vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, we denote its Jacobian by $J_f(\cdot)$.

Let $\mathcal{H}$ and $\mathcal{H}'$ be Hilbert spaces of dimensions N and N', respectively.

Definition 43 (Linear map) *A map $\Phi : L(\mathcal{H}) \to L(\mathcal{H}')$ is a linear map if the following holds $\forall X, Y \in L(\mathcal{H})$ and $\forall \alpha, \beta \in \mathbb{C}$,*

$$\Phi(\alpha X + \beta Y) = \alpha \Phi(X) + \beta \Phi(Y). \tag{4.3.1}$$

Definition 44 (Adjoint of a linear map) *The adjoint* $\Phi^\dagger : L(\mathcal{H}') \to L(\mathcal{H})$ *of a linear map* $\Phi : L(\mathcal{H}) \to L(\mathcal{H}')$ *is the unique linear map such that*

$$\langle Y, \Phi(X)\rangle = \langle \Phi^\dagger(Y), X\rangle \quad \forall X \in L(\mathcal{H}),\ \forall Y \in L(\mathcal{H}'), \tag{4.3.2}$$

where $\langle C, D\rangle := \mathrm{Tr}[C^\dagger D]$ *is the Hilbert–Schmidt inner product.*

Definition 45 (Hermiticity-preserving linear map) *A linear map* $\Phi : L(\mathcal{H}) \to L(\mathcal{H}')$ *is a Hermiticity-preserving map if* $\Phi(X) \in \mathcal{S}^{N'}$ *for all* $X \in \mathcal{S}^N$*. Equivalently,* Φ *is Hermiticity preserving if and only if* $\Phi(X^\dagger) = \Phi(X)^\dagger$ *for all* $X \in L(\mathcal{H})$*.*

Definition 46 (Choi representation of a linear map) *For every linear map* $\Phi : L(\mathcal{H}) \to L(\mathcal{H}')$*, its Choi representation* Γ^Φ *is defined as*

$$\Gamma^\Phi := \sum_{i,j=0}^{N-1} |i\rangle\langle j| \otimes \Phi(|i\rangle\langle j|). \tag{4.3.3}$$

The operator Γ^Φ *is also known as the Choi operator. Additionally, for every linear map* Φ*, the following holds* $\forall X \in L(\mathcal{H}), Y \in L(\mathcal{H}')$*:*

$$\mathrm{Tr}[Y\Phi(X)] = \mathrm{Tr}[(X^\top \otimes Y)\Gamma^\Phi], \tag{4.3.4}$$

which is a direct consequence of the well known fact that

$$\Phi(X) = \mathrm{Tr}_1[\Gamma^\Phi(X^\top \otimes I)], \tag{4.3.5}$$

with the partial trace over the first factor in the tensor-product space (see Ref. [KW20, Proposition 4.2] for a proof of (4.3.5)*).*

Lemma 47 *Given a linear map* $\Phi : L(\mathcal{H}) \to L(\mathcal{H}')$*, its Choi operator* Γ^Φ *is a Hermitian operator if and only if* Φ *is a Hermiticity-preserving map.*

Proof. First, suppose that Φ is Hermiticity preserving. From (4.3.3), we can write

$$(\Gamma^{\Phi})^{\dagger} = \left(\sum_{i,j=0}^{N-1} |i\rangle\langle j| \otimes \Phi(|i\rangle\langle j|)\right)^{\dagger} = \sum_{i,j=0}^{N-1} |i\rangle\langle j|^{\dagger} \otimes (\Phi(|i\rangle\langle j|))^{\dagger} \tag{4.3.6}$$

$$\overset{(a)}{=} \sum_{i,j=0}^{N-1} |i\rangle\langle j|^{\dagger} \otimes \Phi(|i\rangle\langle j|^{\dagger}) = \sum_{i,j=0}^{N-1} |j\rangle\langle i| \otimes \Phi(|j\rangle\langle i|) = \Gamma^{\Phi}, \tag{4.3.7}$$

where equality (a) follows from Definition 45.

Now suppose that Γ^{Φ} is Hermitian. Then it follows from (4.3.5) that $\Phi(X)$ is Hermitian if X is Hermitian, so that Φ is Hermiticity preserving. ■

We now recall some definitions related to the Lipschitz continuity and smoothness of a function.

Definition 48 (Lipschitz continuity) *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is L-Lipschitz continuous if there exists $L > 0$, such that $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$. We say that L is a Lipschitz constant of f.*

For a univariate function f, suppose that the absolute value of its first derivative on an interval I is bounded from above by a positive real L, i.e., $\forall x \in I : |{}^{df(x)}/_{dx}| \leq L$. Then L is a Lipschitz constant of f.

Lemma 49 (Lipschitz constant for a multivariate function) *For a function $f : \mathbb{R}^n \to \mathbb{R}$ with bounded partial derivatives, the value $L = \sqrt{n} \max_i \{\sup_{\boldsymbol{x}} |\partial f(\boldsymbol{x})/\partial x_i|\}$ is a Lipschitz constant of f.*

Proof. See Appendix 4.A. ■

Lemma 50 (Lipschitz constant for a multivariate vector-valued function) *For a multivariate vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, if each of its components, f_i, is L_i-Lipschitz, then $L = \sqrt{\sum_{i=1}^{m} L_i^2}$ is a Lipschitz constant of f.*

Proof. See Appendix 4.B. ■

Definition 51 (Smoothness) *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is ℓ-smooth if its gradient is ℓ-Lipschitz, i.e., if there exists $\ell > 0$ such that $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq \ell \|\boldsymbol{x} - \boldsymbol{x}'\|$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$.*

As most of the objective functions that we deal with in this paper are non-convex in some parameters, it is vital to focus on local optimality rather than global optimality because finding globally optimal points of a non-convex function is generally intractable. Therefore, the notion of ϵ-stationary points is important for us. Intuitively, a point is ϵ-stationary if the norm of the gradient at that point is very small. Formally, we define an ϵ-stationary point as follows:

Definition 52 (ϵ-stationary point) *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function, and let $\epsilon \geq 0$. A point $\boldsymbol{x} \in \mathbb{R}^n$ is an ϵ-stationary point of f if $\|\nabla f(\boldsymbol{x})\| \leq \epsilon$.*

The above definition applies to first-order stationary points. This definition is important because we use inexact first-order solvers that converge to approximate first-order stationary points, and such a definition acts as a stopping criterion.

Definition 53 (Polyak–Łojasiewicz (PL) Inequality) *A function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies the PL inequality if, for some $\mu > 0$, the following holds for all $\boldsymbol{x} \in \mathbb{R}^n$:*

$$\frac{1}{2} \|f(\boldsymbol{x})\|^2 \geq \mu(f(\boldsymbol{x}) - f^*), \tag{4.3.8}$$

where f^* is the globally optimal value of f.

In other words, the above inequality implies that every stationary point is a global minimum.

4.3.1 Semidefinite Programming

In this section, we recall some basic aspects of semidefinite programming [BV04, Wat18]. A semidefinite program is an optimization problem for which the goal is to optimize a linear function over the intersection of the positive semidefinite cone with an affine space. SDPs extend linear programs (LPs), such that the vector inequalities of LPs are generalized to matrix inequalities.

To begin with, recall the standard or canonical form of an SDP [BV04]:

$$\begin{aligned} &\sup_{X \geqslant 0} \quad \mathrm{Tr}[CX] \\ &\text{subject to} \quad \mathrm{Tr}[A_i X] \leq b_i; \quad \forall i \in [M], \end{aligned} \tag{4.3.9}$$

where $C, A_1, \ldots, A_M \in \mathcal{S}^N$ and $b_1, \ldots, b_M \in \mathbb{R}$. The standard form of an SDP is widely known for designing approximation algorithms for combinatorial optimization problems. A more general form of an SDP, as considered in [Wat18], is as follows:

$$\begin{aligned} p^* &:= \sup_{X \geqslant 0} \quad \mathrm{Tr}[CX] \\ &\text{subject to} \quad \Phi(X) \leqslant B, \end{aligned} \tag{4.3.10}$$

where $C \in \mathcal{S}^N$, the map Φ is Hermiticity preserving (see Definition 4.3.1), $B \in \mathcal{S}^M$, and p^* is the optimal value of the program (4.3.10). The aforementioned form is known as the primal form of an SDP, and the corresponding dual form is given as

$$\begin{aligned} d^* &:= \inf_{Y \geqslant 0} \quad \mathrm{Tr}[BY] \\ &\text{subject to} \quad \Phi^\dagger(Y) \geqslant C, \end{aligned} \tag{4.3.11}$$

where $Y \in S_+^M$, the map $\Phi^\dagger$ is the adjoint of Φ (see Definition 44), and d^* is the optimal value of the program (4.3.11).

The duality theorem of SDPs states that if both the primal and dual programs have feasible solutions, then the optimum of the primal program is bounded from above by

the optimum of the dual program. Under a very mild condition that the primal has a feasible solution and the dual has a strictly feasible solution (or vice versa), strong duality holds; i.e., the duality gap (difference between p^* and d^*) is closed. In the optimization literature, this condition is well known as Slater's condition (see Theorem 1.18 of [Wat18]). Throughout this paper, we assume that strong duality holds.

4.3.2 Variational Quantum Algorithms and Parameter-Shift Rule

Variational quantum algorithms are hybrid quantum-classical algorithms, designed for solving optimization tasks with an objective function of the following form:

$$\mathcal{F}(\rho) = \sum_k g_k(\mathrm{Tr}[H_k\rho]), \tag{4.3.12}$$

where $\rho \in \mathcal{D}^N$ is a density operator, $\{H_k\}_k$ is a set of problem-specific Hermitian operators, i.e., $H_k \in \mathcal{S}^N$ for all k, and $\{g_k\}_k$ is a problem-specific set of functions [CAB+21, BCLK+22]. Additionally, each g_k is a function of the expectation value of H_k with respect to ρ. The corresponding optimization problem is as follows:

$$\min_{\rho\in\mathcal{D}^N} \mathcal{F}(\rho). \tag{4.3.13}$$

When the dimension N is large, the evaluation of the expectation values of Hermitian operators with respect to ρ is computationally intractable using classical algorithms. VQAs provide a quantum advantage because these hybrid algorithms attempt to circumvent this dimensionality problem by evaluating the expectation values of Hermitian operators using a quantum computer. Specifically, these algorithms utilize a parameterized quantum circuit to explore a problem-specific subspace of density operators and evaluate the expectation values with respect to these density operators. We discuss more about

the quantum advantage of VQAs at the end of this section. First, let us discuss what we mean by a parameterized quantum circuit and how this circuit prepares a parameterized quantum state.

Parameterization: Let $|\mathbf{0}\rangle_{RS}$ denote the all-zeros state of systems R and S, each of which consists of n qubits. Let $\rho_S \in \mathcal{D}^{2^n}$, and let U^{ρ}_{RS} be a quantum circuit that prepares a purification $|\psi\rangle_{RS}$ of ρ_S when U^{ρ}_{RS} is applied to the initial state $|\mathbf{0}\rangle_{RS}$. Here, the subscripts S and R are used to denote the system of interest and a reference system, respectively. VQAs simulate the space of density operators by parameterizing this quantum circuit as $U_{RS}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_r)^\top \in [0, 2\pi]^r$. Specifically, a VQA applies a parameterized quantum circuit $U_{RS}(\boldsymbol{\theta})$ to the initial pure state $|\mathbf{0}\rangle_{RS}$ to generate a parameterized pure state

$$|\psi(\boldsymbol{\theta})\rangle_{RS} := U_{RS}(\boldsymbol{\theta})|\mathbf{0}\rangle_{RS} \in (\mathbb{C}^2)^{\otimes 2n}. \tag{4.3.14}$$

Let $\rho(\boldsymbol{\theta})_S := \mathrm{Tr}_R[|\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})|_{RS}]$ denote the reduced density operator of $|\psi(\boldsymbol{\theta})\rangle_{RS}$. By using the fact that $\mathrm{Tr}[H_k\rho(\boldsymbol{\theta})] = \langle\psi(\boldsymbol{\theta})|_{RS}(I_R \otimes H_k)|\psi(\boldsymbol{\theta})\rangle_{RS}$ and under the assumption that the parameterized circuit $U_{RS}(\boldsymbol{\theta})$ is fully expressive, the objective function in (4.3.12) and its associated optimization problem (4.3.13) can be written in terms of the parameter $\boldsymbol{\theta}$ as follows:

$$\mathcal{F}(\boldsymbol{\theta}) := \sum_k g_k(\langle\psi(\boldsymbol{\theta})|_{RS}(I_R \otimes H_k)|\psi(\boldsymbol{\theta})\rangle_{RS}) \tag{4.3.15}$$

$$= \sum_k g_k(\langle I_R \otimes H_k\rangle_{\boldsymbol{\theta}}), \tag{4.3.16}$$

$$\boldsymbol{\theta}^* := \arg\min_{\boldsymbol{\theta}\in[0,2\pi]^r} \mathcal{F}(\boldsymbol{\theta}), \tag{4.3.17}$$

where it is implicit that H_k acts on system S.

Assumption 2 *We assume that the objective function in* (4.3.16) *is 'faithful,' which means that the minimum of this objective function corresponds to the optimal value of the problem* (4.3.13).

A parameterized quantum circuit $U(\boldsymbol{\theta})$ is also known as a variational ansatz, and its choice plays an important role in obtaining an approximation of the optimal value. Throughout this paper, we use the terms "parameterized quantum circuit" and "variational ansatz" interchangeably. Furthermore, there are problem-specific ansatzes as well as problem-independent ansatzes. For our case, we use a problem-independent ansatz having the following form:

$$U(\boldsymbol{\theta}) = U_r(\theta_r)U_{r-1}(\theta_{r-1})\cdots U_1(\theta_1), \tag{4.3.18}$$

where each unitary $U_j(\theta_j)$ is written as

$$U_j(\theta_j) = e^{-i\theta_j H_j}W_j, \tag{4.3.19}$$

with W_j as an unparameterized unitary.

Assumption 3 *We assume that the number of parameters, r, of a parameterized circuit is $O(\text{poly}(n))$.*

The above assumption is natural in the context of variational quantum algorithms, as we only have access to quantum circuits with short depth.

Each Hermitian operator H_k is arbitrary. In general, we can express a Hermitian operator as a weighted sum of tensor products of Pauli operators, i.e.,

$$H = \sum_{i=1}^{p} w_i\, \sigma_{i,1} \otimes \cdots \otimes \sigma_{i,n}, \tag{4.3.20}$$

where $w_i \in \mathbb{R}$, $\sigma_{i,j} \in \{I, \sigma_x, \sigma_y, \sigma_z\}$, and σ_x, σ_y, and σ_z are the Pauli operators. From (4.3.20) we see that the expectation value of an arbitrary Hermitian operator is equal to a linear combination of the expectation values of each tensor product of Pauli operators. However, in general, this linear combination may contain many terms.

Assumption 4 *We assume that the number of terms in* (4.3.20) *is polynomial in* n*; i.e.,* $p = O(\text{poly}(n))$.

The above three assumptions are standard in the literature on variational quantum algorithms. Overall, VQAs use a quantum circuit with parameter $\boldsymbol{\theta}$ to estimate the expectation value of a given Hermitian matrix, and they utilize a classical optimizer to solve the optimization problem $\min_{\boldsymbol{\theta}\in[0,2\pi]^r} \mathcal{F}(\boldsymbol{\theta})$. In each round, a VQA updates the parameters of the quantum circuit according to a classical optimization algorithm. We can update these parameters according to gradient-free approaches that include Nelder–Mead [NM65], Simultaneous Perturbation Stochastic Approximation (SPSA) [Spa92], and Particle Swarm Optimization [KE95]. The main drawbacks of such gradient-free methods include slower convergence and less robustness against noise. On the contrary, if the evaluation of the gradient of a given objective function is not computationally expensive, then first-order methods like gradient descent are more suitable.

VQAs have an advantage as they do not use a classical method to evaluate gradients. Instead, they evaluate the partial derivatives of $\mathcal{F}(\boldsymbol{\theta})$ with respect to each parameter using the same quantum circuit but with shifted parameters. It is possible to evaluate the partial derivatives on a quantum computer using the parameter-shift rule [LYPS17, MNKF18, SBG+19]. Formally, we state the parameter-shift rule for the evaluation of the partial derivatives of $\langle H\rangle_{\boldsymbol{\theta}}$ with respect to its j^{th} component, i.e., θ_j, as

$$\frac{\partial\langle H\rangle_{\boldsymbol{\theta}}}{\partial\theta_j} = \frac{1}{2}\left(\langle H\rangle_{\boldsymbol{\theta}+(\pi/2)\hat{\boldsymbol{e}}_j} - \langle H\rangle_{\boldsymbol{\theta}-(\pi/2)\hat{\boldsymbol{e}}_j}\right), \tag{4.3.21}$$

where $\hat{\boldsymbol{e}}_j$ is a unit vector with 1 as its j^{th} element and 0 otherwise.

From (4.3.21), we note that for the exact evaluation of the partial derivatives of $\langle H\rangle_{\boldsymbol{\theta}}$ at a parameter value $\boldsymbol{\theta}$, we need to compute the expectation values of this operator at the

shifted parameters exactly. However, the exact evaluation of the expectation value of an operator requires an infinite number of measurements on a quantum circuit. In reality, we perform only a limited number of measurements and then take the average of those values. Therefore, it is important to have an unbiased estimator of the expectation value. We assume that this is the case for all of our algorithms. Such an assumption is standard in the literature, as it acts as a good starting point for understanding the nature of the algorithms under such settings.

Unbiased Estimator: First, we consider a simple objective function consisting of a single expectation value term, i.e.,

$$\mathcal{F}(\boldsymbol{\theta}) = \langle H \rangle_{\boldsymbol{\theta}}. \tag{4.3.22}$$

Now, we define a k-sample mean unbiased estimator of this expectation value as follows:

Definition 54 (k-sample mean unbiased estimator of expectation value) *Given a quantum circuit $U(\boldsymbol{\theta})$ with parameter $\boldsymbol{\theta} \in [0, 2\pi]^r$, we define $u_k^H(\boldsymbol{\theta})$ as an average of k measurements of the observable H with respect to the pure state $U(\boldsymbol{\theta})|\mathbf{0}\rangle$. It is a k-sample mean unbiased estimator of (4.3.22) if the following holds:*

$$\mathbb{E}[u_k^H(\boldsymbol{\theta})] = \langle H \rangle_{\boldsymbol{\theta}}. \tag{4.3.23}$$

Next, we define an unbiased estimator of the partial derivatives of $\langle H \rangle_{\boldsymbol{\theta}}$. According to the parameter-shift rule in (4.3.21), the partial derivative of the expectation values of a Hermitian operator is a linear combination of its expectation values with shifted parameters. Setting

$$g_j^{H,k}(\boldsymbol{\theta}) \coloneqq \frac{1}{2}\left(u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j) - u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)\right), \tag{4.3.24}$$

it follows that $g_j^{H,k}(\boldsymbol{\theta})$ is an unbiased estimator for that partial derivative because

$$\mathbb{E}\Big[g_j^{H,k}(\boldsymbol{\theta})\Big] = \mathbb{E}\left[\frac{1}{2}\Big(u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j) - u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)\Big)\right] \tag{4.3.25}$$

$$= \frac{1}{2}\Big(\mathbb{E}[u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j)] - \mathbb{E}[u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)]\Big) \tag{4.3.26}$$

$$= \frac{1}{2}\Big(\langle H\rangle_{\boldsymbol{\theta}+(\pi/2)\hat{\boldsymbol{e}}_j} - \langle H\rangle_{\boldsymbol{\theta}-(\pi/2)\hat{\boldsymbol{e}}_j}\Big) \tag{4.3.27}$$

$$= \frac{\partial\langle H\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}. \tag{4.3.28}$$

We denote an unbiased estimator of the full gradient, i.e., $\nabla_\theta\langle H\rangle_{\boldsymbol{\theta}}$, as $g^{H,k}(\boldsymbol{\theta})$. Furthermore, we evaluate the mean square error of $g^{H,k}(\boldsymbol{\theta})$ as follows,

$$\mathbb{E}\Big[\Big\|g^{H,k}(\boldsymbol{\theta}) - \nabla_\theta\langle H\rangle_{\boldsymbol{\theta}}\Big\|^2\Big]$$

$$= \mathbb{E}\left[\sum_{j=1}^{r}\left(g_j^{H,k}(\boldsymbol{\theta}) - \frac{\partial\langle H\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}\right)^2\right] \tag{4.3.29}$$

$$= \sum_{j=1}^{r}\mathbb{E}\left[\left(g_j^{H,k}(\boldsymbol{\theta}) - \frac{\partial\langle H\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}\right)^2\right] \tag{4.3.30}$$

$$= \sum_{j=1}^{r}\mathrm{Var}\Big(g_j^{H,k}(\boldsymbol{\theta})\Big) \tag{4.3.31}$$

$$= \sum_{j=1}^{r}\frac{1}{4}\mathrm{Var}\Big(u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j) - u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)\Big) \tag{4.3.32}$$

$$= \frac{1}{4}\sum_{j=1}^{r}\mathrm{Var}\Big(u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j)\Big) + \mathrm{Var}\Big(u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)\Big) \tag{4.3.33}$$

$$= \frac{1}{4}\sum_{j=1}^{r}\frac{1}{k}\mathrm{Var}\Big(u_1^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j)\Big) + \frac{1}{k}\mathrm{Var}\Big(u_1^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)\Big), \tag{4.3.34}$$

where we denote the variance of a random variable X as $\mathrm{Var}(X)$. The fourth equality follows from the definition of $g_j^{H,k}(\boldsymbol{\theta})$, given by (4.3.24). The fifth equality uses the fact that $\mathrm{Var}(X - Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$ if X and Y are independent random variables. For our case, the sample means $u_k^H(\boldsymbol{\theta} + (\pi/2)\hat{\boldsymbol{e}}_j)$ and $u_k^H(\boldsymbol{\theta} - (\pi/2)\hat{\boldsymbol{e}}_j)$ are independent because they are computed using two different quantum circuit evaluations. For computing gradients in our algorithms, we take a sufficient number of samples, k, such that the mean square

error is very small throughout the paper. This implies that the stochastic gradient ($g^{H,k}(\boldsymbol{\theta})$) is almost equal to the exact gradient ($\nabla_\theta \langle H \rangle_{\boldsymbol{\theta}}$).

Potential quantum advantage of a VQA when N is large: According to (4.3.20), we consider a Hermitian operator H consisting of a sum of p weighted Pauli strings where $p = O(\text{poly}(n))$ (Assumption 4). We can measure a given Pauli string P_i of arbitrary size with respect to a given quantum state in constant time. If the desired precision of the expectation value's estimate is ϵ, then we need to make $O(1/\epsilon^2)$ repetitions of the procedure. Here the procedure consists of preparing the quantum state and then measuring the expectation value. As we have p Pauli strings, overall we need no more than $O(\max_i |w_i|^2 \, p/\epsilon^2)$ repetitions to estimate the expectation value of H to precision ϵ. In contrast, if the evaluation of the expectation value of a Pauli string P_i with respect to a general quantum state is conducted using known classical algorithms, it appears that we need $O(2^n)$ time, as well as space. This is because the size of P_i is $2^n \times 2^n$, and the same goes for the size of the matrix needed to represent the quantum state. Therefore, if all the above mentioned assumptions hold, then we have an exponential advantage over the best known classical algorithms for estimating the expectation value of a Hermitian operator. A similar argument can be provided for estimating gradients because the partial derivatives of the expectation value with respect to quantum circuit parameters depend on the expectation value evaluated on shifted parameters.

4.4 Variational Quantum Algorithms for SDPs

4.4.1 General Form (GF) of SDPs

In this section, we consider the general form of an SDP, as given in (4.3.10). We write it concisely as follows:

$$p^* = \sup_{X \geqslant 0} \{\mathrm{Tr}[CX] : \Phi(X) \leqslant B\}. \tag{4.4.1}$$

Next, we modify the above formulation as follows:

$$\begin{aligned}
p^* &= \sup_{X \geqslant 0} \{\mathrm{Tr}[CX] : \Phi(X) \leqslant B\} && (4.4.2)\\
&\overset{(a)}{=} \sup_{X \geqslant 0} \left\{ \mathrm{Tr}[CX] + \inf_{Y \geqslant 0} \{\mathrm{Tr}[(B - \Phi(X))Y]\} \right\} && (4.4.3)\\
&\overset{(b)}{=} \sup_{X \geqslant 0} \inf_{Y \geqslant 0} \{\mathrm{Tr}[CX] + \mathrm{Tr}[BY] - \mathrm{Tr}[Y\Phi(X)]\} && (4.4.4)\\
&\overset{(c)}{=} \sup_{X \geqslant 0} \inf_{Y \geqslant 0} \left\{ \mathrm{Tr}[CX] + \mathrm{Tr}[BY] - \mathrm{Tr}[(X^\top \otimes Y)\Gamma^\Phi] \right\} && (4.4.5)\\
&= \sup_{X \geqslant 0} \inf_{Y \geqslant 0} \left\{ \mathrm{Tr}[C^\top X^\top] + \mathrm{Tr}[BY] - \mathrm{Tr}[(X^\top \otimes Y)\Gamma^\Phi] \right\} && (4.4.6)\\
&= \sup_{X \geqslant 0} \inf_{Y \geqslant 0} \left\{ \mathrm{Tr}[C^\top X] + \mathrm{Tr}[BY] - \mathrm{Tr}[(X \otimes Y)\Gamma^\Phi] \right\} && (4.4.7)\\
&\overset{(d)}{=} \sup_{\substack{\rho \in \mathcal{D}^N, \\ \lambda \geq 0}} \inf_{\substack{\sigma \in \mathcal{D}^M, \\ \mu \geq 0}} \left\{ \lambda \,\mathrm{Tr}[C^\top \rho] + \mu \,\mathrm{Tr}[B\sigma] - \lambda\mu \,\mathrm{Tr}[\Gamma^\Phi(\rho \otimes \sigma)] \right\}. && (4.4.8)
\end{aligned}$$

Equality (a) follows due to the equivalence between both problems. If we pick a primal PSD variable X that does not satisfy the constraint, i.e., $\Phi(X) \not\leqslant B$, then the inner minimization results in the value $-\infty$. This is because, in such a case, there exists at least one negative eigenvalue of $B - \Phi(X)$. If we set $Y = s|e_i\rangle\langle e_i|$, where $|e_i\rangle$ is a unit vector in the negative eigenspace corresponding to that eigenvalue, then, due to the inner minimization, we can take the limit $s \to \infty$. This in turn implies that $\inf_{Y \geqslant 0} \{\mathrm{Tr}[(B - \Phi(X))Y]\} = -\infty$. In other words, the inner minimization forces the outer maximization to pick a feasible X

and imposes an infinite penalty if chosen otherwise.

Equality (b) follows by taking the infimum outside. This formulation is the Lagrangian of the original primal problem (4.2.1), where

$$\mathcal{L}(X, Y) := \mathrm{Tr}[CX] + \mathrm{Tr}[BY] - \mathrm{Tr}[Y\Phi(X)] \tag{4.4.9}$$

is a Lagrangian and Y is a dual PSD variable.

Equality (c) follows from the definition of the Choi representation of the linear map Φ (see Definition 46), where Γ^{Φ} is the Choi operator of Φ. According to Lemma 47, the Choi operator Γ^{Φ} is Hermitian.

Equality (d) follows from the substitution: $X = \lambda\rho$ and $Y = \mu\sigma$, where $\rho \in \mathcal{D}^N, \sigma \in \mathcal{D}^M$, $\lambda = \mathrm{Tr}[X]$, and $\mu = \mathrm{Tr}[Y]$.

Now, we are interested in solving the following unconstrained optimization problem, i.e., the equality (d):

$$p^* = \sup_{\substack{\rho\in\mathcal{D}^N,\\ \lambda\geq 0}} \inf_{\substack{\sigma\in\mathcal{D}^M,\\ \mu\geq 0}} \left\{\lambda\,\mathrm{Tr}[C^\top\rho] + \mu\,\mathrm{Tr}[B\sigma] - \lambda\mu\,\mathrm{Tr}[\Gamma^{\Phi}(\rho\otimes\sigma)]\right\}. \tag{4.4.10}$$

This optimization problem is now expressed in terms of the expectation values of the Hermitian operators $C^\top$, B, and Γ^{Φ} with respect to the density operators ρ, σ, and $\rho\otimes\sigma$, respectively.

Variational Quantum Algorithm for SDPs in GF

As stated earlier, when the dimension N of the Hermitian operators is large, solving the problem (4.4.10) is generally intractable using a gradient-based classical algorithm. Therefore, we propose a variational quantum algorithm for the optimization problem (4.4.10).

First, we introduce a parameterization of the density operators, i.e., ρ and σ, by using parameterized quantum circuits. Second, we optimize the modified objective function of the problem (4.4.10) over the parameters of those quantum circuits using our variational quantum algorithm.

Parameterization: Let $\rho_{S_1}(\boldsymbol{\theta}_1)$ be the density operator prepared by first applying the quantum circuit $U_{R_1S_1}(\boldsymbol{\theta}_1)$ to the all-zeros state of the quantum system R_1S_1 and then tracing out the system R_1. Similarly, let $\sigma_{S_2}(\boldsymbol{\theta}_2)$ be the density operator prepared by first applying the quantum circuit $U_{R_2S_2}(\boldsymbol{\theta}_2)$ to the all-zeros state of the quantum system R_2S_2 and then tracing out the system R_2. Here, $\boldsymbol{\theta}_1 \in [0, 2\pi]^{r_1}$ and $\boldsymbol{\theta}_2 \in [0, 2\pi]^{r_2}$. Also, we set $r_1, r_2 = O(\text{poly}(n))$. Defining

$$|\psi(\boldsymbol{\theta}_1)\rangle_{R_1S_1} \coloneqq U_{R_1S_1}(\boldsymbol{\theta}_1)|\mathbf{0}\rangle_{R_1S_1}, \tag{4.4.11}$$

$$|\varphi(\boldsymbol{\theta}_2)\rangle_{R_2S_2} \coloneqq U_{R_2S_2}(\boldsymbol{\theta}_2)|\mathbf{0}\rangle_{R_2S_2}, \tag{4.4.12}$$

we have that

$$\rho_{S_1}(\boldsymbol{\theta}_1) = \text{Tr}_{R_1}[|\psi(\boldsymbol{\theta}_1)\rangle\langle\psi(\boldsymbol{\theta}_1)|_{R_1S_1}], \tag{4.4.13}$$

$$\sigma_{S_2}(\boldsymbol{\theta}_2) = \text{Tr}_{R_2}[|\varphi(\boldsymbol{\theta}_2)\rangle\langle\varphi(\boldsymbol{\theta}_2)|_{R_2S_2}]. \tag{4.4.14}$$

Furthermore, the parameterized quantum circuits, i.e., $U_{R_1S_1}(\boldsymbol{\theta}_1)$ and $U_{R_2S_2}(\boldsymbol{\theta}_2)$, are of the form shown in (4.3.18). As these quantum circuits generate purifications of density operators, the following equalities hold:

$$\text{Tr}[C^\top \rho_{S_1}(\boldsymbol{\theta}_1)] = \langle\psi(\boldsymbol{\theta}_1)|_{R_1S_1} \left(I_{R_1} \otimes C_{S_1}^\top\right) |\psi(\boldsymbol{\theta}_1)\rangle_{R_1S_1} \tag{4.4.15}$$

$$= \langle I \otimes C^\top \rangle_{\boldsymbol{\theta}_1}, \tag{4.4.16}$$

$$\text{Tr}[B\sigma_{S_2}(\boldsymbol{\theta}_2)] = \langle\varphi(\boldsymbol{\theta}_2)|_{R_2S_2} \left(I_{R_2} \otimes B_{S_2}\right) |\varphi(\boldsymbol{\theta}_2)\rangle_{R_2S_2} \tag{4.4.17}$$

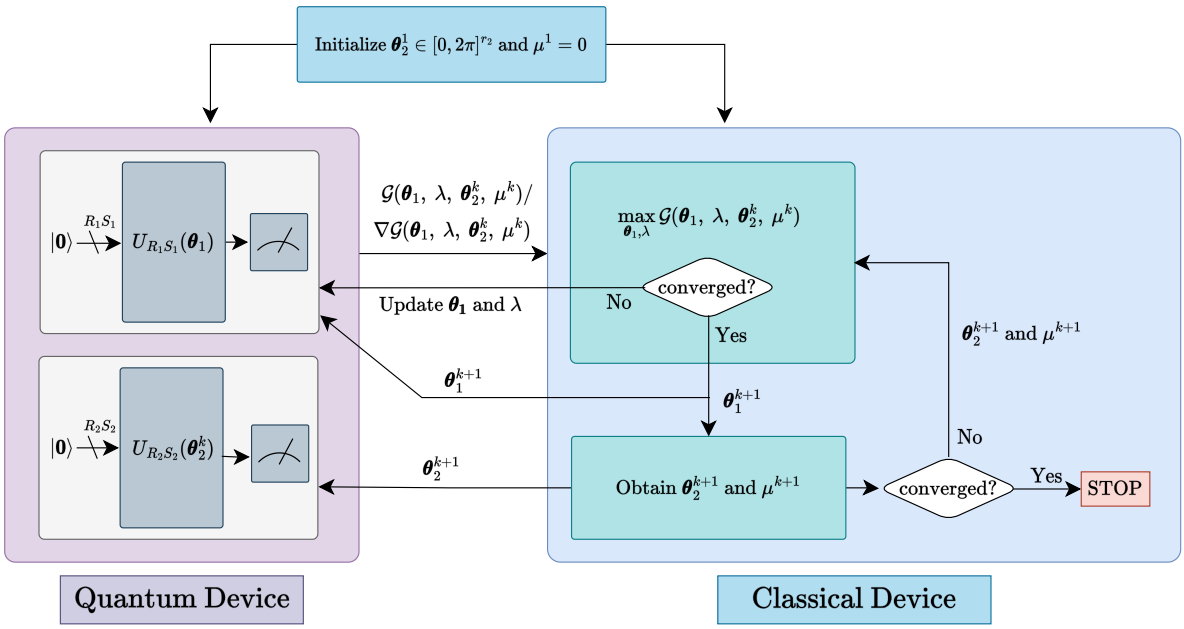


Figure 4.1: This figure depicts iVQAGF algorithm where we utilize two parameterized quantum circuits, i.e., $U_{R_1S_1}(\boldsymbol{\theta}_1)$ and $U_{R_2S_2}(\boldsymbol{\theta}_2)$.

$$= \langle I \otimes B \rangle_{\boldsymbol{\theta}_2}, \tag{4.4.18}$$

$$\begin{aligned} \mathrm{Tr}[\Gamma^{\Phi}(\rho_{S_1}(\boldsymbol{\theta}_1) \otimes \sigma_{S_2}(\boldsymbol{\theta}_2))] &= \langle \phi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)|_{R_1R_2S_1S_2} \left(I_{R_1} \otimes I_{R_2} \otimes \Gamma^{\Phi}_{S_1S_2} \right) |\phi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\rangle_{R_1R_2S_1S_2} \\ &= \langle I \otimes I \otimes \Gamma^{\Phi} \rangle_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2}, \end{aligned} \tag{4.4.19}$$

where

$$|\phi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\rangle_{R_1R_2S_1S_2} := |\psi(\boldsymbol{\theta}_1)\rangle_{R_1S_1} \otimes |\varphi(\boldsymbol{\theta}_2)\rangle_{R_2S_2}. \tag{4.4.20}$$

Due to the parameterization of the density operators $\rho_{S_1}(\boldsymbol{\theta}_1)$ and $\sigma_{S_2}(\boldsymbol{\theta}_2)$, the problem (4.4.10) transforms into the following optimization problem:

$$p^* := \sup_{\substack{\boldsymbol{\theta}_1 \in [0,2\pi]^{r_1}, \\ \lambda \geq 0}} \inf_{\substack{\boldsymbol{\theta}_2 \in [0,2\pi]^{r_2}, \\ \mu \geq 0}} \mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu), \tag{4.4.21}$$

where

$$\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu) := \lambda \langle I \otimes C^{\top} \rangle_{\boldsymbol{\theta}_1} + \mu \langle I \otimes B \rangle_{\boldsymbol{\theta}_2} - \lambda\mu \langle I \otimes I \otimes \Gamma^{\Phi} \rangle_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2} \tag{4.4.22}$$

and we optimize over the space of quantum circuit parameters $\boldsymbol{\theta}_1 \in [0, 2\pi]^{r_1}$ and $\boldsymbol{\theta}_2 \in [0, 2\pi]^{r_2}$. As mentioned before, we assume that the objective function $\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ is faithful, which means that the global optimal value of the optimization problem (4.4.21) is equal to p^*. Additionally, the objective function of (4.4.21) is generally non-convex as a function of the quantum circuit parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Hence, the max-min problem (4.4.21) is a non-convex–non-concave optimization problem. Due to the fact that obtaining a globally optimal point of a non-convex–non-concave function is generally NP-hard [DDG+22, Section 2.1], we focus on finding first order ϵ-stationary points of (4.4.21). As a global optimal point is also a stationary point, if we use techniques to initialize quantum circuit parameters according to the problem at hand such that these parameters lie in the vicinity of a global optimal point, then our algorithm will converge to that point.

Definition 55 (First order ϵ-stationary point of (4.4.21)) *A point $(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ is a first order ϵ-stationary point of (4.4.21) if and only if $\|\nabla \mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)\| \leq \epsilon$.*

We propose a variational quantum algorithm for obtaining a first order ϵ-stationary point of (4.4.21). We call this algorithm *inexact Variational Quantum Algorithm for General Form* (iVQAGF), and the pseudocode for this algorithm is provided in Algorithm 5. It is an inexact version because we solve the subproblem involving a maximization (Step 4) to approximate stationary points instead of solving it until global optimality is reached, due to the non-convex nature of the objective function in terms of the quantum circuit parameters. Furthermore, iVQAGF is a hybrid quantum-classical algorithm, as we have two parameterized quantum circuits for estimating expectation values of Hermitian operators at any given step and a classical optimizer that updates the parameters of these quantum circuits as per the algorithm (see Figure 4.1).

At any step of the algorithm, the expectation value $\langle H \rangle_{\boldsymbol{\theta}}$ of a Hermitian operator H is

Algorithm 5: `iVQAGF`$(\Gamma^{\Phi}, C, B, \eta_1, \eta_2, \epsilon)$

Input: Hermitian operators Γ^{Φ}, C, and B, learning rates $\eta_1, \eta_2 > 0$, precision $\epsilon > 0$.

Initialization: $\boldsymbol{\theta}_2^1 \in [0, 2\pi]^{r_2}$, $\mu^1 = 0$.

 # For any step, expectation values of observables and their gradients are evaluated using parameterized quantum circuits.

for $k = 1, 2, \ldots,$ **do**

Maximize $\mathcal{G}(\cdot,\ \cdot,\ \boldsymbol{\theta}_2^k,\ \mu^k)$ using a first order method such as gradient descent, where $\boldsymbol{\theta}_2^k,\ \mu^k$ are fixed, to obtain $(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1})$ such that the following holds:

$$\left\|\nabla\mathcal{G}(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1},\ \boldsymbol{\theta}_2^k,\ \mu^k)\right\| \leq \epsilon.$$

$\mu^{k+1} = \mu^k - \eta_1 \nabla_{\mu}\mathcal{G}(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1},\ \boldsymbol{\theta}_2^k,\ \mu^k)$

$\boldsymbol{\theta}_2^{k+1} = \boldsymbol{\theta}_2^k - \eta_2 \nabla_{\boldsymbol{\theta}_2}\mathcal{G}(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1},\ \boldsymbol{\theta}_2^k,\ \mu^k)$

if $\left\|\nabla\mathcal{G}(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1},\ \boldsymbol{\theta}_2^{k+1},\ \mu^{k+1})\right\| \leq \epsilon$ **then**

STOP and return $\mathcal{G}(\boldsymbol{\theta}_1^{k+1},\ \lambda^{k+1},\ \boldsymbol{\theta}_2^{k+1},\ \mu^{k+1})$

end if

end for

evaluated using a quantum circuit with parameter $\boldsymbol{\theta}$. Moreover, the partial derivatives of $\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ with respect to the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ depend on the partial derivatives of the expectation values of the Hermitian operators $I \otimes C^{\top}$, $I \otimes B$, and $I \otimes I \otimes \Gamma^{\Phi}$ with respect to those parameters. We evaluate the partial derivative of the expectation value of a Hermitian operator with respect to the quantum circuit parameters using the parameter-

shift rule (see (4.3.21)). We do not explicitly mention these quantum-circuit calls in the main algorithm.

Unbiased Estimators: Due to the fact that the evaluation of the expectation value of a Hermitian operator using a quantum circuit is stochastic in nature, we use unbiased estimators of expectation values and their corresponding partial derivatives (see (4.3.23) and (4.3.25)). Now, from (4.4.22), we can see that $\nabla_{\theta_1}\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ is a linear combination of $\nabla_{\theta_1}\langle I \otimes C^\top\rangle_{\theta_1}$ and $\nabla_{\theta_1}\langle I \otimes I \otimes \Gamma^\Phi\rangle_{\theta_1,\theta_2}$. Therefore, the unbiased estimator of $\nabla_{\theta_1}\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ is also a linear combination of the unbiased estimators of $\nabla_{\theta_1}\langle I \otimes C^\top\rangle_{\theta_1}$ and $\nabla_{\theta_1}\langle I \otimes I \otimes \Gamma^\Phi\rangle_{\theta_1,\theta_2}$. Similarly, the unbiased estimator of $\nabla_{\theta_2}\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ is a linear combination of the unbiased estimators of $\nabla_{\theta_2}\langle I \otimes B\rangle_{\theta_2}$ and $\nabla_{\theta_2}\langle I \otimes I \otimes \Gamma^\Phi\rangle_{\theta_1,\theta_2}$. Also, the unbiased estimators of $\nabla_\lambda\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ and $\nabla_\mu\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ depend on a linear combination of the unbiased estimators of expectation values of Hermitian operators, i.e., $\langle I \otimes C^\top\rangle_{\theta_1}$, $\langle I \otimes B\rangle_{\theta_2}$, and $\langle I \otimes I \otimes \Gamma^\Phi\rangle_{\theta_1,\theta_2}$. Finally, as we have the unbiased estimators of all these partial derivatives, we have an unbiased estimator of the full gradient $\nabla\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$ of the function $\mathcal{G}(\boldsymbol{\theta}_1, \lambda, \boldsymbol{\theta}_2, \mu)$.

4.4.2 Equality Constrained Standard Form (ECSF) of SDPs

In this section, we shift our focus to the standard form of SDPs. Due to their specific problem structure and the aforementioned assumption of it being weakly constrained (i.e, $N \gg M$), the design of more sophisticated variational quantum algorithms for solving them is possible. Moreover, we establish a convergence rate for one of the algorithms.

We consider the standard form of SDPs as given in (4.3.9). Here, the Hermiticity-preserving map Φ and the Hermitian operator B have a diagonal form:

$$\Phi(X) = \operatorname{diag}\left(\operatorname{Tr}[A_1 X], \ldots, \operatorname{Tr}[A_M X]\right), \tag{4.4.23}$$

$$B = \operatorname{diag}(b_1, \ldots, b_M), \tag{4.4.24}$$

where $A_1, \ldots, A_M \in \mathcal{S}^N$ and $b_1, \ldots, b_M \in \mathbb{R}$.

The equality constrained primal SDP is given as follows:

$$p^* = \sup_{X \succcurlyeq 0} \{\operatorname{Tr}[CX] : \boldsymbol{\Phi}(X) = \boldsymbol{b}\}, \tag{4.4.25}$$

where $\boldsymbol{b} = (b_1, \ldots, b_M)^\top$ and $\boldsymbol{\Phi}$ is the vector form of the original linear map, i.e., $\boldsymbol{\Phi}(X) = (\operatorname{Tr}[A_1 X], \ldots, \operatorname{Tr}[A_M X])^\top$. Taking this formulation into account, we write the optimization over the Lagrangian $\mathcal{L}(X, Y)$ in (4.4.9) as

$$p^* = \sup_{X \succcurlyeq 0} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \mathcal{L}(X, \boldsymbol{y}), \tag{4.4.26}$$

where

$$\mathcal{L}(X, \boldsymbol{y}) \coloneqq \operatorname{Tr}[CX] + \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{\Phi}(X)) \tag{4.4.27}$$

and $\boldsymbol{y} \in \mathbb{R}^M$ is the dual vector of the Lagrangian $\mathcal{L}(X, \boldsymbol{y})$. As discussed earlier, the inner minimization with respect to $\boldsymbol{y}$ results in $-\infty$ if X violates the constraint in (4.4.25). On the contrary, if there exists a PSD operator X that satisfies the constraints, then (4.4.26) reduces to $\sup_{X \succcurlyeq 0} \operatorname{Tr}[CX]$. This is because $\inf_{\boldsymbol{y} \in \mathbb{R}^M} \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{\Phi}(X)) = 0$ in this case. Hence, there is an equivalence between (4.4.25) and (4.4.26), as indicated previously.

For the equality constrained problem (4.4.25), we consider the Augmented Lagrangian $\mathcal{L}_c(X, \boldsymbol{y})$ as the objective function, which consists of a quadratic penalty term $\frac{c}{2} \|\boldsymbol{b} - \boldsymbol{\Phi}(X)\|^2$ in addition to the terms of the original Lagrangian $\mathcal{L}(X, \boldsymbol{y})$. Therefore, the optimization problem with the modified objective function can be written as follows:

$$p^* = \sup_{X \succcurlyeq 0} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \mathcal{L}_c(X, \boldsymbol{y}), \tag{4.4.28}$$

where $c > 0$ is the penalty parameter and

$$\mathcal{L}_c(X, \boldsymbol{y}) \coloneqq \operatorname{Tr}[CX] + \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{\Phi}(X)) - \frac{c}{2} \|\boldsymbol{b} - \boldsymbol{\Phi}(X)\|^2. \tag{4.4.29}$$

The method for optimization associated with the Augmented Lagrangian formulation is known as the Augmented Lagrangian Method (ALM), independently introduced in [Hes69] and [Pow69]. This classical method involves the following three steps that iterate until convergence:

1. $X^{k+1} := \arg\max_{X \succcurlyeq 0} \mathcal{L}_c(X, \boldsymbol{y}^k)$,
2. Update the dual variable $\boldsymbol{y}$ according to $\boldsymbol{y}^{k+1} := \boldsymbol{y}^k - c\,(\boldsymbol{b} - \boldsymbol{\Phi}(X))$,
3. Update the penalty parameter c according to $c_{k+1} := c_0 \mu^{k+1}$, where $\mu > 1$.

Due to the equivalence between (4.4.25) and (4.4.28), iterating through the above steps until convergence leads to the primal optimal value. Hence, we can write

$$p^* = \sup_{X \succcurlyeq 0} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \mathcal{L}_c(X, \boldsymbol{y}) \tag{4.4.30}$$

$$= \sup_{X \succcurlyeq 0} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \left\{ \mathrm{Tr}[CX] + \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{\Phi}(X)) - \frac{c}{2} \|\boldsymbol{b} - \boldsymbol{\Phi}(X)\|^2 \right\}. \tag{4.4.31}$$

ALM is a well-studied method for solving unconstrained optimization problems with convex objective functions. It is considered superior to methods that exclusively involve the original Lagrangian or exclusively involve the penalty term [Ber76]. ALM converges faster than the original Lagrangian method, as it involves a quadratic penalty term in addition to a linear penalty term of the original Lagrangian. Furthermore, it solves many issues associated with both these methods, such as ill-conditioning arising due to large values of the penalty parameter.

Now, we make a mild assumption that one of the constraints is a trace constraint on the primal PSD operator X. Specifically, let $A_M = \mathbb{I}$ and $b_M = \lambda$. Hence, the constraint is $\mathrm{Tr}[X] = \lambda$. Subsequently, substituting $X = \lambda\rho$, where $\rho \in \mathcal{D}^N$ in (4.4.30), we obtain the

following:

$$p^* = \sup_{\rho \in \mathcal{D}^N} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \mathcal{L}_c(\lambda\rho, \boldsymbol{y}) \tag{4.4.32}$$

$$= \sup_{\rho \in \mathcal{D}^N} \inf_{\boldsymbol{y} \in \mathbb{R}^M} \left\{ \lambda \operatorname{Tr}[C\rho] + \boldsymbol{y}^\top (\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\rho)) - \frac{c}{2} \|\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\rho)\|^2 \right\}. \tag{4.4.33}$$

This optimization problem is now expressed in terms of the expectation values of the Hermitian operators $C,\ A_1, \ldots, A_M$ with respect to the density operator ρ. According to Assumption 4, the matrices $C,\ A_1, \ldots, A_M$ consist of poly(n) terms.

Variational Quantum Algorithm for SDPs in ECSF

Solving the unconstrained optimization problem (4.4.32) using ALM is computationally intractable if the dimension N of the operators is large. Therefore, we propose a variational quantum algorithm to solve this problem.

As before, we first introduce a parameterization of the density matrix ρ and then optimize the modified objective function over the subspace of those parameters using our variational quantum algorithm.

Parameterization: Let $U_{RS}(\boldsymbol{\theta})$ be a parameterized quantum circuit with parameter $\boldsymbol{\theta} \in [0, 2\pi]^r$, and suppose that it acts on the all-zeros state of the quantum system RS and prepares a purification of the density operator $\rho_S(\boldsymbol{\theta})$. The form of $U_{RS}(\boldsymbol{\theta})$ is given by (4.3.18). Moreover, the following equality holds because $U_{RS}(\boldsymbol{\theta})$ generates a purification of $\rho_S(\boldsymbol{\theta})$:

$$\operatorname{Tr}[H\rho_S(\boldsymbol{\theta})] = \langle \mathbf{0}|_{RS}\, U_{RS}^\dagger(\boldsymbol{\theta})\, (I_R \otimes H_S)\, U_{RS}(\boldsymbol{\theta}) |\mathbf{0}\rangle_{RS} = \langle I \otimes H \rangle_{\boldsymbol{\theta}}, \tag{4.4.34}$$

where $H \in \{C, A_1, \ldots, A_M\}$.

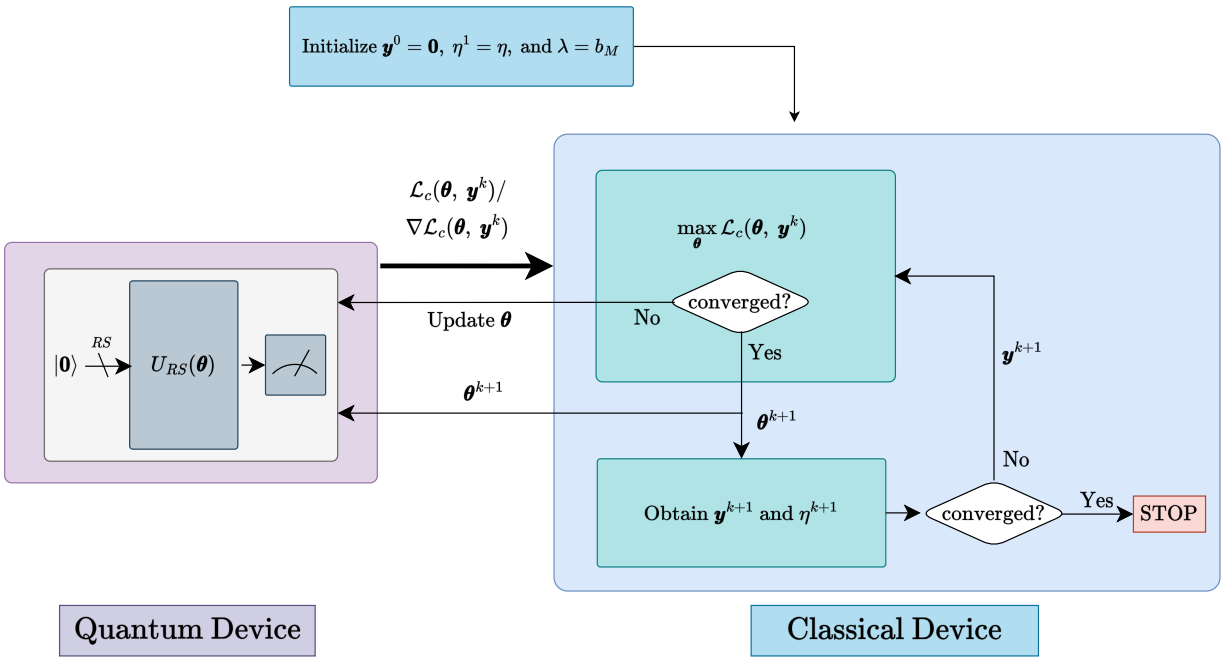


Figure 4.2: This figure depicts the iVQAEC algorithm in which we utilize one parameterized quantum circuit, i.e., $U_{RS}(\boldsymbol{\theta})$.

Due to the parameterization of the density operator $\rho_S(\boldsymbol{\theta})$, the problem (4.4.32) transforms into the following optimization problem:

$$p^* := \sup_{\boldsymbol{\theta}\in[0,2\pi]^r} \inf_{\boldsymbol{y}\in\mathbb{R}^M} \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y}), \tag{4.4.35}$$

where

$$\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y}) := \lambda \langle I \otimes C \rangle_{\boldsymbol{\theta}} + \boldsymbol{y}^\top (\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\boldsymbol{\theta})) - \frac{c}{2} \|\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\boldsymbol{\theta})\|^2 , \tag{4.4.36}$$

$$\boldsymbol{\Phi}(\boldsymbol{\theta}) := (\langle I \otimes A_1 \rangle_{\boldsymbol{\theta}}, \ldots, \langle I \otimes A_M \rangle_{\boldsymbol{\theta}})^\top . \tag{4.4.37}$$

As mentioned before, we assume that the objective function $\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$ is faithful, which means that the global optimal value of the optimization problem (4.2.4) is equal to p^*.

We now focus on solving the optimization problem (4.2.4) using the modified Lagrangian $\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$. Note that, in general, the objective function provided above has a non-

convex landscape with respect to the quantum circuit parameter $\boldsymbol{\theta}$. Also, it is linear in terms of the dual variable $\boldsymbol{y}$. Therefore, the optimization problem (4.2.4) is a non-convex–concave optimization problem. For a convex-concave setting, the natural algorithm to consider is to first perform the maximization until a globally optimal point is reached and then update the dual variable, and repeat these steps until convergence (similar to ALM). On the contrary, for a non-convex–concave setting, solving the maximization until global optimality is reached is generally NP-hard [DDG+22, Section 2.1]. Therefore, we focus on obtaining approximate stationary points of (4.2.4). As a global optimal point is also a stationary point, if we use techniques to initialize quantum circuit parameters according to the problem at hand, such that these parameters lie in the vicinity of a global optimal point, then our algorithm will converge to that point.

Definition 56 (First order (ϵ, c)-stationary point of (4.2.4)) *For $\epsilon, c > 0$, a point $\boldsymbol{\theta} \in [0, 2\pi]^r$ is a first order (ϵ, c)-stationary point of (4.2.4) if there exists $\boldsymbol{y} \in \mathbb{R}^M$ such that the following hold:*

$$\|\nabla_{\theta} \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})\| \leq \epsilon, \qquad \|\lambda \boldsymbol{\Phi}(\boldsymbol{\theta}) - \boldsymbol{b}\| \leq \epsilon. \tag{4.4.38}$$

The above condition acts as a measure of closeness to the first order (ϵ, c)-stationary points of (4.2.4). Such a measure is useful to give a stopping criterion for the algorithm.

We now propose a variational quantum algorithm for obtaining a first order (ϵ, c)-stationary point of (4.2.4). We call this algorithm *inexact Variational Quantum Algorithm for Equality Constrained standard form* (iVQAEC). It is an inexact version because we solve the subproblem involving the maximization to approximate stationary points instead of solving it until global optimality is reached, due to the nonconcave nature of the objective function in terms of quantum circuit parameters. The pseudocode of iVQAEC is given by Algorithm 6.

Algorithm 6: iVQAEC$(C, \{A_i\}_{i=1}^{M},\ \boldsymbol{b},\ \epsilon',\ \eta,\ \mu)$

Input: Hermitian operators $(C, A_1, \ldots, A_M)$, vector $\boldsymbol{b}$, final precision $\epsilon' > 0$, learning rate $\eta > 0$, and a constant $\mu > 1$.

Initialization: $\boldsymbol{y}^1 = \boldsymbol{0}, \eta^1 = \eta, \lambda = b_M$

 # For any step, expectation values of observables and their gradients are evaluated using a parameterized quantum circuit.

for $k = 1, 2, \ldots,$ **do**

  $c_k = -\mu^k$, $\epsilon_{k+1} = 1/\mu^k$

  Maximize $\mathcal{L}_{c_k}(\cdot, \boldsymbol{y}^k)$, where $\boldsymbol{y}^k$ is kept fixed, to obtain $\boldsymbol{\theta}^{k+1}$ such that the following holds:

$$\left\|\nabla \mathcal{L}_{c_k}(\boldsymbol{\theta}^{k+1}, \boldsymbol{y}^k)\right\| \leq \epsilon_{k+1}$$

  $\eta^{k+1} = \eta^1 \min\left\{\frac{\left\|\lambda \boldsymbol{\Phi}(\boldsymbol{\theta}^1) - \boldsymbol{b}\right\| \ln^2 2}{\left\|\lambda \boldsymbol{\Phi}(\boldsymbol{\theta}^{k+1}) - \boldsymbol{b}\right\| (k+1) \ln^2 (k+2)},\ 1\right\}$

  $\boldsymbol{y}^{k+1} = \boldsymbol{y}^k - \eta^{k+1}\left(\boldsymbol{b} - \lambda \boldsymbol{\Phi}(\boldsymbol{\theta}^{k+1})\right)$

  if $\left\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{c_k}(\boldsymbol{\theta}^{k+1}, \boldsymbol{y}^{k+1})\right\| + \left\|\lambda \boldsymbol{\Phi}(\boldsymbol{\theta}^{k+1}) - \boldsymbol{b}\right\| \leq \epsilon'$ **then**

    STOP and return $\mathcal{L}_{c_k}(\boldsymbol{\theta}^{k+1}, \boldsymbol{y}^{k+1})$

  end if

end for

We run iVQAEC on a classical computer, and at any step of the algorithm, the expectation value of a Hermitian operator $H \in \{C, A_1, \ldots, A_M\}$ (i.e., $\langle H \rangle_{\boldsymbol{\theta}}$) is evaluated using a quantum circuit with parameter $\boldsymbol{\theta}$ (see Figure 4.2). Moreover, the partial derivative of $\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$ with respect to the parameter $\boldsymbol{\theta}$ depends on the partial derivatives of the expectation values of the Hermitian operators $I \otimes C$, $I \otimes A_1$, $\ldots$, $I \otimes A_M$ with respect to $\boldsymbol{\theta}$. For evaluating the partial derivatives of the latter with respect to $\boldsymbol{\theta}$, we use the parameter-shift

rule (see (4.3.21)). Moreover, we do not explicitly mention these calls to quantum circuits in the pseudocode of the algorithm.

Unbiased Estimators: As mentioned earlier in the previous section, due to the fact that evaluation of the expectation value of a Hermitian operator using a quantum circuit is stochastic in nature, we use unbiased estimators of expectation values and their corresponding partial derivatives (see (4.3.23) and (4.3.25)). Furthermore, it suffices to have *independent* unbiased estimators of $\nabla_{\boldsymbol{\theta}}\langle I \otimes C\rangle_{\boldsymbol{\theta}}$, $\nabla_{\boldsymbol{\theta}}\langle I \otimes A_1\rangle_{\boldsymbol{\theta}}$, $\ldots$, $\nabla_{\boldsymbol{\theta}}\langle I \otimes A_M\rangle_{\boldsymbol{\theta}}$, $\langle I \otimes A_1\rangle_{\boldsymbol{\theta}}$, $\ldots$, $\langle I \otimes A_M\rangle_{\boldsymbol{\theta}}$ in order to construct an unbiased estimator of $\nabla_{\boldsymbol{\theta}}\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$.

Convergence Rate

In this section, we provide the convergence rate and total iteration complexity of iVQAEC in terms of the number of iterations of the for-loop at Step 3 of Algorithm 6 needed to arrive at an approximate stationary point of (4.4.35).

Recently, the authors of [SEA+19] studied the following non-convex optimization problem:

$$\inf_{x\in\mathbb{R}^d} \{f(x) + g(x) : \boldsymbol{A}(x) = 0\}, \tag{4.4.39}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a continuously-differentiable non-convex function, $\boldsymbol{A} : \mathbb{R}^d \to \mathbb{R}^m$ is a nonlinear operator, and $g : \mathbb{R}^d \to \mathbb{R}$ is a convex function. The constrained optimization problem (4.4.39) is equivalent to the following unconstrained minimax formulation:

$$\inf_{\boldsymbol{x}\in\mathbb{R}^d} \sup_{\boldsymbol{y}\in\mathbb{R}^m} \left\{\mathcal{L}_\beta(x, \boldsymbol{y}) + g(\boldsymbol{x})\right\}, \tag{4.4.40}$$

where

$$\mathcal{L}_\beta(x, \boldsymbol{y}) \coloneqq f(\boldsymbol{x}) + \boldsymbol{y}^\top A(\boldsymbol{x}) + \frac{\beta}{2}\|\boldsymbol{A}(\boldsymbol{x})\|^2, \tag{4.4.41}$$

$\beta > 1$, and $\mathcal{L}_\beta(x, y)$ is the augmented Lagrangian. As the classic ALM algorithm is suited for solving optimization problems involving convex objective functions, the authors proposed an inexact Augmented Lagrangian method, known as the iALM algorithm for obtaining an approximate first order stationary point of (4.4.40). It is an inexact version because the subproblem involving a maximization is over a non-convex function. They claim that if we use an inexact solver for that subproblem at each iteration, then their algorithm converges to an approximate first order stationary point (see Theorem 4.1 of [SEA+19]). They also provide the total iteration complexity of iALM in terms of the number of iterations of the for-loop (see Corollary 4.2 of [SEA+19]).

Our problem (4.4.35) also consists of an augmented Lagrangian term, which is non-convex in terms of the quantum circuit parameters $\boldsymbol{\theta}$, and $g(\boldsymbol{\theta}) = 0$. Additionally, for solving our problem, we use an inexact solver for the subproblem (Step 5) of iVQAEC, and the update rule for modifying the learning rate η at each iteration (Step 6) is similar to their iALM algorithm.

Now, the following theorem characterizes iVQAEC's convergence rate and total iteration complexity for finding an approximate stationary point of the problem (4.4.35) in terms of the number of iterations of the for-loop (Step 3). The proof of this theorem is provided by the proofs of Theorem 4.1 and Corollary 4.2 of [SEA+19].

Theorem 57 (Convergence rate and total iteration complexity of iVQAEC) *For integers $k_1 \geq k_0 \geq 2$, consider the interval $K := \{k_0, \ldots, k_1\}$, and let $\{\boldsymbol{\theta}^k\}_{k \in K}$ be the output sequence of iVQAEC on the interval K, where $\boldsymbol{\theta}^k \in [0, 2\pi]^r$ for all $k \in K$. Suppose that $f(\boldsymbol{\theta}) := \langle I \otimes C \rangle_{\boldsymbol{\theta}}$ and $\boldsymbol{A}(\boldsymbol{\theta}) := \lambda \boldsymbol{\Phi}(\boldsymbol{\theta}) - \boldsymbol{b}$ have Lipschitz constants L_f and $L_{\boldsymbol{A}}$. Also suppose that the function $\mathcal{L}_c(\cdot, \boldsymbol{y})$ is smooth for every $\boldsymbol{y} \in \mathbb{R}^M$ (i.e., its gradient is $L_{c,\boldsymbol{y}}$-Lipschitz continuous). Additionally, suppose*

that there exists $\nu > 0$ *such that*

$$\nu \|\boldsymbol{A}(\boldsymbol{\theta}^k)\| \leq \left\| -J_{\boldsymbol{A}}(\boldsymbol{\theta}^k)^\top \boldsymbol{A}(\boldsymbol{\theta}^k) \right\|, \tag{4.4.42}$$

for every $k \in K$*. If an inexact solver is used in Step 5 of the algorithm, then* $\boldsymbol{\theta}^k$ *is a first order* (ϵ_k, c_k)*-stationary point of* (4.4.35) *with*

$$\epsilon_k = \frac{1}{c_{k-1}} \left(\frac{2(L_f + y_{\max} L_{\boldsymbol{A}})(1 + \eta^k L_{\boldsymbol{A}})}{\nu} + 1 \right) =: \frac{Q(f, \boldsymbol{A}, \eta^1)}{c_{k-1}}, \tag{4.4.43}$$

for every $k \in K$*, where* $y_{\max}(\boldsymbol{\theta}^1, \boldsymbol{y}^0, \eta^1)$ *is given as*

$$y_{\max}(\boldsymbol{\theta}^1, \boldsymbol{y}^0, \eta^1) = \|\boldsymbol{y}^0\| + \eta^1 \|\boldsymbol{A}(\boldsymbol{\theta}^1)\| \ln^2 2. \tag{4.4.44}$$

Additionally, if we use the Accelerated Proximal Gradient Method (APGM) from [LL15] as an inexact solver for the subproblem (Step 5), then the algorithm finds a first-order (ϵ_T, c_T)*-stationary point, after* T *calls to the solver, where*

$$T = O\left(\frac{rQ^3}{\epsilon_T^4} \log_\mu \left(\frac{Q}{\epsilon_T} \right) \right) = \tilde{O}\left(\frac{rQ^3}{\epsilon_T^4} \right), \tag{4.4.45}$$

with $Q \equiv Q(f, \boldsymbol{A}, \eta^1)$*, and* μ *is a constant as mentioned in Algorithm 6.*

The inequality given by (4.4.42) is known as the PL inequality for minimizing $\|\boldsymbol{A}(\boldsymbol{\theta})\|^2$ [KNS16]. This regularity condition is used in the proof of Theorem 4.1 of [SEA+19] for obtaining a bound on $\left\|\boldsymbol{A}(\boldsymbol{\theta}^k)\right\|$ for all $k \in K$. For our purposes, we assume that this condition holds for all $\boldsymbol{\theta} \in [0, 2\pi]^r$. Then, Theorem 57 is a direct consequence of the following statements:

(A) **Smoothness of** $\mathcal{L}_c(\cdot, \boldsymbol{y})$: The function $\mathcal{L}_c(\cdot, \boldsymbol{y})$ is smooth for every $\boldsymbol{y} \in \mathbb{R}^M$ (i.e., its gradient is $L_{c,\boldsymbol{y}}$-Lipschitz continuous).

(B) **Lipschitz continuity of** $f(\boldsymbol{\theta})$ **and** $\boldsymbol{A}(\boldsymbol{\theta})$: The functions $f(\boldsymbol{\theta})$ and $\boldsymbol{A}(\boldsymbol{\theta})$ are Lipschitz continuous with constants L_f and $L_{\boldsymbol{A}}$, respectively.

We prove the statements (A) and (B) in Lemmas 59 and 60, respectively. Before we prove these statements, we state the following lemma that concerns Lipschitz continuity of the function $h : [0, 2\pi]^r \to \mathbb{R}$ defined as

$$h(\boldsymbol{\theta}) \coloneqq \langle I \otimes O \rangle_{\boldsymbol{\theta}} = \langle \mathbf{0} | U^\dagger(\boldsymbol{\theta})(I \otimes O)U(\boldsymbol{\theta}) | \mathbf{0} \rangle), \tag{4.4.46}$$

and its gradient $\nabla h(\boldsymbol{\theta})$, where $O \in \mathcal{S}^N$.

Lemma 58 (Lipschitz continuity of $h(\boldsymbol{\theta})$ and $\nabla h(\boldsymbol{\theta})$) *The function $h : [0, 2\pi]^r \to \mathbb{R}$ and its gradient, i.e., the vector-valued function $\nabla h : [0, 2\pi]^r \to \mathbb{R}^r$, are L_h-Lipschitz and $L_{\nabla h}$-Lipschitz continuous, respectively, for some $L_h, L_{\nabla h} > 0$.*

Proof. The proof is given in Appendix 4.C. ■

Next, we formally write the statements (A) and (B) as lemmas.

Lemma 59 (Smoothness of $\mathcal{L}_c(\cdot, \boldsymbol{y})$) *For all $\boldsymbol{y} \in \mathbb{R}^M$, there exists $L_{c,\boldsymbol{y}} > 0$ such that the gradient of the function $\mathcal{L}_c(\cdot, \boldsymbol{y}) : [0, 2\pi]^r \to \mathbb{R}$ is $L_{c,\boldsymbol{y}}$-Lipschitz continuous.*

Proof. The proof is given in Appendix 4.D. ■

Lemma 60 (Lipschitz continuity of $f(\boldsymbol{\theta})$ and $\boldsymbol{A}(\boldsymbol{\theta})$) *The function $f : [0, 2\pi]^r \to \mathbb{R}$, where $f(\boldsymbol{\theta}) = \langle I \otimes C \rangle_{\boldsymbol{\theta}}$, and the linear map $\boldsymbol{A}(\boldsymbol{\theta}) : [0, 2\pi]^r \to \mathbb{R}^M$ are L_f-Lipschitz and $L_{\boldsymbol{A}}$-Lipschitz continuous, respectively, for some $L_f, L_{\boldsymbol{A}} > 0$.*

Proof. The proof directly follows from the Lipschitz continuity of $h(\boldsymbol{\theta})$ (Lemma 58). ■

4.4.3 Inequality Constrained Standard Form (ICSF) of SDPs

In this section, we consider the following inequality constrained primal form of SDPs:

$$\begin{aligned} p^* = \sup_{X \succcurlyeq 0} \ & \mathrm{Tr}[CX] \\ \text{subject to} \ & \boldsymbol{\Phi}(X) \leq \boldsymbol{b}. \end{aligned} \tag{4.4.47}$$

Here we take vector forms of the Hermiticity-preserving linear map Φ and the Hermitian operator B because both have a diagonal form. Furthermore, we assume that the last constraint is the trace constraint on the primal PSD variable X. As such, we set $A_M = I$ and $\mathrm{Tr}[X] \leq b_M$. Additionally, for this case also, we make an assumption that the SDPs are weakly constrained, i.e., $N \gg M$. The corresponding dual form is given as follows:

$$\begin{aligned} d^* = \inf_{\boldsymbol{y} \geq 0} \ & \boldsymbol{b}^\top \boldsymbol{y} \\ \text{subject to} \ & \Phi^\dagger(\boldsymbol{y}) \succcurlyeq C, \end{aligned} \tag{4.4.48}$$

where $\boldsymbol{y} = (y_1, \ldots, y_M)^\top$ is a dual variable. We can determine the dual map $\Phi^\dagger$ as follows, using the definition of the adjoint of a linear map (see Definition (4.3.2)):

$$\boldsymbol{y}^\top \boldsymbol{\Phi}(X) = \sum_{i=1}^{M-1} y_i \, \mathrm{Tr}[A_i X] + y_M \, \mathrm{Tr}[X] \tag{4.4.49}$$

$$= \mathrm{Tr}\left[\left(\sum_{i=1}^{M-1} y_i A_i + y_M I\right) X\right], \tag{4.4.50}$$

which implies that

$$\Phi^\dagger(\boldsymbol{y}) = \sum_{i=1}^{M-1} y_i A_i + y_M I. \tag{4.4.51}$$

We derive unconstrained formulations for the primal and dual forms of inequality constrained SDPs. First, by taking the primal form of SDPs into account, we introduce slack variables and convert the inequality constrained problem (4.4.47) to the following

equality constrained problem:

$$\begin{aligned} p^* = \sup_{X \succcurlyeq 0, z \geq 0} \quad & \mathrm{Tr}[CX] \\ \text{subject to} \quad & \boldsymbol{b} - \boldsymbol{\Phi}(X) = \boldsymbol{z}, \end{aligned} \tag{4.4.52}$$

where $\boldsymbol{z} = (z_1, \ldots, z_M)^\top$ is a vector of slack variables and $z_1, \ldots, z_M \geq 0$. Now, with the problem (4.4.52) being an equality constrained problem, we can use iVQAEC (see Algorithm 6) to solve it.

Next, we focus on the dual form of inequality constrained SDPs as follows:

$$d^* = \inf_{\boldsymbol{y} \geq 0} \left\{ \boldsymbol{b}^\top \boldsymbol{y} : \Phi^\dagger(\boldsymbol{y}) \succcurlyeq C \right\} \tag{4.4.53}$$

$$= \inf_{y_1, \ldots, y_M \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + y_M b_M : \sum_{i=1}^{M-1} y_i A_i + y_M I \succcurlyeq C \right\} \tag{4.4.54}$$

$$= \inf_{y_1, \ldots, y_M \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + y_M b_M : y_M I \succcurlyeq C - \sum_{i=1}^{M-1} y_i A_i \right\}. \tag{4.4.55}$$

Now, consider that

$$\inf_{t \geq 0} \{ t : tI \succcurlyeq H \} = \max\{0, \lambda_{\max}(H)\} = \max \left\{ 0, \sup_{\rho \in \mathcal{D}^N} \mathrm{Tr}[H\rho] \right\}, \tag{4.4.56}$$

where H is a Hermitian operator, $\lambda_{\max}(H)$ is the maximum eigenvalue of H, and the supremum is over the set $\mathcal{D}^N$ of density operators. Using this fact, we convert the constrained problem (4.4.55) into the following unconstrained problem:

$$d^* = \inf_{y_1, \ldots, y_{M-1} \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + b_M \cdot \max \left\{ 0, \sup_{\rho \in \mathcal{D}^N} \mathrm{Tr} \left[\left(C - \sum_{i=1}^{M-1} y_i A_i \right) \rho \right] \right\} \right\} \tag{4.4.57}$$

$$= \inf_{\bar{\boldsymbol{y}} \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + b_M \cdot \max \left\{ 0, \sup_{\rho \in \mathcal{D}^N} \mathrm{Tr} \left[H(\bar{\boldsymbol{y}}) \rho \right] \right\} \right\}, \tag{4.4.58}$$

where we set

$$\bar{\boldsymbol{y}} := (y_1, \ldots, y_{M-1})^\top, \tag{4.4.59}$$

$$H(\bar{\mathbf{y}}) \coloneqq C - \sum_{i=1}^{M-1} y_i A_i. \tag{4.4.60}$$

Furthermore, as the first term is independent of ρ, we have that

$$d^* = \inf_{\bar{\mathbf{y}} \geq 0} \sup_{\rho \in \mathcal{D}^N} \left\{ \sum_{i=1}^{M} b_i y_i + b_M \cdot \max\left\{0, \operatorname{Tr}\left[H(\bar{\mathbf{y}})\rho\right]\right\} \right\}. \tag{4.4.61}$$

We can use Sion's minimax theorem [Sio58] and interchange the supremum and infimum because the set $\mathcal{D}^N$ of density operators is compact and convex, the objective function is convex with respect to $\bar{\mathbf{y}}$ for all $\rho \in \mathcal{D}^N$, and it is quasi-concave with respect to $\rho \in \mathcal{D}^N$ for all $\bar{\mathbf{y}} \geq 0$. Hence,

$$d^* = \sup_{\rho \in \mathcal{D}^N} \inf_{\bar{\mathbf{y}} \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + b_M \cdot \max\left\{0, \operatorname{Tr}\left[H(\bar{\mathbf{y}})\rho\right]\right\} \right\}. \tag{4.4.62}$$

The above problem is not differentiable at some parameter values due to the max operation in the objective function. In order to smoothen the sharp corners that result from this operation, we modify the above problem in the following way by introducing $\gamma > 0$:

$$d' \coloneqq - \sup_{\rho \in \mathcal{D}^N} \inf_{\bar{\mathbf{y}} \geq 0} \left\{ \sum_{i=1}^{M-1} b_i y_i + \frac{b_M}{\gamma} \ln\left(e^{\gamma \operatorname{Tr}[H(\bar{\mathbf{y}})\rho]} + 1\right) \right\}. \tag{4.4.63}$$

In the above, we used the scaled version of the log-sum-exp function [BV04, Section 3.1.5] as an approximation to the original max function in (4.4.62). This approximation can be controlled by varying the parameter γ because the following holds for all $x, y \in \mathbb{R}$:

$$\max\{x, y\} \leq \frac{\ln(e^{\gamma x} + e^{\gamma y})}{\gamma} \leq \max\{x, y\} + \frac{\ln(2)}{\gamma}. \tag{4.4.64}$$

By using this approximation, the objective function in (4.4.63) does not have any sudden corners where the partial derivatives can change drastically. In fact, the objective function of (4.4.63) is infinitely differentiable. It is also equivalent to the previous objective function in (4.4.62) in the limit $\gamma \to \infty$. For all finite values of γ, the value d^* is bounded from above by d'.

Variational Quantum Algorithm for SDPs in ICSF

Solving the unconstrained optimization problem (4.4.63) using a gradient based classical technique is computationally expensive if the dimension N of the operators is large. Hence, in this section, we propose a variational quantum algorithm to solve this problem.

First, we introduce a parameterization of density operators using parameterized quantum circuits, in order to efficiently estimate the expectation values of the Hermitian operators $C, A_1, \ldots, A_{M-1}$. Second, we optimize the resulting objective function using our variational quantum algorithm.

Parameterization: Let $U_{RS}(\boldsymbol{\theta})$ be a parameterized quantum circuit with a parameter $\boldsymbol{\theta} \in [0, 2\pi]^r$, and let it act on the all-zeros state of the quantum system RS and prepare a purification of the density operator $\rho_S(\boldsymbol{\theta})$. The structure of $U_{RS}(\boldsymbol{\theta})$ is given by (4.3.18). Furthermore, the following equality holds because $U_{RS}(\boldsymbol{\theta})$ generates a purification of $\rho_S(\boldsymbol{\theta})$:

$$\mathrm{Tr}[H\rho_S(\boldsymbol{\theta})] = \langle \mathbf{0}|_{RS}\, U_{RS}^{\dagger}(\boldsymbol{\theta})\,(I_R \otimes H_S)\, U_{RS}(\boldsymbol{\theta})|\mathbf{0}\rangle_{RS} = \langle I \otimes H\rangle_{\boldsymbol{\theta}}, \tag{4.4.65}$$

where $H \in \{C, A_1, \ldots, A_{M-1}\}$.

Due to the parameterization of the density operator $\rho_S(\boldsymbol{\theta})$, the problem (4.4.63) transforms into the following optimization problem:

$$d' := \sup_{\boldsymbol{\theta}\in[0,2\pi]^r} \inf_{\bar{\mathbf{y}}\geq 0} \mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\mathbf{y}}), \tag{4.4.66}$$

where

$$\mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\mathbf{y}}) := \sum_{i=1}^{M-1} b_i y_i + \frac{b_M}{\gamma} \ln\left(e^{\gamma\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}} + 1\right) \tag{4.4.67}$$

and we optimize over the space of quantum circuit parameters $\boldsymbol{\theta} \in [0, 2\pi]^r$. As mentioned before, we assume that the objective function $\mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\mathbf{y}})$ is faithful, which means that the

global optimal value of the optimization problem (4.4.66) is equal to d'.

Algorithm 7: `iVQAIC`$(C, \{A_i\}_{i=1}^M, \{b_i\}_{i=1}^M, \eta, \epsilon, \gamma)$

Input: Hermitian operators $(C, \{A_i\}_{i=1}^M)$, scalars $\{b_i\}_{i=1}^M$, learning rate $\eta > 0$, precision $\epsilon > 0$, constant $\gamma > 0$.

Initialization: $\bar{\boldsymbol{y}}^1 = \boldsymbol{0}$.

 # For any step, expectation values of observables and their gradients are evaluated using a parameterized quantum circuit.

for $k = 1, 2, \ldots,$ **do**

  Maximize $\mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\boldsymbol{y}}^k)$, where $\bar{\boldsymbol{y}}^k$ is fixed, to obtain $\boldsymbol{\theta}^{k+1}$ such that the following holds:

$$\left\|\nabla \mathcal{F}_\gamma(\boldsymbol{\theta}^{k+1}, \bar{\boldsymbol{y}})\right\| \leq \epsilon.$$

  $\bar{\boldsymbol{y}}^{k+1} = \bar{\boldsymbol{y}}^k - \eta \nabla_{\bar{\boldsymbol{y}}} \mathcal{F}_\gamma(\boldsymbol{\theta}^{k+1}, \bar{\boldsymbol{y}}^k)$

  if $\left\|\nabla \mathcal{F}_\gamma(\boldsymbol{\theta}^{k+1}, \bar{\boldsymbol{y}}^{k+1})\right\| \leq \epsilon$ **then**

    STOP and return $\mathcal{F}_\gamma(\boldsymbol{\theta}^{k+1}, \bar{\boldsymbol{y}}^{k+1})$

  end if

end for

The objective function of (4.4.66) is in general non-convex with respect to the quantum circuit parameter $\boldsymbol{\theta}$. On the contrary, it is concave in $\bar{\boldsymbol{y}}$. Hence, the optimization problem (4.4.66) is a non-convex–concave optimization problem. For such a setting, finding a globally optimal solution is generally NP-hard [DDG+22, Section 2.1]. Therefore, we focus on obtaining approximate stationary points of (4.4.66). As a global optimal point is also a stationary point, if we use techniques to initialize quantum circuit parameters according to the problem at hand such that these parameters lie in the vicinity of a global

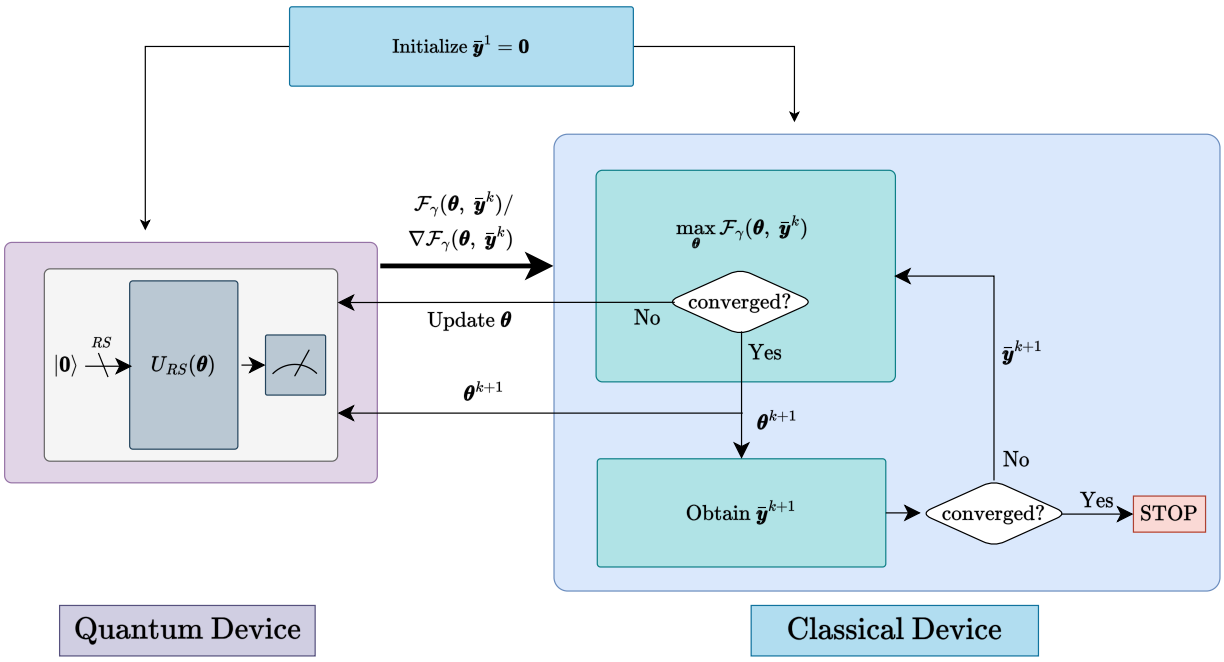


Figure 4.3: This figure depicts the iVQAIC algorithm in which we utilize one parameterized quantum circuit, i.e., $U_{RS}(\boldsymbol{\theta})$.

optimal point, then our algorithm will converge to that point.

Definition 61 (First order ϵ-stationary points of (4.4.66)) *For $\epsilon > 0$, a point $(\boldsymbol{\theta}, \bar{\boldsymbol{y}}) \in [0, 2\pi]^r \times \mathbb{R}^{M-1}$ is a first order ϵ-stationary point of (4.4.66) if the following holds:*

$$\left\|\nabla\mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\boldsymbol{y}})\right\| \leq \epsilon. \tag{4.4.68}$$

We use the above condition as a stopping criterion for our variational quantum algorithm.

In order to obtain a first order ϵ-stationary point of (4.4.66), we propose a variational quantum algorithm (see Algorithm 7). We call this algorithm *inexact Variational Quantum Algorithm for Inequality Constrained standard form* (iVQAIC). It is an inexact version because

we solve the subproblem involving the maximization to an approximate stationary point instead of solving it until global optimality is reached, due to the non-convex nature of the objective function in terms of quantum circuit parameters. Its pseudocode is provided in Algorithm 7, and it is depicted in Figure 4.3.

We run iVQAIC on a classical computer and utilize a parameterized quantum circuit $U_{RS}(\boldsymbol{\theta})$ with a parameter $\boldsymbol{\theta} \in [0, 2\pi]^r$ for estimating the expectation values of the Hermitian operators. Moreover, the partial derivative of $\mathcal{F}_\gamma(\boldsymbol{\theta}, \bar{\boldsymbol{y}})$ with respect to the quantum circuit parameters $\boldsymbol{\theta}$ depends on the partial derivatives of the expectation values of the Hermitian operators $I \otimes C$, $I \otimes A_1$, $\ldots$, $I \otimes A_{M-1}$ with respect to $\boldsymbol{\theta}$. In order to compute the partial derivatives of the latter with respect to $\boldsymbol{\theta}$, we use the parameter-shift rule (see (4.3.21)). Note that we do not explicitly mention these quantum-circuit calls in the pseudocode of the algorithm.

For this case, we do not state a theorem that indicates the convergence rate for finding approximate stationary points of (4.4.66) in terms of the number of iterations in the for-loop of the algorithm. Rather, we prove a property of the objective function that is necessary for providing such a convergence analysis, i.e., smoothness of $\mathcal{F}(\cdot, \bar{\boldsymbol{y}})$ for a fixed $\bar{\boldsymbol{y}} \geq 0$. We state it formally in the following lemma:

Lemma 62 (Smoothness of $\mathcal{F}_\gamma(\cdot, \bar{\boldsymbol{y}})$) *For all $\bar{\boldsymbol{y}} \geq 0$, there exists $L_{\gamma,\bar{\boldsymbol{y}}} > 0$ such that the gradient of the function $\mathcal{F}_\gamma(\cdot, \bar{\boldsymbol{y}}) : [0, 2\pi]^r \to \mathbb{R}$ is $L_{\gamma,\bar{\boldsymbol{y}}}$-Lipschitz continuous.*

Proof. The proof is given in Appendix 4.E. ■

4.5 Numerical Simulations

For validating our reformulations of SDPs and verifying the convergence of their respective algorithms to approximate stationary points, we randomly generated SDPs such that they contain a valid feasible region. To verify the convergence of the algorithms proposed in this paper, we focus on three cases based on the number M of constraints and the dimension N of the input Hermitian operators, and whether the problem is an equality or inequality constrained problem:

1. $N \approx M$: Here, we consider a well-known and extensively studied MaxCut problem. First, we briefly recall what the problem is and its SDP relaxation in the next subsection. This problem is taken into account because the number of constraints and the dimension of the matrices are equal ($N = M$). For this case, we evaluated the performance of iVQAGF (see Algorithm 5), as it does not make the weakly-constrained assumption on SDPs. Figure 4.4 shows the convergence of iVQAGF for solving randomly generated SDP instances of the MaxCut problem. Furthermore, we performed these simulations for different dimensions of the input matrices. Specifically, we considered $N \in \{8, 16, 32\}$.
2. $N \gg M$: We divide this case further according to the type of constraints: (2a) equality constraints and (2b) inequality constraints. For randomly generated equality-constrained problems, we report the performance of iVQAEC (see Algorithm 6). Similarly, we analyse the performace of iVQAIC (see Algorithm 7) for randomly generated instances of an inequality constrained problem. Here also we consider $N \in \{8, 16, 32\}$. Figure 4.5 and Figure 4.6 showcase the convergence of iVQAEC and iVQAIC, respectively, for randomly generated instances of their respective problems.

We take into account Assumption 4 while creating the input Hermitian matrices. We assume that the Pauli string decomposition of these input matrices are provided beforehand. Additionally, for the analysis of iVQAGF for solving MaxCut, we assume that we are provided with the Pauli string decomposition of $C^\top$ and the Choi operator of the linear map Φ, i.e., Γ^Φ.

We executed our algorithms using Pennylane's Python libraries where we set QASM simulator of the Qiskit Python package as a backend.[2] Pennylane is an open-source Python library developed by Xanadu for differential programming of quantum computers [BIS+22]. Similarly, Qiskit is an open-source package/interface developed by IBM to interact with the underlying quantum computer [T+23]. Additionally, the QASM simulator simulates a real IBM Quantum Backend, which is actually noisy in nature due to gate errors and decoherence. We integrated Pennylane and Qiskit's QASM simulator using the Pennylane-Qiskit plugin.

We use the Strong Entangling Layers template of Pennylane as our variational ansatz, where each layer consists of $O(\text{poly}(n))$ single-qubit rotations and entangling gates. We then repeat this layer $O(\text{poly}(n))$ number of times, where n is the number of qubits. Therefore, the overall gate complexity is $O(\text{poly}(n))$.

In order to assess the convergence of our methods to a globally optimal point, we initialize the quantum circuit parameter such that it lies in the convex region of that globally optimal point. We then report the results and their associated analyses on how well our algorithms perform on the QASM noisy simulator and compare the results with a noiseless simulator. We also report the time complexity (number of iterations of the for-loops) of our algorithms, i.e., how fast our algorithms converge to an actual globally optimal

[2]The source code is available here.

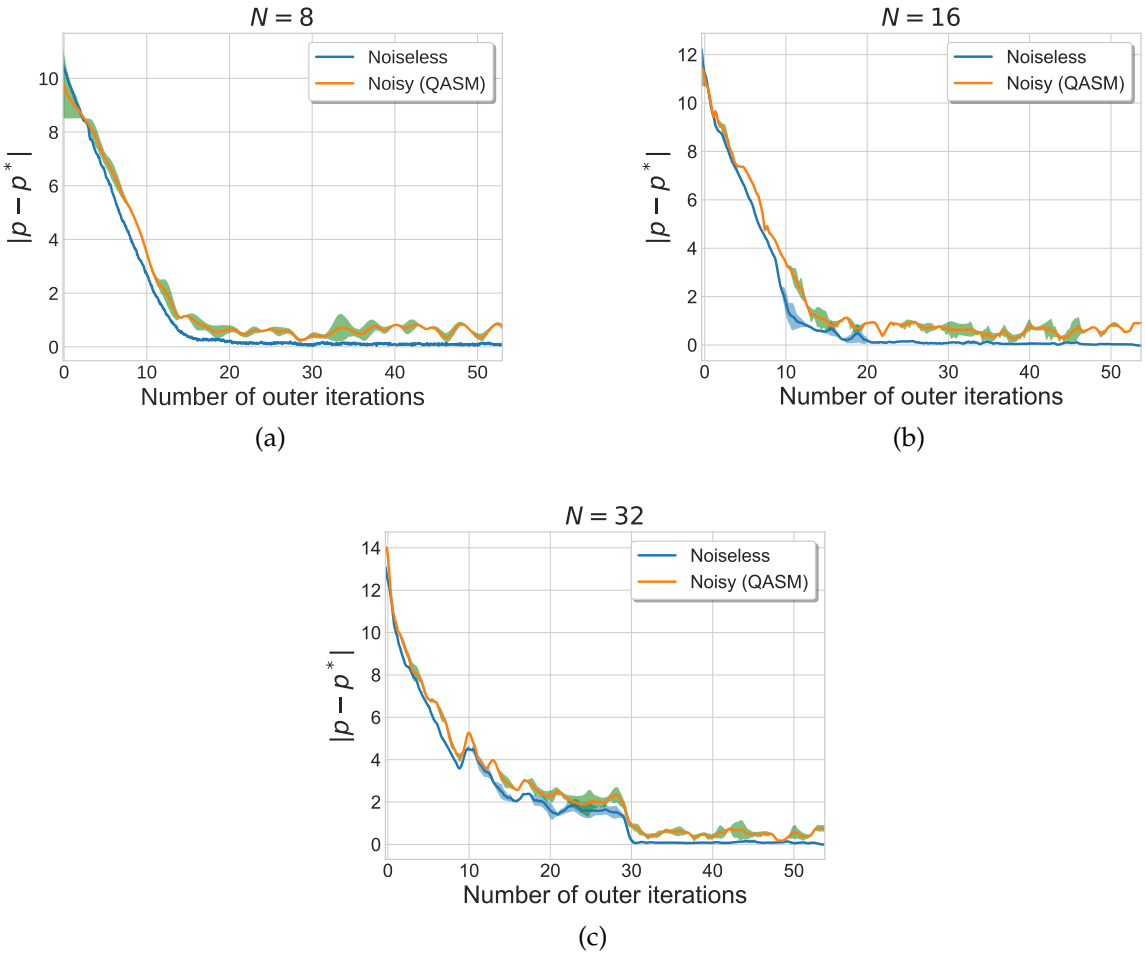


Figure 4.4: Convergence of iVQAGF for three randomly generated MaxCut-SDP instances with different numbers of vertices in the graph: $N \in \{8, 16, 32\}$.

point.

First, for the sake of completeness, we recall the definition of a cut and a MaxCut of a given graph.

Definition 63 (Cut and MaxCut) *A cut is a bi-partition W of the vertex set V of a graph $G = (V, E)$, where $|V| = N$. An edge $(i, j) \in E$ is part of the cut set if its vertices i and j lie in separate*

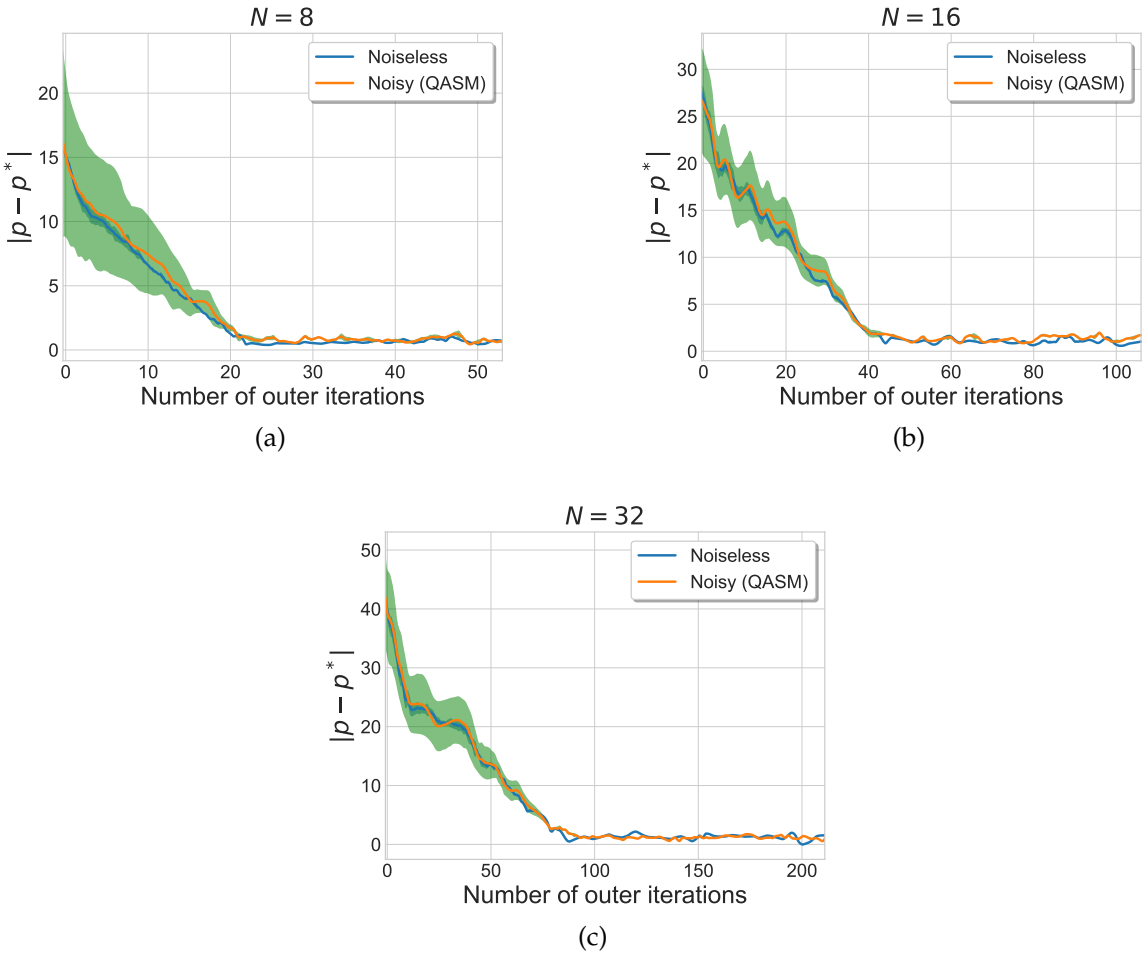
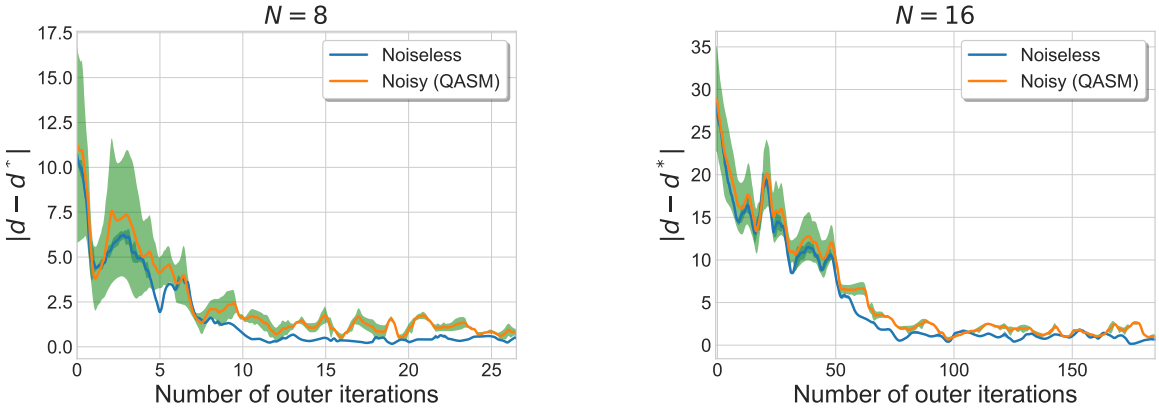


Figure 4.5: Convergence of iVQAEC for three separate cases of randomly generated equality constrained semidefinite programs and nonempty feasible regions: $N \in \{8, 16, 32\}$.

partitions. A MaxCut is the largest cut possible of a graph G.

The problem of finding a MaxCut of a graph can be formulated as a Quadratic Integer Program (QIP) [MP90]:

$$\begin{aligned} [\text{QIP}]: \quad & \sup \sum_{(i,j)\in E} \frac{1}{4}(x_i - x_j)^2 \\ & \text{subject to } \; x_i \in \{-1, 1\}; \; \forall i \in V. \end{aligned} \tag{4.5.1}$$



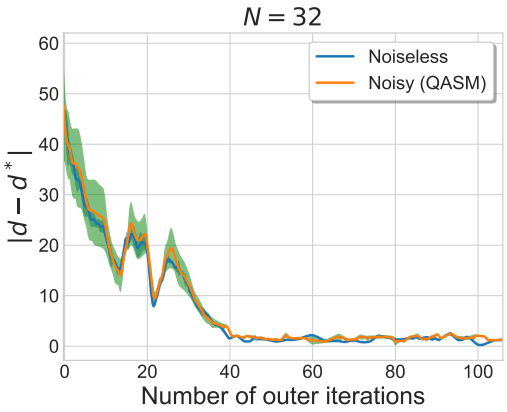


Figure 4.6: Convergence of iVQAIC for three separate cases of randomly generated inequality constrained semidefinite programs with nonempty feasible regions: $N \in \{8, 16, 32\}$.

Here, the optimal value of the above optimization is the optimal cut size (i.e., the number of the edges in the MaxCut). This quadratic integer program is in general computationally intractable [PY91]. However, there exist LP and SDP relaxations for the above program [GW95].

$$\begin{aligned} [\text{LP}]: \quad & \sup \sum_{(i,j)\in E} \frac{1}{4}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 \\ & \text{subject to } \|\boldsymbol{v}_i\|^2 = 1, \boldsymbol{v}_i \in \mathbb{R}^N; \ \forall i \in V. \end{aligned} \tag{4.5.2}$$

The above mentioned LP formulation is not good enough as it is a 1/2-approximation to the original problem. It is well known that this can be extended to an SDP formulation in which an algorithm proposed in [GW95] gives a 0.879-approximation of the original QIP in (4.5.1):

$$\begin{aligned} [\text{SDP}]: \quad & \sup \sum_{(i,j)\in E} \frac{1}{4}(X_{ii} - 2X_{ij} + X_{jj}) \\ & \text{subject to } \; X \succcurlyeq 0, \; X_{ii} = 1, \; \forall i \in V. \end{aligned} \tag{4.5.3}$$

For our case, we numerically simulate iVQAGF for solving the aforementioned SDP relaxation of a MaxCut problem. Recasting the constraints of the above MaxCut SDP as constraints of the general form of SDP, we can write them as $\Phi(X) = B$, where

$$B = I, \qquad \Phi(X) = \text{diag}\left(\text{Tr}[A_1 X], \ldots, \text{Tr}[A_M X]\right), \tag{4.5.4}$$

and $A_i = |i\rangle\langle i|$. Therefore, the Choi operator for this linear map is given as

$$\Gamma^{\Phi} = \sum_{i,j=0}^{N-1} |i\rangle\langle j| \otimes \Phi(|i\rangle\langle j|) = \sum_{i=0}^{N-1} |i\rangle\langle i| \otimes |i\rangle\langle i|. \tag{4.5.5}$$

Additionally, the objective function can be written as $\text{Tr}[CX]$, where $C = L/4$ and L is the Laplacian matrix for the graph G.

The Pauli decomposition of the above Choi operator consists of $N = 2^n$ Pauli strings. This gives the impression that we need to compute 2^n expectation values in order to compute the expectation value of Γ^{Φ}. However, all these Pauli strings can be constructed from just I and σ_z operators, because it is clearly a diagonal matrix. This implies that all the Pauli strings are mutually commuting, and their expectation values can be estimated simultaneously [CvSW+21]. Therefore, we just need to compute a single expectation value to evaluate the expectation value of Γ^{Φ}.

The shaded regions in Figure 4.4, 4.5, and 4.6 signify the variance in the values for different runs (specifically 20) of the algorithm. The y-axis measures the difference between

the cost function value at each iteration and the actual optimal value evaluated with the CVXPY package [DB16]. The x-axis shows the number of outer iterations of algorithms, i.e., the time needed to converge to the actual optimal value.

The numerical simulations demonstrate that all three algorithms indeed converge to their respective optimal values approximately. This numerical evidence suggests that all three proposed algorithms work well in practice. In other words, convergence to the optimal parameters in the presence of noise showcases noise resilience of our variational quantum algorithms.

4.6 Conclusion

In this paper, we proposed variational quantum algorithms for solving semidefinite programs. We considered three constrained formulations of SDPs, which were first converted to unconstrained forms by employing a series of reductions. When the dimension N of the input Hermitian operators of SDPs is large and for these unconstrained forms, the computation of the objective function's gradient is difficult when using known classical techniques. To address this problem, we utilized parameterized quantum circuits to estimate these gradients. We also established the convergence rate and total iteration complexity of one of our proposed VQAs. Finally, we numerically simulated our variational quantum algorithms for different instances of SDPs, and the results of these simulations provide evidence that convergence still occurs in noisy settings.

The estimation of the gradients using parameterized quantum circuits is stochastic in nature. In this paper, we assumed that we have unbiased estimators of these gradients, and the variance of these estimators is also small. Therefore, it remains open to study the

effect of the variance of these estimators on the convergence rate of our algorithms.

4.7 Acknowledgments

DP and MMW acknowledge support from the National Science Foundation under Grant No. 1907615. PJC acknowledges initial support from the Los Alamos National Laboratory (LANL) ASC Beyond Moore's Law project, and later support from the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under the Accelerated Research in Quantum Computing (ARQC) program. DP acknowledges Prof. Rahul Shah's helpful suggestions and support from the National Science Foundation under Grant No. 2137057.

BIBLIOGRAPHY

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, October 2019. doi:10.1038/s41586-019-1666-5.

[AK16] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM*, 63(2):1–35, May 2016. doi:10.1145/2837020.

[Ang23] Armando Angrisani. *The disparate impact of noise on quantum learning algorithms*. Theses, Sorbonne Université, December 2023.

[BCLK+22] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94:015004, February 2022. arXiv:2101.08448, doi:10.1103/RevModPhys.94.015004.

[Ber76] Dimitri P. Bertsekas. Multiplier methods: A survey. *Automatica*, 12(2):133–145, March 1976. doi:10.1016/0005-1098(76)90077-7.

[BHVK22] Kishor Bharti, Tobias Haug, Vlatko Vedral, and Leong-Chuan Kwek. Noisy intermediate-scale quantum algorithm for semidefinite programming. *Physical Review A*, 105:052445, May 2022. arXiv:2106.03891, doi:10.1103/PhysRevA.105.052445.

[BIS+22] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations, 2022. arXiv:1811.04968.

[BKL+19] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.27.

[BS17] Fernando G. S. L. Brandão and Krysta M. Svore. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE Computer Society, October 2017. doi:10.1109/FOCS.2017.45.

[BV04] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. doi:10.1017/CBO9780511804441.

[CAB+21] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon Benjamin, Suguro Endo, Keisuke Fujii, Jarrod Ryan McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, September 2021. doi:10.1038/s42254-021-00348-9.

[CCHL23] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. The complexity of NISQ. *Nature Communications*, 14(1):6001, 2023. doi:10.1038/s41467-023-41217-6.

[CvSW+21] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley. Efficient quantum measurement of Pauli operators in the presence of finite sampling error. *Quantum*, 5:385, 2021. doi:10.22331/q-2021-01-20-385.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[DDG+22] Marina Danilova, Pavel Dvurechensky, Alexander Gasnikov, Eduard Gorbunov, Sergey Guminov, Dmitry Kamzolov, and Innokentiy Shibaev. *Recent Theoretical Advances in Non-Convex Optimization*, volume 191, pages 79–163. Springer International Publishing, April 2022. arXiv:2012.06188, doi:10.1007/978-3-031-00832-0_3.

[DH96] Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum, July 1996. arXiv:quant-ph/9607014. arXiv:quant-ph/9607014.

[DISZ18] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018. arXiv:1711.00141.

[Eld03] Yonina C. Eldar. A semidefinite programming approach to optimal unambiguous discrimination of quantum states. *IEEE Transactions on Information Theory*, 49(2):446–456, February 2003. doi:10.1109/tit.2002.807291.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, November 2014. arXiv:1411.4028. arXiv:1411.4028.

[Gro97] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, July 1997. doi:10.1103/PhysRevLett.79.325.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995. doi:10.1145/227683.227684.

[HD21] Patrick Huembeli and Alexandre Dauphin. Characterizing the loss landscape of variational quantum circuits. *Quantum Science and Technology*, 6(2):025011, February 2021. doi:10.1088/2058-9565/abdbc9.

[Hes69] Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, November 1969. doi:10.1007/BF00927673.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, October 2009. doi:10.1103/PhysRevLett.103.150502.

[HRU+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. arXiv:1706.08500.

[HV03] Ramesh Hariharan and Vishwanathan Vinay. String matching in $O(n + m)$ quantum time. *Journal of Discrete Algorithms*, 1(1):103–110, November 2003. doi:10.1016/S1570-8667(03)00010-8.

[JGN+17] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1724–1732. PMLR, 2017.

[KE95] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, November 1995. doi:10.1109/ICNN.1995.488968.

[KNS16] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak–Lojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, Cham, Switzerland, September 2016. doi:10.1007/978-3-319-46128-1_50.

[KW20] Sumeet Khatri and Mark M. Wilde. Principles of quantum communication theory: A modern approach, 2020. arXiv:2011.04672v2.

[LL15] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[LS06] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *Society for Industrial and Applied Mathematics Journal on Optimization*, 1(2):166–190, July 2006. doi:10.1137/0801013.

[LYPS17] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. Hybrid quantum-classical approach to quantum optimal control. *Physical Review Letters*, 118(15):150503, April 2017. doi:10.1103/PhysRevLett.118.150503.

[MA21] Jakub Marecek and Albert Akhriev. A cutting-plane method for semidefinite programming with potential applications on noisy quantum devices, 2021. arXiv:2110.03400. arXiv:2110.03400.

[MHA20] Anirudha Majumdar, Georgina Hall, and Amir Ali Ahmadi. Recent

scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:331–360, 2020. doi:10.1146/annurev-control-091819-074326.

[MJS19] Eric V. Mazumdar, Michael I. Jordan, and S. Shankar Sastry. On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games, 2019. arXiv:1901.00838.

[MLZ+19] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra(-gradient) mile. In *International Conference on Learning Representations*, 2019. arXiv:1807.02629.

[MNKF18] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, September 2018. doi:10.1103/PhysRevA.98.032309.

[MP90] Bojan Mohar and Svatopluk Poljak. Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40(2):343–352, 1990.

[NM65] John A. Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965. doi:10.1093/comjnl/7.4.308.

[Par03] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003. doi:10.1007/s10107-003-0387-5.

[PCW24] Dhrumil Patel, Patrick J. Coles, and Mark M. Wilde. Variational quantum algorithms for semidefinite programming. *Quantum*, 8:1374, June 2024. doi:10.22331/q-2024-06-17-1374.

[PMS+14] 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*, 5:4213, July 2014. doi:10.1038/ncomms5213.

[Pow69] Michael J. D. Powell. A method for nonlinear constraints in minimization problems. *Optimization*, , pages 283–298, 1969.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. doi:10.22331/q-2018-08-06-79.

[PW00] Florian A. Potra and Stephen J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1):281–302, December 2000. doi:10.1016/S0377-0427(00)00433-7.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, December 1991. doi:10.1016/0022-0000(91)90023-X.

[RLLY22] Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, 37(3):1087–1121, 2022. arXiv:1810.02060, doi:10.1080/10556788.2021.1895152.

[SB20] Ivan Supic and Joseph Bowles. Self-testing of quantum systems: a review. *Quantum*, 4:337, September 2020. doi:10.22331/q-2020-09-30-337.

[SBG+19] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, March 2019. doi:10.1103/PhysRevA.99.032331.

[SEA+19] Mehmet Fatih Sahin, Armin Eftekhari, Ahmet Alacaoglu, Fabian Latorre Gómez, and Volkan Cevher. An inexact augmented Lagrangian framework for nonconvex optimization with nonlinear constraints. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13943–13955, 2019.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, New Mexico, USA, November 1994. IEEE Computer Society. doi:10.1109/SFCS.1994.365700.

[Sio58] Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, March 1958. doi:10.2140/pjm.1958.8.171.

[Spa92] James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. doi:10.1109/9.119632.

[T+23] Matthew Treinish et al. Qiskit: An open-source framework for quantum computing, 2023. https://doi.org/10.5281/zenodo.2573505. doi:10.5281/zenodo.2573505.

[vAGGdW20] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. *Quantum*, 4:230, February 2020. doi:10.22331/q-2020-02-14-230.

[Wan18] Xin Wang. *Semidefinite optimization for quantum information*. PhD thesis, University of Technology Sydney, Centre for Quantum Software and Information, Faculty of Engineering and Information Technology, July 2018.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. doi:10.1017/9781316848142.

[WXD18] Xin Wang, Wei Xie, and Runyao Duan. Semidefinite programming strong converse bounds for classical capacity. *IEEE Transactions on Information Theory*, 64(1):640–653, January 2018. doi:10.1109/tit.2017.2741101.

[YKL75] Horace Yuen, Robert Kennedy, and Melvin Lax. Optimum testing of multiple hypotheses in quantum detection theory. *IEEE Transactions on Information Theory*, 21(2):125–134, March 1975. doi:10.1109/TIT.1975.1055351.

APPENDIX

4.A Proof of Lemma 49

The proof is rather straightforward. First, let us consider a function with two parameters x and y, so that $f : \mathbb{R}^2 \to \mathbb{R}$. Suppose that the function $(\cdot) \to f(\cdot, y)$ is Lipschitz continuous with Lipschitz constant L_X, for all $y \in \mathbb{R}$, and suppose that the function $(\cdot) \to f(x, \cdot)$ is Lipschitz continuous with Lipschitz constant L_Y, for all $x \in \mathbb{R}$. Therefore, the following holds according to the definition of Lipschitz continuity (recall Definition 48):

$$|f(x, y) - f(x', y)| \leq L_X |x - x'| \quad \forall x, x', y \in \mathbb{R}, \tag{4.A.1}$$

$$|f(x, y) - f(x, y')| \leq L_Y |y - y'| \quad \forall x, y, y' \in \mathbb{R}. \tag{4.A.2}$$

Now consider the following:

$$\begin{aligned}
|f(x, y) - f(x', y')| &= |f(x, y) - f(x', y) + f(x', y) - f(x', y')| && (4.A.3)\\
&\overset{(a)}{\leq} |f(x, y) - f(x', y)| + |f(x', y) - f(x', y')| && (4.A.4)\\
&\overset{(b)}{\leq} L_X |x - x'| + L_Y |y - y'| && (4.A.5)\\
&\leq \max\{L_X, L_Y\} (|x - x'| + |y - y'|) && (4.A.6)\\
&\overset{(c)}{\leq} \sqrt{2} \max\{L_X, L_Y\} \|(x, y) - (x', y')\|, && (4.A.7)
\end{aligned}$$

where the inequality (a) follows from the triangle inequality, inequality (b) follows from (4.A.1)–(4.A.2), and the last inequality (c) follows from the fact that $\|\cdot\|_1 \leq \sqrt{2} \|\cdot\|$ for the two-variable case. Therefore, $L = \sqrt{2} \max\{L_X, L_Y\}$ is a Lipschitz constant for f.

The proof given above for two variables can be easily extended to a function f with an n-variable input, using the fact that $\|\cdot\|_1 \leq \sqrt{n} \|\cdot\|$, where $L = \sqrt{n} \max_i\{L_i\}_i$ is a Lipschitz

constant given in terms of

$$L_i = \sup_{\boldsymbol{x}} \left| \frac{\partial f(\boldsymbol{x})}{\partial x_i} \right|. \tag{4.A.8}$$

Here, $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$.

4.B Proof of Lemma 50

By hypothesis, each component f_i of the vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ is L_i-Lipschitz continuous. Hence, the following holds for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ and $i \in \{1, \ldots, m\}$:

$$\left| f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}') \right| \leq L_i \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|. \tag{4.B.1}$$

Now consider the following:

$$\left\| f(\boldsymbol{x}) - f(\boldsymbol{x}') \right\|^2 = \sum_{i=1}^{m} \left| f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}') \right|^2 \tag{4.B.2}$$

$$\leq \sum_{i=1}^{m} L_i^2 \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|^2 \tag{4.B.3}$$

$$= \left(\sum_{i=1}^{m} L_i^2 \right) \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|^2. \tag{4.B.4}$$

Hence, $L = \sqrt{\sum_{i=1}^{m} L_i^2}$ is a Lipschitz constant for the vector-valued function f.

4.C Proof of Lemma 58

According to Lemma 49, a Lipschitz constant for the multivariate function $h(\boldsymbol{\theta}) = \langle I \otimes O \rangle_{\boldsymbol{\theta}}$ is as follows:

$$L_h = \sqrt{r} \max_i \left\{ \sup_{\boldsymbol{\theta}} \left| \frac{\partial h(\boldsymbol{\theta})}{\partial \theta_i} \right| \right\}_i \tag{4.C.1}$$

$$\overset{(a)}{=} \sqrt{r} \max_i \left\{ \sup_{\boldsymbol{\theta}} \left| 2 \operatorname{Re} \left[\langle \mathbf{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i} | \mathbf{0} \rangle \right] \right| \right\}_i \tag{4.C.2}$$

$$\overset{(b)}{\leq} 2\sqrt{r} \max_i \left\{ \sup_{\boldsymbol{\theta}} \left| \langle \mathbf{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) U_f(\boldsymbol{\theta}) e^{-i\theta_i H_i} (-iH_i) U_b(\boldsymbol{\theta}) | \mathbf{0} \rangle \right| \right\}_i \tag{4.C.3}$$

$$\overset{(c)}{\leq} 2\sqrt{r} \max_i \left\{ \sup_{\boldsymbol{\theta}} \left\| U^\dagger(\boldsymbol{\theta}) (I \otimes O) U_f(\boldsymbol{\theta}) e^{-i\theta_i H_i} (-iH_i) U_b(\boldsymbol{\theta}) \right\| \right\}_i \tag{4.C.4}$$

$$\overset{(d)}{\leq} 2\sqrt{r} \max_i \left\{ \sup_{\boldsymbol{\theta}} \left\| U^\dagger(\boldsymbol{\theta}) \right\| \|O\| \left\| U_f(\boldsymbol{\theta}) \right\| \left\| e^{-i\theta_i H_i} \right\| \|-iH_i\| \|U_b(\boldsymbol{\theta})\| \right\}_i \tag{4.C.5}$$

$$\overset{(e)}{\leq} 2\sqrt{r} \|O\| \max_i \{\|H_i\|\}_i \,. \tag{4.C.6}$$

Here, equality (a) is a consequence of the following fact:

$$\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_i} = \langle \mathbf{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i} | \mathbf{0} \rangle + \langle \mathbf{0} | \frac{\partial U^\dagger(\boldsymbol{\theta})}{\partial \theta_i} (I \otimes O) U(\boldsymbol{\theta}) | \mathbf{0} \rangle, \tag{4.C.7}$$

where we used the product rule. Inequality (b) follows from the fact that $|\operatorname{Re}[z]| \leq |z|$ for all $z \in \mathbb{C}$. Additionally, we assume $U(\boldsymbol{\theta}) = U_f(\boldsymbol{\theta}) e^{-i\theta_i H_i} U_b(\boldsymbol{\theta})$, so that

$$\frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i} = U_f(\boldsymbol{\theta}) e^{-i\theta_i H_i} (-iH_i) U_b(\boldsymbol{\theta}). \tag{4.C.8}$$

Inequality (c) follows from the fact that $\|A\| = \sup_{|\psi\rangle, |\phi\rangle} \{ |\langle \psi | A | \phi \rangle| : \||\psi\rangle\| = \||\phi\rangle\| = 1 \}$. Inequality (d) follows from two properties of the spectral norm of a matrix: $\|AB\| \leq \|A\| \|B\|$ and $\|A \otimes B\| \leq \|A\| \|B\|$. The inequality (e) follows from the fact that $\|V\| = 1$ for every unitary V. Similarly, we can evaluate $L_{\nabla h}$ because each component of ∇h, i.e., $\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_i}$ is Lipschitz continuous. This is because according to the parameter-shift rule, we can write $\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_i}$ in terms of the linear combination of $h(\boldsymbol{\theta} + (\pi/4)\hat{\boldsymbol{e}}_i)$ and $h(\boldsymbol{\theta} - (\pi/4)\hat{\boldsymbol{e}}_i)$.

4.D Proof of Lemma 59

The multivariate vector-valued function $\nabla_{\boldsymbol{\theta}} \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$, where $\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$ is defined in (4.4.36)–(4.4.37), is $L_{c,\boldsymbol{y}}$-Lipschitz continuous for a fixed $\boldsymbol{y} \in \mathbb{R}^M$ if all its components, i.e., $\left\{ \frac{\partial \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})}{\partial \theta_i} \right\}_{i=1}^r$,

are Lipschitz continuous for a fixed $\boldsymbol{y} \in \mathbb{R}^M$. In order to prove that $\frac{\partial \mathcal{L}_c(\boldsymbol{\theta},\boldsymbol{y})}{\partial \theta_i}$ is $L^i_{c,\boldsymbol{y}}$-Lipschitz continuous, we need to bound the Lipschitz constant $L^i_{c,\boldsymbol{y}}$ from above. According to Lemma 49, we state the following:

$$L^i_{c,\boldsymbol{y}} = \sqrt{r} \max_j \left\{ \sup_{\boldsymbol{\theta}} \left| \frac{\partial^2 \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})}{\partial \theta_i \partial \theta_j} \right| \right\}_j \tag{4.D.1}$$

$$\leq \sqrt{r} \max_j \left\{ \sup_{\boldsymbol{\theta}} \left\{ \lambda \left| \frac{\partial^2 \langle I \otimes C \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right| + \lambda \sum_{m=1}^{M} |y_m| \left| \frac{\partial^2 \langle I \otimes A_m \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right| \right.\right.$$
$$+ \lambda c \sum_{m=1}^{M} \left(|b_m| \left| \frac{\partial^2 \langle I \otimes A_m \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right| + \lambda \left| \langle I \otimes A_m \rangle_{\boldsymbol{\theta}} \right| \left| \frac{\partial^2 \langle I \otimes A_m \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right| \right.$$
$$\left.\left.\left. + \lambda \left| \frac{\partial \langle I \otimes A_m \rangle_{\boldsymbol{\theta}}}{\partial \theta_i} \right| \left| \frac{\partial \langle I \otimes A_m \rangle_{\boldsymbol{\theta}}}{\partial \theta_j} \right| \right) \right\} \right\}_j , \tag{4.D.2}$$

where $j \in \{1, \ldots, r\}$, and the aforementioned inequality follows from the triangle inequality. Let us first bound $\sup_{\boldsymbol{\theta}} \left| \frac{\partial^2 \langle I \otimes O \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right|$ from above as follows, where $O \in \{C, A_1, \ldots, A_M\}$:

$$\sup_{\boldsymbol{\theta}} \left| \frac{\partial^2 \langle I \otimes O \rangle_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_j} \right| \tag{4.D.3}$$

$$\overset{(a)}{=} \sup_{\boldsymbol{\theta}} \left\{ \left| 2 \operatorname{Re} \left[\langle \boldsymbol{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) \frac{\partial^2 U(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} | \boldsymbol{0} \rangle \right] + 2 \operatorname{Re} \left[\langle \boldsymbol{0} | \frac{\partial U^\dagger(\boldsymbol{\theta})}{\partial \theta_i} (I \otimes O) \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_j} | \boldsymbol{0} \rangle \right] \right| \right\}$$

$$\overset{(b)}{\leq} 2 \sup_{\boldsymbol{\theta}} \left\{ \left| \operatorname{Re} \left[\langle \boldsymbol{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) \frac{\partial^2 U(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} | \boldsymbol{0} \rangle \right] \right| + \left| \operatorname{Re} \left[\langle \boldsymbol{0} | \frac{\partial U^\dagger(\boldsymbol{\theta})}{\partial \theta_i} (I \otimes O) \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_j} | \boldsymbol{0} \rangle \right] \right| \right\}$$

$$\overset{(c)}{\leq} 2 \sup_{\boldsymbol{\theta}} \left\{ \left| \langle \boldsymbol{0} | U^\dagger(\boldsymbol{\theta}) (I \otimes O) U_{ff}(\boldsymbol{\theta}) e^{-i\theta_j H_j} (-iH_j) U_{fb}(\boldsymbol{\theta}) e^{-i\theta_i H_i} (-iH_i) U_b(\boldsymbol{\theta}) | \boldsymbol{0} \rangle \right| \right.$$
$$\left. + \left| \langle \boldsymbol{0} | U_b^\dagger(\boldsymbol{\theta}) e^{i\theta_i H_i} (iH_i) U_f^\dagger(\boldsymbol{\theta}) (I \otimes O) U_{ff}(\boldsymbol{\theta}) e^{-i\theta_j H_j} (-iH_j) U_{b'}(\boldsymbol{\theta}) | \boldsymbol{0} \rangle \right| \right\}$$

$$\overset{(d)}{\leq} 4 \|O\| \left\| H_j \right\| \|H_i\| , \tag{4.D.4}$$

where equality (a) follows from the chain rule applied twice (see (4.C.7)). Inequality (b) follows from the triangle inequality. Inequality (c) follows from the fact that $|\operatorname{Re}[z]| \leq |z|$ for all $z \in \mathbb{C}$, as well as from the assumption that $U(\boldsymbol{\theta})$ has the following decomposition:

$$U(\boldsymbol{\theta}) = U_{ff}(\boldsymbol{\theta}) e^{-i\theta_j H_j} U_{fb}(\boldsymbol{\theta}) e^{-i\theta_i H_i} U_b(\boldsymbol{\theta}). \tag{4.D.5}$$

Additionally, let $U_{ff}(\boldsymbol{\theta})e^{-i\theta_j H_j}U_{fb}(\boldsymbol{\theta}) = U_f(\boldsymbol{\theta})$ and $U_{fb}(\boldsymbol{\theta})e^{-i\theta_i H_i}U_b(\boldsymbol{\theta}) = U_{b'}(\boldsymbol{\theta})$. Inequality (d) follows from a set of arguments similar to those in the proof of Lemma 58.

Second, we bound $\sup_{\boldsymbol{\theta}} |\langle I \otimes O\rangle_{\boldsymbol{\theta}}|$ from above as follows, where $O \in \{A_1, \ldots, A_M\}$:

$$\sup_{\boldsymbol{\theta}} |\langle I \otimes O\rangle_{\boldsymbol{\theta}}| = \sup_{\boldsymbol{\theta}} \left|\langle \mathbf{0}|U^\dagger(\boldsymbol{\theta})(I \otimes O)U(\boldsymbol{\theta})|\mathbf{0}\rangle\right| \tag{4.D.6}$$

$$\leq \|O\| \,. \tag{4.D.7}$$

Finally, we bound $\sup_{\boldsymbol{\theta}} \left|\frac{\partial\langle I\otimes O\rangle_{\boldsymbol{\theta}}}{\partial\theta_i}\right|$ from above as follows, where $O \in \{A_1, \ldots, A_M\}$:

$$\sup_{\boldsymbol{\theta}} \left|\frac{\partial\langle I \otimes O\rangle_{\boldsymbol{\theta}}}{\partial\theta_i}\right| = \sup_{\boldsymbol{\theta}} \left|2\,\mathrm{Re}\left[\langle \mathbf{0}|U^\dagger(\boldsymbol{\theta})(I \otimes O)\frac{\partial U(\boldsymbol{\theta})}{\partial\theta_i}|\mathbf{0}\rangle\right]\right| \tag{4.D.8}$$

$$\leq 2\,\|O\|\,\|H_i\| \,. \tag{4.D.9}$$

The above inequality follows from similar arguments made in the proof of Lemma 58.

Note that the upper bounds in (4.D.4), (4.D.7), and (4.D.9) are independent of $\boldsymbol{\theta}$. Hence, using these upper bounds, we bound $L^i_{c,\boldsymbol{y}}$ from above as follows:

$$L^i_{c,\boldsymbol{y}} \leq \sqrt{r}\max_j \Bigg\{4\lambda\,\|C\|\left\|H_j\right\|\|H_i\| + \lambda\sum_{m=1}^{M} 4\,|y_m|\,\|A_m\|\left\|H_j\right\|\|H_i\| \tag{4.D.10}$$

$$+ \lambda c\sum_{m=1}^{M}\left(4\,|b_m|\,\|A_m\|\left\|H_j\right\|\|H_i\| + 4\lambda\,\|A_m\|^2\left\|H_j\right\|\|H_i\| + 4\lambda\,\|A_m\|^2\left\|H_j\right\|\|H_i\|\right)\Bigg\}_j$$

$$= 4\lambda\,\sqrt{r}\,\|H_i\|\left(\|C\| + \sum_{m=1}^{M}\left((|y_m| + c\,|b_m|)\,\|A_m\| + 2c\lambda\,\|A_m\|^2\right)\right)\max_j\left\{\left\|H_j\right\|\right\}_j \tag{4.D.11}$$

From (4.D.11) we conclude that the Lipschitz constant $L^i_{c,\boldsymbol{y}}$ is bounded from above by a positive number. Hence, the Lipschitz constant $L_{c,\boldsymbol{y}}$ of $\nabla_{\boldsymbol{\theta}}\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$ is also bounded from above by a positive number because $L_{c,\boldsymbol{y}} = \sqrt{\sum_{i=1}^{r}\left(L^i_{c,\boldsymbol{y}}\right)^2}$ according to Lemma 50. Thus, the function $\mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{y})$ is $L_{c,\boldsymbol{y}}$-smooth for a fixed $\boldsymbol{y} \in \mathbb{R}^M$.

4.E Proof of Lemma 62

The multivariate vector-valued function $\nabla_{\boldsymbol{\theta}}\mathcal{F}_\gamma(\boldsymbol{\theta},\bar{\mathbf{y}})$ for a fixed $\bar{\mathbf{y}} \geq 0$, is $L_{\gamma,\bar{\mathbf{y}}}$-Lipschitz continuous if all its components, i.e., $\left\{\frac{\partial\mathcal{F}_\gamma(\boldsymbol{\theta},\bar{\mathbf{y}})}{\partial\theta_i}\right\}_{i=1}^{r}$, are Lipschitz continuous for a fixed $\bar{\mathbf{y}} \geq 0$. In order to prove that $\frac{\partial\mathcal{F}_\gamma(\boldsymbol{\theta},\bar{\mathbf{y}})}{\partial\theta_i}$ is $L^i_{\gamma,\bar{\mathbf{y}}}$-Lipschitz continuous, we first bound the Lipschitz constant $L^i_{\gamma,\bar{\mathbf{y}}}$ from above as follows:

$$L^i_{\gamma,\bar{\mathbf{y}}} = \sqrt{r}\max_j\left\{\sup_{\boldsymbol{\theta}}\left|\frac{\partial^2\mathcal{F}_\gamma(\boldsymbol{\theta},\bar{\mathbf{y}})}{\partial\theta_i\partial\theta_j}\right|\right\}_j \tag{4.E.1}$$

$$= b_M\sqrt{r}\max_j\left\{\sup_{\boldsymbol{\theta}}\left|\frac{e^{\gamma\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}}{(e^{\gamma\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}+1)}\frac{\partial^2\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i\partial\theta_j}\right.\right. \\ \left.\left. + \frac{e^{\gamma\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}}{(e^{\gamma\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}+1)^2}\gamma\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i}\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}\right|\right\}_j \tag{4.E.2}$$

$$\leq b_M\sqrt{r}\max_j\left\{\sup_{\boldsymbol{\theta}}\left|\frac{\partial^2\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i\partial\theta_j}+\gamma\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i}\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}\right|\right\}_j \tag{4.E.3}$$

$$\leq b_M\sqrt{r}\max_j\left\{\sup_{\boldsymbol{\theta}}\left|\frac{\partial^2\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i\partial\theta_j}\right|+\gamma\left|\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_i}\right|\left|\frac{\partial\langle I\otimes H(\bar{\mathbf{y}})\rangle_{\boldsymbol{\theta}}}{\partial\theta_j}\right|\right\}_j \tag{4.E.4}$$

$$\leq 4b_M\sqrt{r}\,\|H_i\|\left(\|H(\bar{\mathbf{y}})\|+4\gamma\,\|H(\bar{\mathbf{y}})\|^2\right)\max_j\left\{\left\|H_j\right\|\right\}_j. \tag{4.E.5}$$

The last inequality follows from (4.D.9) and (4.D.4). We see that $L^i_{\gamma,\bar{\mathbf{y}}}$ is bounded from above by a positive number for a fixed $\bar{\mathbf{y}}$. Therefore, the Lipschitz constant $L_{\gamma,\bar{\mathbf{y}}}$ is also bounded from above by a positve number because according to Lemma 50, we have $L_{\gamma,\bar{\mathbf{y}}} = \sqrt{\sum_{i=1}^{r}\left(L^i_{\gamma,\bar{\mathbf{y}}}\right)^2}$. Thus, the function $\mathcal{F}_\gamma(\boldsymbol{\theta},\bar{\mathbf{y}})$ is $L_{\gamma,\bar{\mathbf{y}}}$-smooth for a fixed $\bar{\mathbf{y}} \geq 0$.

CHAPTER 5

OPTIMAL COHERENT QUANTUM PHASE ESTIMATION VIA TAPERING[1]

5.1 Abstract

Quantum phase estimation is one of the fundamental primitives that underpins many quantum algorithms, including Shor's algorithm for efficiently factoring large numbers. Due to its significance as a subroutine, in this work, we consider the coherent version of the phase estimation problem, where given an arbitrary input state and black-box access to unitaries U and controlled-U, the goal is to estimate the phases of U in superposition. Most existing phase estimation algorithms involve intermediary measurements that disrupt coherence. Only a couple of algorithms, including the standard quantum phase estimation algorithm, consider this coherent setting. However, the standard algorithm only succeeds with a constant probability. To boost this success probability, one can employ the coherent median technique, resulting in an algorithm with optimal query complexity (the total number of calls to U and controlled-U). However, this coherent median technique requires a large number of ancilla qubits and a computationally expensive quantum sorting network.

To address this, in this work, we propose an improved version of this standard algorithm called the tapered quantum phase estimation algorithm. It leverages tapering/window functions commonly used in signal processing. Our algorithm achieves the optimal query complexity without requiring the expensive coherent median technique

[1]This chapter is based verbatim on the work [PTSS24], with minor corrections, an improved presentation, and an additional section on uncomputing. This work will soon be published in *PRX Quantum*; for the most up-to-date version of this chapter, refer to the arXiv preprint, which will be updated following publication.

to boost success probability. We also show that the tapering functions that we use are optimal by formulating optimization problems with different optimization criteria. Beyond the asymptotic regime, we also provide non-asymptotic query complexity of our algorithm, as it is crucial for practical implementation. In the appendices, we provide an efficient algorithm to prepare the quantum state corresponding to the optimal taper and an error bound for QPE when the phase estimate is used to control an operation and is subsequently uncomputed.

5.2 Introduction

Quantum phase estimation (QPE) has been central to the field of quantum computing since its introduction [Sho94, Kit95]. It has been used in Shor's algorithm for efficiently factoring large numbers [Sho94], in the Harrow–Hassidim–Lloyd (HHL) algorithm to solve a system of linear equations [HHL09], for quantum amplitude estimation [BHMT02], for quantum principal component analysis [LMR14], for fast Quantum Merlin-Arthur (QMA) amplification [NWZ09], and as a subroutine in many other applications [WBL12, LMR13, CSS18, WBD+21, ESP21, ALL+21, Ral20].

At its core, the goal of QPE is to estimate the phase of an eigenvalue of a given unitary. Let U be a unitary acting on a d-dimensional Hilbert space, $\mathcal{H}$, and suppose that $e^{2\pi i\theta}$ is one of the eigenvalues of U, where θ lies in the range $[0, 1)$. Then, the QPE problem is to output an estimate $\tilde{\theta}$ that is δ-close to θ with a probability at least $1 - \varepsilon$ for all $\delta, \varepsilon > 0$.

To solve this problem, it is common to assume that we are given black-box access to U and its controlled version (controlled-U), as well as sample access to an eigenvector $|\psi_\theta\rangle$ of U corresponding to the eigenvalue $e^{2\pi i\theta}$. For our purposes, we define the cost of an

algorithm in terms of its query complexity, which is the number of times U and controlled-U are applied. An upper bound on the gate complexity can be obtained from the query complexity for a given circuit implementation of the unitary by simple multiplication.

Probably the simplest approach to the phase estimation problem is the Hadamard test. It estimates the real and imaginary parts of the overlap $\langle\psi_\theta|U|\psi_\theta\rangle$, giving $e^{2\pi i\theta}$. This can then be used to approximate θ. However, this approach requires $O\left(\delta^{-2}\log(1/\varepsilon)\right)$ queries to achieve a desired precision, δ. A quadratic improvement in precision, i.e., $O\left(\delta^{-1}\log(1/\varepsilon)\right)$, can be achieved using the well-known algorithm proposed by Kitaev in 1995 [Kit95]. A crucial insight for this quadratic speedup was to extract the phase value bit-by-bit using controlled-U^{2^j}, rather than simply using controlled-U in each iteration like in the Hadamard test. Interestingly, the authors of Ref. [MdW23] recently showed that any algorithm solving the QPE problem requires $\Omega\left(\delta^{-1}\log(1/\varepsilon)\right)$ queries, making Kitaev's algorithm optimal in that sense.

In many practical applications of QPE, it is not realistic to assume that we have access to the exact eigenvector, $|\psi_\theta\rangle$. Instead, we have access to an arbitrary quantum state, $|\psi\rangle$, that can be written as a superposition of eigenvectors $\{|\psi_{\theta_r}\rangle\}_r$ of U:

$$|\psi\rangle = \sum_r c_r|\psi_{\theta_r}\rangle. \tag{5.2.1}$$

Suppose that the eigenvalues corresponding to these eigenvectors are $\left\{e^{2\pi i\theta_r}\right\}_r$. A more practical goal is then to estimate phases coherently in superposition, i.e., to prepare a state

$$\sum_r c_r\left(|\omega_r^\delta\rangle + |\omega_r^{\delta\perp}\rangle\right)|\psi_{\theta_r}\rangle \tag{5.2.2}$$

where

$$|\omega_r^\delta\rangle = \sum_{s:|\theta_r-\tilde{\theta}_s|\le\delta} c'_{r,s}|\tilde{\theta}_s\rangle \tag{5.2.3}$$

is the part of the ancilla state that encodes a δ-approximation to θ_r in the computational basis. For all r, we want $\||\omega_r^\delta\rangle\| \geq \sqrt{1-\varepsilon}$ which implies that a δ-approximation can be read out with probability $\geq 1-\varepsilon$. This coherent setting is more useful because most algorithms employ QPE as a subroutine, rather than as a standalone algorithm.

Both the Hadamard test and Kitaev's algorithm fall under the category of 'iterative' QPE algorithms. An iterative algorithm consists of multiple successive computations, each consisting of simple quantum circuits. Each iteration, however, consists of measurements at the end of the circuit evaluation followed by some classical post-processing. Applying these algorithms to an arbitrary state aside from an eigenvector of U causes decoherence due to these inherent measurements in each iteration. Ultimately, this leads to the preparation of an incoherent state as opposed to the pure state, given by (5.2.2).

To the best of our knowledge, only two algorithms exist in the literature that can perform phase estimation coherently [CEMM98, Ral21], with the standard QPE algorithm (also known as the textbook QPE algorithm) being the most well-known [CEMM98]. This algorithm incorporates the (inverse) quantum Fourier transform (QFT) as a subroutine and uses $O(\delta^{-1})$ queries to prepare the desired coherent state with probability at least $4/\pi^2$ in a single run. However, this constant success probability may not be sufficient for quantum algorithms that use QPE as a subroutine. Therefore, boosting this success probability to the desired $1-\varepsilon$ becomes extremely crucial. To achieve this, the standard QPE algorithm is executed $O(\log(1/\varepsilon))$ times in parallel, and the median of the outputs is computed coherently [NWZ09]. Therefore, the overall algorithm, i.e., the standard QPE combined with the coherent median computation for boosting success probability, has query complexity of $O(\delta^{-1}\log(1/\varepsilon))$. This query complexity is optimal, matching the corresponding lower bound $\Omega(\delta^{-1}\log(1/\varepsilon))$. However, the coherent median step involves using a large number of ancilla qubits and a quantum sorting network [HNS02, Kla03, BBG+13], which

is computationally expensive.

An alternative approach to boost the success probability, without using a sorting network, involves using $m = O(\log(1/\varepsilon))$ additional ancilla qubits all prepared in the uniform superposition state, but this increases the overall query complexity to $O\left(\delta^{-1}\varepsilon^{-1}\right)$ [CEMM98]. This is because the query complexity of the standard QPE algorithm grows exponentially with the number of these additional qubits used, i.e., $O\left(\delta^{-1}2^{m}\right)$. Substituting $m = O(\log(1/\varepsilon))$ into this yields the stated complexity. It is worth noting that this complexity is exponentially worse in ε compared to using the coherent median approach. Therefore, in this work, we investigate the following question and answer it positively: can we improve the standard QPE algorithm such that it maintains the optimal query complexity of $O(\delta^{-1}\log(1/\varepsilon))$ without employing the median approach and the associated quantum sorting network?

5.2.1 Contributions

We propose an improved version of the standard QPE algorithm, which we call the *tapered QPE (tQPE) algorithm*. The rationale behind this name will become apparent later in the discussion. We show that the tQPE algorithm uses exponentially fewer additional qubits, m, to achieve a success probability arbitrarily close to one. This improvement directly leads to an exponentially smaller query complexity in terms of ε.

To be more precise, we demonstrate that our tQPE algorithm requires only $m = \lceil \log_2 \log(1/\varepsilon) \rceil$ additional qubits to achieve a success probability at least $1-\varepsilon$ (see Sec. 5.5.3). We obtain this improvement by framing the problem as an optimization over the choice of the ancilla qubit state that maximizes the success probability. This results in a more

effective choice for the initial state of the ancilla register. In the case of the standard QPE algorithm, the ancilla register is initialized to the uniform superposition state. However, we propose to replace this state with the state corresponding to the most-frequency-concentrated discrete prolate spheroidal sequence (DPSS), which is a widely used *window/tapering* function in the field of classical signal processing. This motivates the name of our algorithm. We will provide a brief note on DPSS and on window/tapering functions in general later in Sec. 5.3.

The key idea behind this substitution is that DPSS maximizes signal concentration within a given spectral band. This is particularly important for QPE, as it helps in maximizing the probability of obtaining phase estimates that are δ-close to the true phase θ. To this end, the query complexity of our algorithm is then $O\left(\delta^{-1}\log(1/\varepsilon)\right)$, which scales exponentially better in terms of ε than that of the standard QPE algorithm, which, as mentioned before, is $O\left((\delta\varepsilon)^{-1}\right)$. Furthermore, the query complexity of our algorithm saturates the lower bound $\Omega\left(\delta^{-1}\log(1/\varepsilon)\right)$, which means that it is optimal. Having said that, we would also like to emphasize one of the implications of our result: tQPE can be used directly for fast QMA amplification instead of running the standard QPE algorithm $O(\log(1/\varepsilon))$ times for computing the median. This suggests an alternative approach for fast QMA amplification that employs tapering functions.

A natural question following the aforementioned result is whether one can initialize the ancilla register to this DPSS state efficiently. We answer this question by providing an explicit algorithm for doing so, which we describe in detail in App. 5.A. We subsequently show that the gate complexity of this algorithm is comparable to that of the standard QPE algorithm for initializing the uniform superposition state, up to log log factors.

The tQPE algorithm, like the standard QPE algorithm, involves applying the inverse

QFT to the ancilla qubits. However, since tQPE uses exponentially fewer additional qubits, the inverse-QFT circuit is also smaller compared to that used in the standard algorithm. This can be seen as follows. In Ref. [HH00], the authors showed that the gate complexity of the inverse QFT acting on p qubits is $O(p \log p)$. Therefore, the gate complexity of the inverse QFT in the case of the standard algorithm is

$$O\left(\left(\log \frac{1}{\delta} + \log \frac{1}{\varepsilon}\right) \log\left(\log \frac{1}{\delta} + \log \frac{1}{\varepsilon}\right)\right)$$

because the number of ancilla qubits required for this algorithm is $p = O(\log(1/\delta) + \log(1/\varepsilon))$. On the other hand, the gate complexity of this transform in the case of tQPE is

$$O\left(\left(\log \frac{1}{\delta} + \log\log \frac{1}{\varepsilon}\right) \log\left(\log \frac{1}{\delta} + \log\log \frac{1}{\varepsilon}\right)\right)$$

because the ancilla qubits required for this algorithm is $p = O(\log(1/\delta) + \log\log(1/\varepsilon))$, a significant improvement.

We also study a special scenario where one is not allowed to use any additional qubits, i.e., $m = 0$ (see Sec. 5.5.2) and one wishes to maximize the success probability of outputting one of the two nearest estimates when θ lies exactly in between two phase estimates. The optimal input states for the ancilla qubits for this scenario define a two dimensional subspace which includes a particular kind of sinusoidal sequence, defined in (5.5.9). Additionally, we perform numerics, plotting the success probability as a function of the distance between the true value and the closest phase estimate (see Sec. 5.5.5). We carry out these numerics for three different input states of the ancilla register: 1) the uniform superposition state, 2) the DPSS state, and 3) the above sinusoidal state. This analysis demonstrates that DPSS performs well over the entire range as compared to the uniform superposition state.

In Appendix 5.G we address the issue of coherent uncomputation of the phase estimate. In many applications such as HHL algorithm [HHL09] and Quantum Metropolis

Sampling [TOV+11], QPE is used to implement a unitary that is controlled by the phase estimate which is subsequently uncomputed. In Theorem 75 we prove an error bound on such uses of QPE that applies to any algorithm for coherent QPE. We anticipate that this bound will be useful for analyzing such algorithms as it accounts for multiple errors that in previous analyses had to be treated individually.

5.2.2 Overview

Let p be the number of ancilla qubits in the QPE circuit, initialized to the all-zeros state. For simplicity, we assume that we have sample access to one of the eigenvectors, $|\psi_\theta\rangle$, of U. In tQPE, as shown in Fig. 5.2, there are three main steps:

1. Initialize the ancilla qubits to a state $|\phi\rangle$. For now, this state serves as a placeholder and will be replaced later with a specific state depending on the problem at hand.
2. Apply unitaries controlled-U^{2^j} to the joint state, $|\phi\rangle|\psi_\theta\rangle$.
3. Apply inverse QFT to the ancilla qubits.

Notably, the standard QPE algorithm is a special case of tQPE, where the ancilla qubits are initialized to the uniform superposition state, i.e.,

$$|\phi\rangle = \frac{1}{2^{p/2}} \sum_i |i\rangle. \tag{5.2.4}$$

The application of the above three steps leads to the following transformation:

$$|0^{\otimes p}\rangle|\psi_\theta\rangle \xrightarrow{\text{tQPE}} \sum_{k=0}^{2^p-1} \hat{\phi}\left(\theta - \frac{k}{2^p}\right)|k\rangle|\psi_\theta\rangle, \tag{5.2.5}$$

where $\hat{\phi}(\cdot)$ is the discrete-time Fourier transform, defined explicitly in (5.4.6), of $|\phi\rangle$.

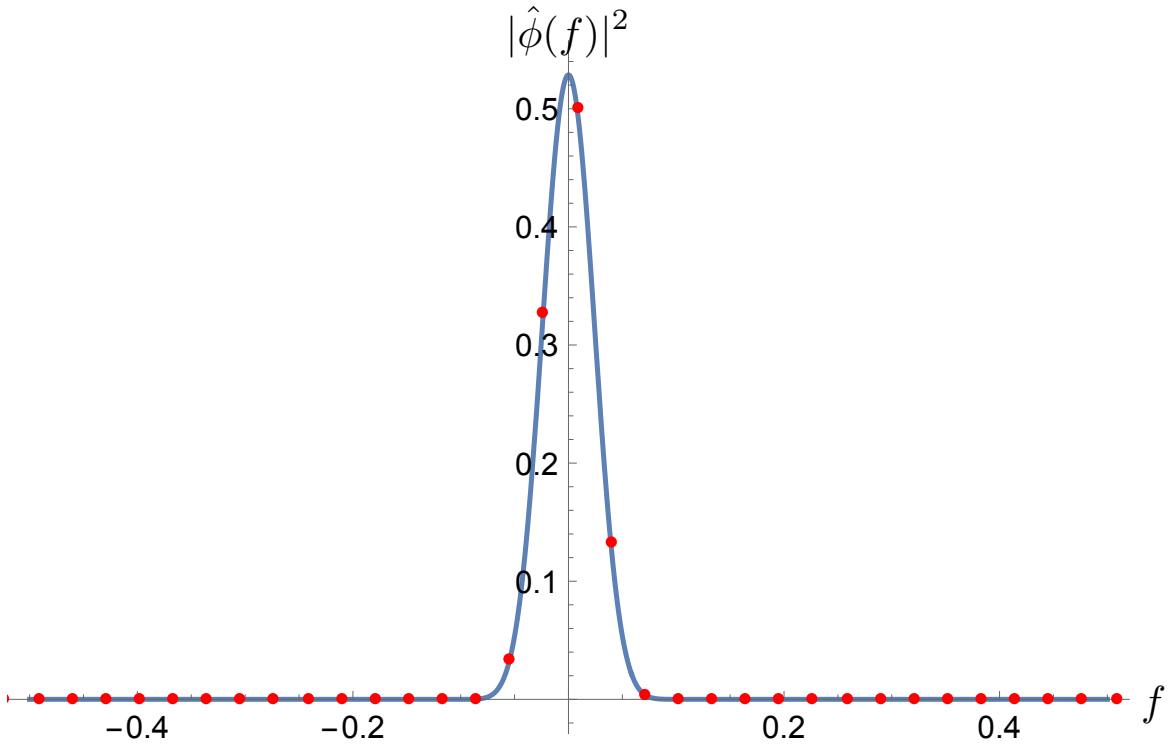


Figure 5.1: For the DPSS taper with $p = 5$, we plot the absolute value squared of the taper in the frequency domain (blue curve) and the discrete values (red dots) at which it is evaluated for the expression of success probability in (5.2.6).

As we can see from the above transformation, the final state of the ancilla register is a superposition of all possible values of phase estimates, $k/2^p$. The goal is to prepare this final state such that the following holds:

$$\sum_{k:|\theta - k/2^p| \leq \delta} \left| \hat{\phi}\left(\theta - \frac{k}{2^p}\right) \right|^2 \geq 1 - \varepsilon. \tag{5.2.6}$$

Intuitively, we want to choose a state, $|\phi\rangle$, of the ancilla register whose energy in the frequency domain is concentrated within the frequency band $[-\delta, \delta]$ and is at least $1 - \varepsilon$. Fig. 5.1 shows an example of a taper whose energy is concentrated around zero in its frequency domain.

We study error in two different settings: average- and worst-case. In the average-case error setting, we formulate an optimization problem that aims to find ancilla states, $|\phi\rangle$, that minimize the average error of the tQPE algorithm when the phases are sampled uniformly at random. We show that solving these optimization problems is equivalent to solving frequency concentration problems arising in classical signal processing and the most-frequency-concentrated DPSS naturally emerges as the optimal solution in the average-case setting. Furthermore, we show that this DPSS also exhibits optimal performance with only a constant overhead in the worst-case setting.

5.2.3 Related Work

QPE is a well-researched problem and hitherto many quantum algorithms have been proposed to solve it under different settings. However, most of the quantum algorithms in the literature [Kit95, SHF13, NLY23, LT22, OTT19, WBC22, CBB20, DL23, GSP21, MdW23] are not suitable for the coherent setting that we are considering in this paper. This point is strongly emphasized in Ref. [Ral21]. As such, only two algorithms, such as those presented in Ref. [CEMM98] and Ref. [Ral21], provide coherent phase estimation algorithms and are thus suitable for use as subroutines in larger algorithms. The approach in Ref. [Ral21] is complementary to ours and achieves similar query complexity. Rall's algorithm employs block-encoding techniques to obtain the phase estimate bit-by-bit and employs techniques from the quantum singular value transform (QSVT) [GSLW19, MRTC21] to enhance the probability of success.

Tapers are not new in the study of quantum algorithms. In 1999, Bužek *et al.* used a sinusoidal taper for the construction of optimal quantum clocks [BDM99]. The authors of [BW00] found that a particular kind of sinusoidal taper is optimal as an N-photon

two-mode input state for obtaining an estimate of the phase difference between two arms of an interferometer. The effect of a cosine taper on the quantum phase estimation has been studied in Ref. [RIK22], and the authors showed a cubic improvement of m in terms of error probability ε, i.e., $m = O(\log(1/\varepsilon^{1/3}))$ as opposed to $m = O(\log(1/\varepsilon))$ for the standard QPE algorithm. Additionally, the cosine taper has been employed in QPE-based algorithms, including the HHL algorithm for solving systems of linear equations [HHL09]. Bump functions have found application in spectral estimates based on time series analysis [Som19]. Tapers have also been explored within quantum spectral filtering methods aimed at efficient state initialization [FGML17]. Furthermore, approximate Gaussian tapers have been utilized in quantum algorithms for spectral density estimation [Rog20] and thermal state preparation [CKBG23, CKG23]. The Kaiser taper [CF+48, Kai66, KRD21, MGB22] has been considered in QPE [BSG+22] and achieves nearly asymptotically optimal scaling. We will further comment on Kaiser tapers and their relation to the DPSS taper in Sec. 5.6. Note that none of the tapers discussed above are optimally frequency-concentrated, resulting in side-lobes that warrant further reduction.

Note added: During the completion of this manuscript, we became aware of a recent study conducted by Greenaway *et al.*, concluding that QPE performed with the Kaiser taper significantly outperformed QSVT-QPE [Ral21] (QPE performed with the QSVT framework) [GPS24]. In particular, they observed that QPE with Kaiser tapers achieved success probabilities that are orders of magnitude higher than QSVT-QPE with better query complexity in the highly relevant regime where the desired success probability is arbitrarily close to 1. This finding further emphasizes the importance of utilizing tapers for the QPE problem.

5.3 A Brief Note on Discrete Prolate Spheroidal Sequences

In the field of classical signal processing and statistics, DPSS, sinusoidal sequences, and other such sequences are referred to as window functions, tapering functions, or simply tapers. These functions are widely used to analyze and modify the frequency spectrum of a given signal. It is important to note that the uniform superposition state used in the standard QPE algorithm is also a type of window function known as a rectangular window. This function is also sometimes referred to as the *tophat* taper, and we will use this name throughout our paper to refer to this taper.

In a series of seminal papers in the field of signal analysis [SP61, LP61, LP62, Sle64, Sle78], Slepian, Pollack, and Landau studied the extent to which a time-limited signal can be band-limited. In other words, they investigated how much the Fourier transform of a signal can be concentrated in a small interval in the frequency domain (i.e., band-limited), given that the signal is only non-zero in a finite interval in the time domain (i.e., time-limited). The discrete-time case was studied in the fifth paper of this series of papers [Sle78]. In this paper, Slepian introduced DPSSs (also called Slepian sequences) as eigenvectors of a kernel arising from a particular frequency concentration algorithm and demonstrated that there exists a DPSS that is both time-limited and maximally band-limited.

For completeness, we provide a detailed derivation of this in App. 5.E. In brief, Slepian constructed and optimized a cost function to find a discrete sequence, whose Fourier transform is maximally concentrated in a given bandwidth. The optimization results in a linear kernel, whose maximal eigenvector (i.e., the eigenvector with the largest eigenvalue) solves the problem. They additionally showed how to efficiently compute it numerically [PW93].

Due to the ability of DPSSs to concentrate the energy of the Fourier transformed function in a small interval in the frequency domain, while also maintaining finite support in the time domain, it is well-suited for a wide variety of applications in signal processing, including signal filtering, and high-resolution spectral and harmonic analysis methods.

Classical signal analysis methods based on Slepian *et al.*'s analysis have resulted in significant advances in spectral and harmonic analysis [Sle78, Tho82, PLVI87, MP99, HTR09, SY12]. Maximizing signal concentration within a given spectral band is not only desirable for classical signal analysis but is also of special significance for QPE as we will see in this paper. Intuitively, the band-limiting property of a taper is important especially for QPE in the sense that it helps in increasing the probability of outputting phase estimates that are δ-close to the true phase θ.

5.4 Tapered Quantum Phase Estimation

Consider a unitary, U, acting on a d-dimensional Hilbert space, $\mathcal{H}$. Let $\{|\psi_{\theta_r}\rangle\}_{r=0}^{d-1}$ be a set of orthogonal eigenvectors of U, and $\left\{e^{2\pi i\theta_r}\right\}_{r=0}^{d-1}$ be the set of corresponding eigenvalues, where $\theta_r \in [0,1)$ for all $r \in \{0,\ldots,d-1\}$. It is assumed that we are given query access to U and controlled-U, as well as sample access to an arbitrary state, $|\psi\rangle = \sum_r c_r|\psi_{\theta_r}\rangle$, or alternatively, query access to the unitary that prepares $|\psi\rangle$. Given $\delta,\varepsilon > 0$, the goal of coherent phase estimation is then to prepare the state $\sum_r c_r|\omega_r^\delta\rangle|\psi_{\theta_r}\rangle$ with probability at least $1-\varepsilon$ such that $|\omega_r^\delta\rangle = \sum_s c'_{r,s}|\tilde{\theta}_{r,s}\rangle$ and $\tilde{\theta}_{r,s}$ is δ-close to θ_r.

In what follows, we focus on a particular eigenvector $|\psi_\theta\rangle$ of U with eigenvalue $e^{2\pi i\theta}$ for simplicity and lay out the details of our tQPE algorithm. This approach can be readily extended to a superposition of eigenvectors as well. To begin with, as shown in Fig. 5.2,

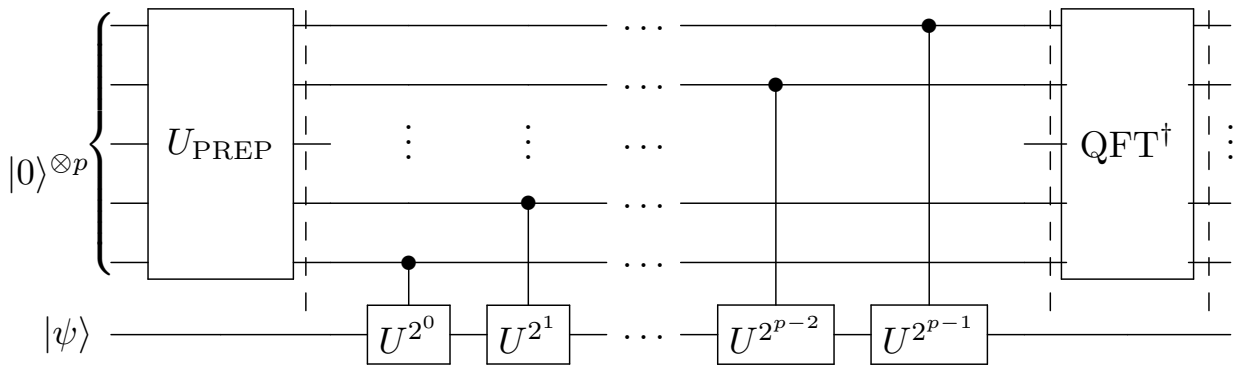


Figure 5.2: Tapered QPE quantum circuit. The system is initialized in the state $|0\rangle^{\otimes p}|\psi\rangle$, where $|\psi\rangle$ is an arbitrary input state. U_{PREP} is the unitary that prepares the taper state $|\phi\rangle$ (see App. 5.A for U_{PREP} that prepares the DPSS taper approximately). The dashed lines indicate the intermediate states corresponding to (5.4.1), (5.4.3), and (5.4.7), in order from left to right.

we prepare an ancilla register in a state $|\phi\rangle$ defined as

$$|\phi\rangle := \sum_{n=0}^{N-1} \phi[n]|n\rangle, \tag{5.4.1}$$

where $\phi[n] \in \mathbb{C}$ for all $n \in \{0, \ldots, N-1\}$. Also, $N := 2^p$, where p is the total number of qubits in the ancilla register. Throughout this paper, we refer to the state of the ancilla register, $|\phi\rangle$, as a taper, and we use the terms "taper register" and "ancilla register" interchangeably. For the time being, please note that $|\phi\rangle$ is an arbitrary state, with a particular choice specified later. We then apply the controlled unitary

$$\sum_{n=0}^{N-1} |n\rangle\langle n| \otimes U^n \tag{5.4.2}$$

to $|\phi\rangle|\psi_\theta\rangle$, resulting in the state

$$\sum_{n=0}^{N-1} \phi[n] e^{2\pi i \theta n} |n\rangle|\psi_\theta\rangle. \tag{5.4.3}$$

Subsequently, we apply the inverse QFT ($\mathrm{QFT}^{-1} = \mathrm{QFT}^\dagger$) on the ancilla register. This

transforms the basis $\{|n\rangle\}_{n=0}^{N-1}$ as

$$|n\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i n k/N} |k\rangle, \tag{5.4.4}$$

giving us the following final state:

$$\sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \phi[n] e^{2\pi i n(\theta - k/N)} \right) |k\rangle |\psi_\theta\rangle. \tag{5.4.5}$$

Note that the expression above inside the parentheses is the discrete-time Fourier transform, defined as

$$\hat{\phi}(f) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \phi[n] e^{2\pi i n f}, \tag{5.4.6}$$

of the time-limited signal, $\phi[n]$, evaluated at the frequency, $\theta - k/N$. Using this, the final state of the algorithm, given by (5.4.5), can be expressed more concisely as:

$$\sum_{k=0}^{N-1} \hat{\phi}(\theta - k/N) |k\rangle |\psi_\theta\rangle, \tag{5.4.7}$$

where we can think of $\left\{\hat{\phi}(\theta - k/N)\right\}_k$ as the probability amplitudes of the discrete frequencies $\{k/N\}_k$ indexed by k. In other words, if the first register corresponding to the taper is measured, the tQPE algorithm outputs a phase estimate, k/N, with probability $\left|\hat{\phi}(\theta - k/N)\right|^2$.

In App. 5.F, we show that the coefficient in (5.4.7) may be understood as the convolution

$$\hat{\phi}(\theta - f) = \left(\sum_{n=0}^{N-1} \phi[n] e^{-2\pi i n f} \right) * \delta(\theta - f)\,, \tag{5.4.8}$$

evaluated at $f = k/N$. That is, the taper is centered about, and independent of, the frequency θ. Conceptually, and important for our discussion below, this distinguishes the taper as a variable function that can take various forms depending on the particular optimization problem of interest.

Recall that the goal of tQPE is to prepare the final state, given by (5.4.7), such that the following holds:

$$\sum_{k:|\theta-k/N|\leq\delta} \left|\hat{\phi}\left(\theta - k/N\right)\right|^2 \geq 1-\varepsilon. \tag{5.4.9}$$

To simplify the analysis, we choose $\delta = 2^{-\ell-1}$ for some positive integer ℓ without loss of generality. If θ can be expressed exactly on ℓ bits, then there exists a $k \in \{0, 1, \ldots, N-1\}$ such that $\theta = k/N$. Otherwise, there is always a phase estimate k'/N such that $|\theta - k'/N| \leq 2^{-p-1}$, and a θ that saturates this bound (i.e., it lies exactly between two possible phase estimates). This constraint naturally arises because the resolution of the ancilla register is 2^{-p}. Therefore, we only consider the case where $\delta \geq 2^{-p-1}$ which implies that $p \geq \ell$. Thus, we set $p = \ell + m$ for some $m \geq 0$ where m can be interpreted as the number of additional qubits used to boost the success probability of estimating phases that are δ-close to θ.

As we can see, the standard QPE algorithm is a special case of the tQPE algorithm, where the taper being used is the tophat taper ($\phi[n] = 1/\sqrt{N}; \forall \mathrm{n}$). For $m = 0$ (i.e., no additional qubits used to boost phase estimation success probability), the tQPE algorithm using the tophat taper outputs a δ-close phase estimate with probability at least $4/\pi^2 \approx 0.405$. As previously stated in the Introduction section, in order to increase the success probability further, a natural approach is to increase m. By doing so, the number of δ-close phase estimates increases, which in turn increases the total probability of outputting such estimates. Specifically, the authors of Ref. [CEMM98] showed that it is sufficient to choose $m = \lceil \log_2(1/(2\varepsilon) + 1/2) \rceil$ to boost the success probability to at least $1-\varepsilon$ when considering the tophat taper.

Although increasing m can reduce the probability of outputting a phase estimate that differs from the true value by more than δ, it also increases the computational cost of the algorithm, since the number of queries to U is proportional to $2^p = 2^{l+m}$ (as shown in

Fig. 5.2). It also increases the size of the inverse-QFT circuit because as mentioned above, the size of this circuit is of the order $O(p \log p)$. Thus, it is of utmost importance to choose our taper, $|\phi\rangle$, optimally so as to minimize m for a given ε.

5.5 Optimal Tapers

In what follows, we derive optimal tapers for the case where the relative position of the true phase with respect to the grid of estimates is known and optimal tapers for the average-case settings which we define later. We then study the worst-case error of these tapers.

5.5.1 The Optimization Problem

We now state the optimization problem more formally:

$$\begin{aligned} \max_{|\phi\rangle \in \mathcal{H}_t} \quad & \sum_{k:\, \left|\theta - \frac{k}{N}\right| \leq \delta} \left| \hat{\phi}\left(\theta - \frac{k}{N}\right) \right|^2 \\ \text{subject to} \quad & \sum_{n=0}^{N-1} |\phi[n]|^2 = 1. \end{aligned} \tag{5.5.1}$$

Here, $\mathcal{H}_t$ denotes the N-dimensional Hilbert space corresponding to the taper register. The objective function above represents the probability of obtaining a phase estimate that is δ-close to the exact value, θ. The constraint arises from the fact that $|\phi\rangle$ is a quantum state, and therefore, it must be normalized.

Due to our choice of $\delta = 2^{-\ell-1}$ and $p = \ell + m$, there are generically 2^m estimates that are δ-close to θ. In the special cases when the phase is on a grid point or exactly between

two grid points there are $2^m + 1$ and $2^m + \delta_{m0}$ estimates that are δ-close to θ, respectively. Depending on the exact value of θ, the index, k, in the objective function of (5.5.1) can range over different values. This significantly complicates the optimization problem. In order to simplify the above expression further, we define the following:

$$\Delta := \theta - \frac{k^*}{N}, \tag{5.5.2}$$

where k^*/N is the phase estimate that is closest to θ on the grid. In other words, Δ is the difference between the best possible phase estimate afforded by the p-qubit taper and the true value, θ. We can now rewrite $\hat{\phi}(\theta - k/N)$ as $\hat{\phi}(\Delta - k/N)$ since the sum is over a dummy index. We observe that phase estimates corresponding to $|k| \leq K$ with $K = 2^{m-1} - 1$ are always δ-close for $m \geq 2$ and $K = 0$ for $m = 0$. This generically leaves out one phase estimate that is furthest away from the true value. In the special cases when the phase is on a grid point or exactly between two grid points the miscount is $2 - 2\delta_{m0}$ and 1, respectively. Finally, all tapers of practical interest we know of have Fourier transforms that are symmetric, have a peak at 0, and decay rapidly, meaning their value at $\Delta \pm (K+1)$ will be extremely small for large K or equivalently small ε. In light of this, we ignore the contribution of those points to the success probability in the optimization problem.

With these considerations, we restate the optimization problem as:

$$\begin{aligned} \max_{|\phi\rangle \in \mathcal{H}_t} \quad & \sum_{k=-K}^{K} \left| \hat{\phi}\left(\Delta - \frac{k}{N}\right) \right|^2 \\ \text{subject to} \quad & \sum_{n=0}^{N-1} |\phi[n]|^2 = 1. \end{aligned} \tag{5.5.3}$$

To convert the above constrained optimization problem to an unconstrained one, we use the Lagrangian formulation:

$$\mathcal{L}(|\phi\rangle, \lambda) = \sum_{j=-K}^{K} \left| \hat{\phi}\left(\Delta - \frac{j}{N}\right) \right|^2 - \lambda \left(\sum_{n=0}^{N-1} |\phi[n]|^2 - 1 \right). \tag{5.5.4}$$

where $\mathcal{L}$ is the Lagrangian and $\lambda \in \mathbb{R}$ is a Lagrange multiplier.

5.5.2 Ideal Case

Finding the optimal taper is equivalent to finding the stationary point of $\mathcal{L}$ that maximizes the objective function of (5.5.3). The stationary points of $\mathcal{L}$ can be found by setting all the partial derivatives of $\mathcal{L}$ to zero. Doing so, we get the following two conditions (see App. 5.B for a detailed derivation):

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{2\pi i\Delta(n-m)}\left(\frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\sin\left(\pi(m-n)/N\right)}\right)\phi[n] = \lambda\phi[m]; \quad \forall m \in \{0,\ldots,N-1\}, \tag{5.5.5}$$

$$\sum_{n=0}^{N-1} |\phi[n]|^2 = 1. \tag{5.5.6}$$

The first equation is an eigenvalue equation, while the second equation is the normalization constraint. By substituting all the stationary points $(|\phi\rangle, \lambda)$ satisfying the above two conditions into the objective function of (5.5.3), we obtain:

$$\sum_{j=-K}^{K} \left|\hat{\phi}\left(\Delta + \frac{j}{N}\right)\right|^2 = \lambda. \tag{5.5.7}$$

This implies that the stationary point, $(|\phi\rangle, \lambda)$, that maximizes the objective function is the eigenvector with maximum eigenvalue, λ. Since the objective function is the probability of outputting one of the $2K+1$ phase estimates closest to θ, using the eigenvector with the maximum eigenvalue (see (5.5.5)) as our taper will maximize this probability.

In App. 5.C, we establish an explicit connection between the eigenvectors of (5.5.5) and the periodic discrete prolate spheroidal sequences (P-DPSS) [ZKD+17]. We call the former quantum periodic discrete prolate spheroidal sequences (QP-DPSS), which depend on the value of Δ. Then, combining results from App. 5.C with known results for the P-DPSS in [XC84], we observe that the eigenvector of (5.5.5) with the largest eigenvalue has an eigenvalue of 1, regardless of the value of Δ. In other words, there exists a taper for which the tQPE algorithm outputs one of the δ-close $2K+1$ phase estimates with probability 1.

It should not be surprising that the above observation holds true. To understand this intuitively, we can break it down into two cases. In the first case, we consider $\Delta = 0$, that is θ lies exactly on one of the grid points (see (5.5.2)). In this case, the standard QPE algorithm using the tophat taper always returns θ with probability 1 because now θ itself is one of the possible phase estimates. Now, let's consider the case where $0 < \Delta \leq \frac{1}{2N}$ or $-\frac{1}{2N} \leq \Delta < 0$. This case can be converted into the $\Delta = 0$ case by shifting the grid of possible phase estimates by Δ. To accomplish this, we apply a unitary operator parameterized by Δ to the tophat taper. This operation shifts all 2^p grid points by Δ, such that θ now lies exactly on a grid point. After that, we output θ with probability 1 because it is now a possible phase estimate.

We find that even for $K = 0$, unit success probability can be achieved with the optimal taper given by:

$$\phi^{\Delta}[n] = \frac{e^{2\pi i \Delta n}}{\sqrt{N}} \,. \tag{5.5.8}$$

As a check, we note that for $\Delta = 0$, we recover the tophat taper which is known to have zero error probability when the true phase is on the grid of estimates. In Fig. 5.3 we plot the success probability of this taper for $\Delta = \pm 1/(2N)$. The above procedure is described in greater detail in App. 5.C.

Special Case: $\Delta = \pm 1/2N$

In this case, there are pairs of grid points that are equidistant to the exact solution. So, it makes sense to consider the probability of outputting an even number of estimates as opposed to an odd number. More specifically, we will focus on the case where we only consider the nearest two estimates. Eq. (5.5.8) shows tapers that return the closest estimate

with probability one for arbitrary Δ. In the special case $\Delta = \pm 1/2N$, there are actually two such tapers: one shown in Eq. (5.5.8) and the other obtained by letting $\Delta \to -\Delta$. One of these tapers outputs the larger and the other the smaller of the two closest estimates with unit probability. Any linear combination of these tapers also succeeds with unit probability. One might be interested in combinations that output each closest estimate with equal probability. One such pair of tapers is

$$\phi^{\sin}[n] = \frac{\sin(\pi n/N)}{\sqrt{N/2}} , \tag{5.5.9}$$

$$\phi^{\cos}[n] = \frac{\cos(\pi n/N)}{\sqrt{N/2}} . \tag{5.5.10}$$

Although these two tapers perform the same on $\Delta = \pm 1/2N$, their performance on other Δ is very different. In Fig. 5.3 we show that the sine taper is superior to the cosine taper because it performs better at all other values of Δ. Note that $\phi^{\pm 1/2N}$ has similar properties to the tophat taper by construction but the sine taper behaves qualitatively differently. This is because in constructing the sine taper we used the additional degree of freedom afforded by taking a linear combination of $\phi^{\pm 1/2N}$ to improve the performance of the taper over all Δ. In fact, the sine (cosine) taper can be obtained by minimizing (maximizing) the average-case error defined in the next section among all linear combinations between the two QP-DPSS tapers above.

5.5.3 Average-Case Optimal Tapers

From the development above, it is important to note that in general the value of Δ is not known *a priori*. Moreover, since Δ depends on the phase θ, it can differ for distinct phases, particularly when the input state is a superposition of eigenvectors of U. Since we are interested in coherent phase estimation, the same taper must work well for all values of Δ

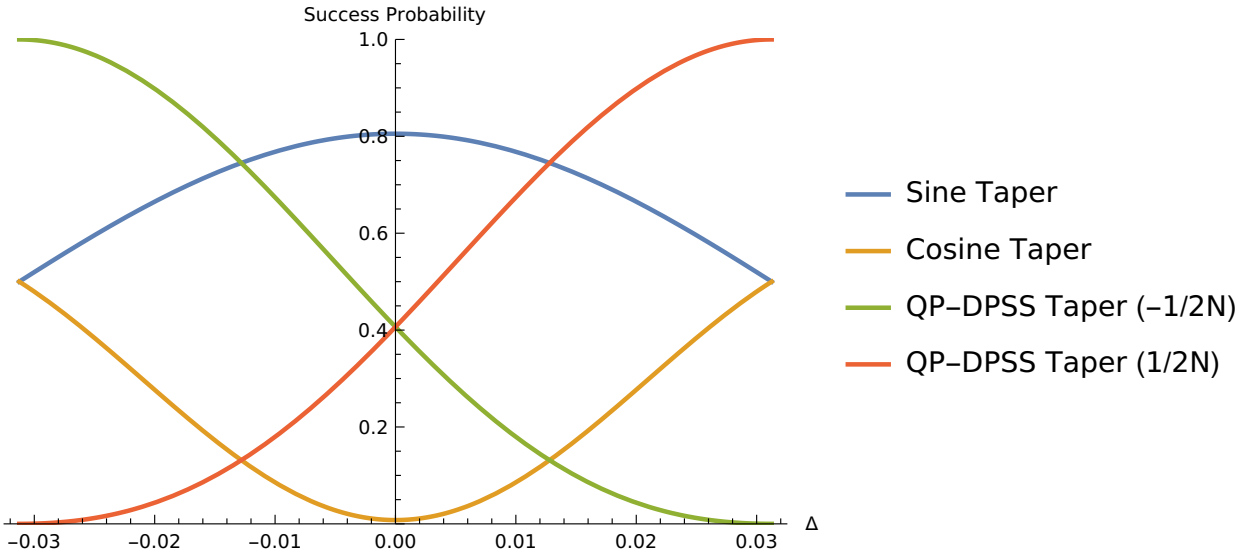


Figure 5.3: We plot the probability of the $\phi^{-1/2N}$ (green), $\phi^{1/2N}$ (red), $\phi^{\sin}$ (blue), and $\phi^{\cos}$ (orange) tapers to output the closest phase estimate as a function of Δ, for $N = 2^5$. Both cosine and sine tapers achieve 0.5 at $\Delta = \pm 1/2N$. This means with unit probability one of the two closest estimates will be returned.

in order to be useful. Therefore, in this subsection, we focus on finding the optimal taper that works best on average. The Lagrangian, $\mathcal{L}_{\text{avg}}$, for this modified optimization problem is similar to the one given by (5.5.4), except that the first term is now an expectation, resulting in the average success probability:

$$\mathcal{L}_{\text{avg}}\left(|\phi\rangle, \lambda\right) = \mathbb{E}_{\Delta\sim\mathcal{D}}\left[\sum_{j=-K}^{K}\left|\hat{\phi}\left(\Delta + \frac{j}{N}\right)\right|^2\right] - \lambda\left(\sum_{n=0}^{N-1}|\phi[n]|^2 - 1\right), \tag{5.5.11}$$

where $\mathcal{D}$ is the uniform distribution defined over the interval $\left[-\frac{1}{2N}, \frac{1}{2N}\right]$. Then solving for the above Lagrangian, we get the following eigenvalue equation (see App. 5.D for a detailed derivation):

$$\frac{1}{N}\sum_{n=0}^{N-1}\frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\pi(m-n)}\phi[n] = \lambda\phi[m]; \quad \forall m \in \{0, \ldots, N-1\}. \tag{5.5.12}$$

We observe that this eigenvalue equation appears commonly in the field of classical

signal processing and that DPSSs are eigenvectors of this equation. Now, with a similar reasoning as we used previously for the ideal case, the eigenvector $|\phi\rangle$ that maximizes the objective function, $\mathcal{L}_{\text{avg}}$ (i.e., average success probability), is the one with maximum eigenvalue. Thus, the optimal eigenvector is the DPSS with the maximum eigenvalue. Note that when we mention DPSS from here on, we refer to the DPSS with the maximum eigenvalue. For completeness, we provide a brief overview of DPSSs from the classical signal analysis point of view in App. 5.E.

The important issue of how to efficiently prepare the DPSS taper state is addressed in detail in Appendix 5.A. The DPSS taper has few appreciable components in the frequency domain, while the rest are extremely small. Thus, truncating the DPSS taper in the frequency domain introduces a very small error. The idea is then to prepare this truncated state using brute force quantum state synthesis approaches such as [SBM05] and then apply QFT to recover the approximate taper in the time domain.

Remark 64 *An important point to note is that the derivation of DPSSs from the classical signal processing point of view (provided in App. 5.E) relies heavily on the fact that the spectrum of frequencies is continuous. In contrast, we have a discrete frequency spectrum in the case of QPE. However, by taking the average of Δ over the uniform distribution, we effectively transform the problem to a continuous one. Thus, the average-case optimization problem has the same form as the classical continuous-frequency case, and as a result, the DPSS taper turns out to be the optimal taper.*

We now state an important theorem from the classical signal processing literature [KRD21] that provides (to date) the most stringent non-asymptotic bounds on the eigenvalues of the kernel in (5.5.12). The key idea is to use this theorem to derive a non-

asymptotic bound on the number of ancilla qubits, i.e., m, needed to achieve the average success probability of at least $1-\varepsilon$, for some $\varepsilon > 0$.

Theorem 65 ([KRD21, Corollary 1]) *For all $N, K \in \mathbb{N}$ such that $K \in \{0, \ldots, N/2-1\}$, the maximum eigenvalue $\lambda_{\max}$ of the kernel in (5.5.12) satisfies*

$$\lambda_{\max} \geq 1 - \min\left\{8\exp\left[-\frac{2K-1}{\frac{2}{\pi^2}\log(4N)}\right], 10\exp\left[-\frac{2K-6}{\frac{2}{\pi^2}\log\left(100K+75\right)}\right]\right\}. \tag{5.5.13}$$

As the maximum eigenvalue of the DPSS kernel in (5.5.12) is equal to the average success probability, the bound on this eigenvalue from the theorem given above naturally bounds the average success probability. Therefore, one way to infer the above theorem statement is as follows: when the maximal DPSS is used as a taper, it will output one of the $2K+1$ phase estimates closest to θ with an average success probability lower-bounded by the quantity on the right hand side of (5.5.13).

Applying Thm. 65 to the tQPE algorithm that uses the DPSS taper, we obtain the following result that holds for all $N, K \in \mathbb{N}$ which we term as the non-asymptotic regime.

Theorem 66 (Non-asymptotic) *To ensure that the output of the tQPE algorithm is one of the $2^m - 1$ phase estimates closest to the true phase θ with an average success probability of at least $1-\varepsilon$, we only require*

$$m = \left\lceil \log_2\left(\left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil + 1\right)\right\rceil + 1\,. \tag{5.5.14}$$

Proof. We use the bound from Thm. 65 to lower-bound the average success probability for outputting one of the $2K+1$ p-bit phase estimates closest to θ. Subsequently, if this bound is at least $1-\varepsilon$, we can say that the average success probability is also at least $1-\varepsilon$.

Formally, this is expressed as follows:

$$\varepsilon \geq \min\left\{8\exp\left[-\frac{2K-1}{\frac{2}{\pi^2}\log(4N)}\right], 10\exp\left[-\frac{2K-6}{\frac{2}{\pi^2}\log(100K+75)}\right]\right\}. \tag{5.5.15}$$

Without loss of generality, we choose to work with the second argument of the min function in the equation above, i.e.,

$$\varepsilon \geq 10\exp\left[-\frac{2K-6}{\frac{2}{\pi^2}\log(100K+75)}\right]. \tag{5.5.16}$$

Taking logarithms and flipping the sign of both sides, we find

$$\log(10/\varepsilon) \leq \frac{2K-6}{\frac{2}{\pi^2}\log(100K+75)}. \tag{5.5.17}$$

By assuming that $K \geq 1$, we first provide a lower bound for $\frac{2K-6}{2/\pi^2\log(100K+75)}$ in the following way:

$$\begin{aligned}
\frac{2K-6}{\frac{2}{\pi^2}\log(100K+75)} &\geq \frac{2K-6}{\frac{2}{\pi^2}\log(175K)} && (5.5.18)\\
&\geq \frac{K-3}{\log(175K)} && (5.5.19)\\
&= \frac{1}{175}\left(\frac{175K}{\log(175K)}\right) - \frac{3}{\log(175K)} && (5.5.20)\\
&\geq \frac{1}{175}\left(\frac{175K}{\log(175K+1)}\right) - 1 && (5.5.21)\\
&\geq \frac{\sqrt{175K}}{175} - 1 && (5.5.22)\\
&= \sqrt{\frac{K}{175}} - 1\,, && (5.5.23)
\end{aligned}$$

where the second-to-last inequality comes from the following logarithmic inequality:

$$\frac{x}{\log(x+1)} \geq \sqrt{x+1} \text{ for all } x \geq -1. \tag{5.5.24}$$

Now, by enforcing the following inequality,

$$\log(10/\varepsilon) \leq \sqrt{\frac{K}{175}} - 1, \tag{5.5.25}$$

we obtain $K \geq 175\left(\log\left(10/\varepsilon\right)+1\right)^2$. In other words, as long as K is at least $\left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil$, we are guaranteed that (5.5.16) holds and the error is bounded by ε. Recall from above that 2^m of the p-bit phase estimates are δ-close. Letting $K = 2^{m-1}-1$, we only need $m = \left\lceil \log_2\left(\left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil + 1\right)\right\rceil + 1$ additional qubits to ensure that our tQPE algorithm has at least $1-\varepsilon$ success probability. ■

Remark 67 *Although the analytical proof for Thm. 66 implies that K needs to be of order $\log^2(1/\varepsilon)$ to achieve an average success probability of $1-\varepsilon$, for all practical purposes K only needs to be of order $\log(1/\varepsilon)$. For example, for all $\varepsilon \geq 10^{-81}$, we have that $K \leq 192$, and as a result, we can replace K inside the logarithm in (5.5.17) with 192 and find that $K = \lceil \log(10/\varepsilon) \rceil + 3$ is sufficient.*

Since Thm. 65 holds for all $N \in \mathbb{N}$, our result described in Thm. 66 also holds for all N including the asymptotic regime where $N \to \infty$ and $\delta \to 0$. However, the bound on the maximum eigenvalue shown in Thm. 65 can be made tighter when we restrict our analysis to the asymptotic case with large N and small δ. In what follows, we state the result for this asymptotic case.

Theorem 68 (Asymptotic) *In the regime where N is large and δ is small, the tQPE algorithm with the DPSS taper outputs one of the 2^m-1 phase estimates closest to the true phase θ with an average success probability of at least $1-\varepsilon$ when*

$$m = \left\lceil \log_2 \log\left(\frac{1}{\varepsilon}\right)\right\rceil \tag{5.5.26}$$

additional qubits are used in the tQPE algorithm.

Proof. Let $W = 2\delta = \frac{1}{2^l}$. Then, $NW = 2^m$ because we know that $N = 2^{l+m}$. In the original paper on DPSS [Sle78], Slepian provided an asymptotic expression for the eigenvalues

$\{\lambda_k\}_k$ of the DPSSs (i.e., when N is large or $\delta, W \to 0$),

$$1 - \lambda_k = \pi^{1/2}(k!)^{-1}2^{(14k+9)/4}\alpha^{(2k+1)/4}(2-\alpha)^{-(k+1/2)}N^{k+1/2}e^{-\gamma N}, \tag{5.5.27}$$

where α and γ are defined as follows:

$$\alpha = 1 - \cos 2\pi W \tag{5.5.28}$$

$$\gamma = \log\left(1 + \frac{2\sqrt{\alpha}}{\sqrt{2} - \sqrt{\alpha}}\right). \tag{5.5.29}$$

From the definition of α, it is clear that $\alpha \to 0$ as $W \to 0$. To this end, we Taylor expand the cosine function, to get the following expression for α:

$$\alpha = \frac{(2\pi W)^2}{2} + O((2\pi W)^4). \tag{5.5.30}$$

Substituting this in the definition of γ, we get

$$\gamma = \log\left(1 + \frac{2\sqrt{\frac{(2\pi W)^2}{2} + O((2\pi W)^4)}}{\sqrt{2} - \sqrt{\frac{(2\pi W)^2}{2} + O((2\pi W)^4)}}\right) \tag{5.5.31}$$

$$\sim \log(1 + 2\pi W) \tag{5.5.32}$$

$$\sim 2\pi W. \tag{5.5.33}$$

Also, $\alpha \sim 2\pi^2 W^2$. Plugging the values of α, γ, and also $k = 0$ (because we are interested in the maximum eigenvalue) into the first equation, we get

$$1 - \lambda_0 \sim \pi^{1/2}2^{9/4}(2\pi^2W^2)^{1/4}\underbrace{(2 - 2\pi^2W^2)^{-1/2}}_{\sim 2^{-1/2}}N^{1/2}e^{-2\pi WN} \tag{5.5.34}$$

$$= \pi^{1/2}2^{9/4}(2\pi^2W^2)^{1/4}2^{-1/2}N^{1/2}e^{-2\pi WN} \tag{5.5.35}$$

$$= 4\pi(NW)^{1/2}e^{-2\pi NW}. \tag{5.5.36}$$

We require that $1 - \lambda_0 \leq \varepsilon$. This implies

$$4\pi(NW)^{1/2}e^{-2\pi NW} \leq \varepsilon \tag{5.5.37}$$

Using the fact that $4\pi x^{1/2} e^{-2\pi x} \leq e^{-x}$ for $x \geq 1$, it suffices to choose

$$m = \lceil \log_2 \log(1/\varepsilon) \rceil \tag{5.5.38}$$

Please note that this result can be further tightened, but it is sufficient for our purpose. ■

Remark 69 *Now, using the fact that $K = 2^{m-1} - 1$, we further obtain:*

$$K = 2^{\lceil \log_2 \log(1/\varepsilon) \rceil - 1} - 1 \leq \log(1/\varepsilon) - 1 \ . \tag{5.5.39}$$

While we do not gain an improvement in m with respect to ε in the asymptotic regime, we observe an improvement in the scaling of K with respect to ε in this regime. In particular, we showed how the upper bound on K decreases from $175(\log(10/\varepsilon) + 1)^2 + 1$ to $\log(1/\varepsilon) - 1$ when we utilize the asymptotic bound that Slepian provides in [Sle78] which holds in the regime where N is large and δ is small. Recall that the tQPE algorithm succeeds in outputting one of the $2K+1$ phase estimates closest to the true phase θ with an average success probability of at least $1 - \varepsilon$. Thus, a smaller K value would imply that the DPSS taper has greater frequency concentration around θ and endows a smaller set of closest phase estimates with the same success probability of $1 - \varepsilon$.

In the regime where ε is very small and δ is finite, the bound from Thm. 65 can be sharpened significantly. This leads directly to the main result of our work. Importantly, the following theorem also applies in the non-asymptotic setting: by choosing a smaller target error probability $\varepsilon' \ll \varepsilon$ and the corresponding number of additional qubits m, one can ensure that the algorithm achieves success probability at least $1 - \varepsilon$.

Theorem 70 (Main result) *The tQPE algorithm with the DPSS taper outputs one of the $2^m - 1$ phase estimates closest to the true phase θ with an average success probability of at least $1 - \varepsilon$ when*

$$m = \log_2 \log \frac{1}{\varepsilon} - \log_2 2\pi + O(1) \tag{5.5.40}$$

additional qubits are used in the tQPE algorithm.

Proof. This argument applies when W—and hence ℓ—is fixed. We focus on the asymptotic regime $\varepsilon \to 0$. Our interest is in the largest eigenvalue λ_0. Using the asymptotic result from [Sle78], for fixed W,

$$1 - \lambda_0 \;\sim\; C(W)\, N^{1/2} e^{-\gamma(W)N}, \tag{5.5.41}$$

where

$$C(W) = \pi^{1/2} 2^{9/4}\, \alpha(W)^{1/4}\, (2 - \alpha(W))^{-1/2}. \tag{5.5.42}$$

Here $\alpha(W)$ and $\gamma(W)$ are given by

$$\alpha = 1 - \cos 2\pi W\,, \tag{5.5.43}$$

$$\gamma = \log\left(1 + \frac{2\sqrt{\alpha}}{\sqrt{2} - \sqrt{\alpha}}\right). \tag{5.5.44}$$

By the definition of $\sim$, for every $\delta \in (0, 1)$ there exists N_0 such that for all $N \geq N_0$, we have

$$1 - \lambda_0 \leq (1 + \delta) C(W)\, N^{1/2} e^{-\gamma(W)N}. \tag{5.5.45}$$

Now since $W = 2^{-l}$, $N = 2^{l+m}$, we have $NW = 2^m =: x$ and $N = x/W$. Plugging $N = x/W$ into (5.5.45), we get

$$1 - \lambda_0 \leq A(W)\, x^{1/2} e^{-\tau(W)x}, \tag{5.5.46}$$

where, again, for brevity, we introduce the following two quantities:

$$A(W) := \frac{(1 + \delta) C(W)}{\sqrt{W}}, \quad \tau(W) := \frac{\gamma(W)}{W}. \tag{5.5.47}$$

We want that $1 - \lambda_0 \leq \varepsilon$. This implies that we want to find such an x such that the following holds:

$$A(W)\, x^{1/2} e^{-\tau(W)x} \leq \varepsilon. \tag{5.5.48}$$

Let us pick the following x:

$$x = \frac{1}{\tau(W)}\left(\log(A(W)/\varepsilon) + \frac{1}{2}\log\log(A(W)/\varepsilon)\right). \tag{5.5.49}$$

Now let us check if this value of x satisfies the condition (5.5.48). We begin by plugging in this value in (5.5.48):

$$A(W)\, x^{1/2} e^{-\tau(W)x} = \varepsilon \sqrt{\frac{x}{\log(A(W)/\varepsilon)}}. \tag{5.5.50}$$

For sufficiently small ε, we have

$$\sqrt{\frac{x}{\log(A(W)/\varepsilon)}} \leq \sqrt{\frac{2}{\tau(W)}}. \tag{5.5.51}$$

This implies

$$A(W)\, x^{1/2} e^{-\tau(W)x} \leq \varepsilon \sqrt{\frac{2}{\tau(W)}}. \tag{5.5.52}$$

Furthermore, since for all $W \in (0, 1/2]$, we have $\tau(W) \geq 2\pi$, we can say that

$$A(W)\, x^{1/2} e^{-\tau(W)x} \leq \varepsilon. \tag{5.5.53}$$

This completes the check that the value of x that we picked satisfies the condition (5.5.48).

Since $x = 2^m$, any integer

$$m \geq \left\lceil \log_2\left(\frac{1}{\tau(W)}\left[\log\left(\frac{A(W)}{\varepsilon}\right) + \frac{1}{2}\operatorname{loglog}\left(\frac{A(W)}{\varepsilon}\right)\right]\right)\right\rceil \tag{5.5.54}$$

guarantees $1 - \lambda_0 \leq \varepsilon$. Now for all $W \in (0, 1/4]$, $A(W)$ is bounded from above as

$$A(W) \leq \frac{4\pi(1+\delta)}{\cos(\pi W)}. \tag{5.5.55}$$

With this, we finally get a fully explicit sufficient bound in terms of ε and W:

$$m \geq \left\lceil \log_2\left(\frac{1}{\tau(W)}\left[\log\left(\frac{4\pi(1+\delta)}{\varepsilon\cos(\pi W)}\right) + \frac{1}{2}\operatorname{loglog}\left(\frac{4\pi(1+\delta)}{\varepsilon\cos(\pi W)}\right)\right]\right)\right\rceil. \tag{5.5.56}$$

Since $\tau(W)$ is a positive constant for fixed W, asymptotically as $\varepsilon \to 0$, we finally have

$$m = \log_2 \log \frac{1}{\varepsilon} - \log_2 \tau(W) + O(1) \tag{5.5.57}$$

so $m = \Theta(\log\log(1/\varepsilon))$ in the finite-W regime. ■

5.5.4 Analysis of Worst-Case Error

Ideally, we would like to find a taper that has the smallest error probability for its worst-case Δ^*. For most tapers used in classical signal processing (including the tophat and DPSS tapers), $\Delta^* = \pm 1/(2N)$ corresponds to the largest error probability. However, in general, different tapers have different worst-case Δ^*, which makes the corresponding optimization problem too hard. In this section, we analyze the performance of the DPSS taper in its worst case, which, as mentioned above, is at $\Delta^* = \pm 1/2N$. We find that, although not optimized for the worst-case scenario, the DPSS taper performs well and is optimal asymptotically.

To perform such a comparison, we first restate the kernel of the DPSS taper which coincides with the eigenvalue equation shown in (5.5.12):

$$\frac{1}{N}\sum_{n=0}^{N-1} \frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\pi(m-n)}\phi[n] = \lambda\phi[m]; \quad \forall m \in \{0,\dots,N-1\}. \tag{5.5.58}$$

Recall that the DPSS taper is the eigensequence with the largest eigenvalue that satisfies the DPSS kernel. When considering the success probability of the DPSS taper in the worst case, where $\Delta = \pm 1/2N$, we let $|\phi\rangle$ denote the DPSS taper and adapt the calculations in App. 5.D to include an additional complex rotation $e^{2\pi i(n'-n)\Delta}$ that translates the phase estimates in the frequency domain by $\Delta = \pm 1/2N$ to obtain the following expression:

$$\sum_{j=-K}^{K}\left|\hat{\phi}\left(\Delta+\frac{j}{N}\right)\right|^2 = \frac{1}{N}\sum_{n,m=0}^{N-1} e^{2\pi i(m-n)\Delta}\frac{\sin\left(\pi(m-n)\left(\frac{2K+1}{N}\right)\right)}{\sin\left(\pi(m-n)/N\right)}\phi^*[n]\phi[m] \tag{5.5.59}$$

$$= \frac{1}{N}\sum_{n,m=0}^{N-1} \cos\left(\pi(m-n)/N\right)\frac{\sin\left(\pi(m-n)\left(\frac{2K+1}{N}\right)\right)}{\sin\left(\pi(m-n)/N\right)}\phi^*[n]\phi[m] \tag{5.5.60}$$

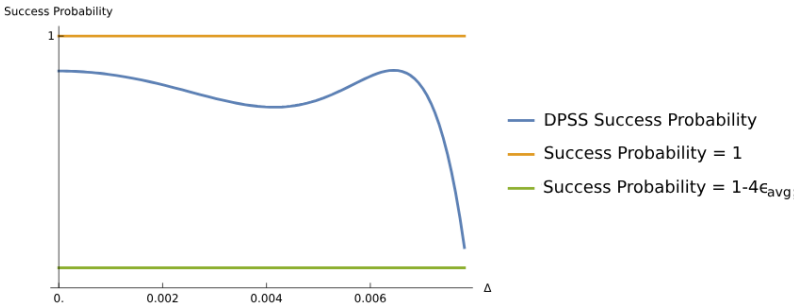
where the last equality comes from substituting $\Delta = \frac{1}{2N}$ and observing how the complex terms from the complex exponential annihilate each other when we run over all summands indexed by m and n. In other words, the success probability for the DPSS taper in

the worst-case would then be the quantity shown in (5.5.60).

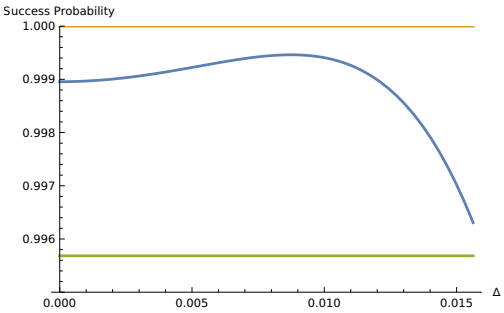
From Thm. 65, we make the observation that the maximum eigenvalue, or the success probability, of the average-case optimal DPSS taper tends to 1 as $N \to \infty$. This is well-aligned with our intuition because we would expect our DPSS taper to have greater spectral concentration in the central lobe in the frequency domain. At the same time, we note that the worst-case success probability of the DPSS taper also converges to 1 when $N \to \infty$. This is a result of the complex exponential term in (5.5.59) approaching 1 as $N \to \infty$, resulting in the worst-case DPSS success probability expression approaching the average-case success probability in the same limit. Therefore, DPSS is optimal asymptotically. While the DPSS taper may not achieve the same $1 - \varepsilon$ success probability as it would in the average case for the non-asymptotic case, i.e., when N is finite, the fact that the worst-case DPSS success probability converges to the average-case DPSS success probability for large values of N implies that the worst-case success probability of the DPSS taper would share a similar dependence on ε. We provide numerical evidence in Figs. 5.4(a), 5.4(b), and 5.4(c) that shows that the worst-case success probability of the DPSS taper never falls below $1 - 4\varepsilon$ for several different parameters.

5.5.5 Numerics

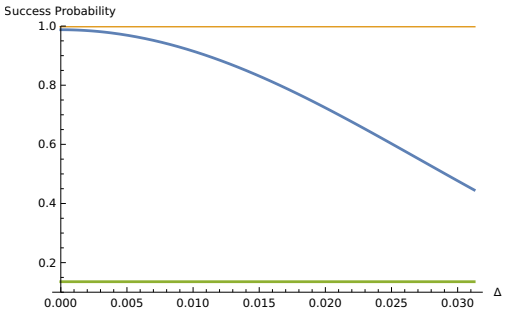
In this section, we provide numerics for three of the tapers we have discussed so far: the DPSS taper, the sine taper, and the tophat taper. In Figs. 5.5(a), 5.5(b), 5.5(c), and 5.5(d), we plot the success probability of each taper as a function of Δ, i.e., the distance between the true phase and the grid point closest to it, for various values of m. The DPSS taper is optimal for the average case. The tophat taper is designed to output the true phase with unit probability when the true phase coincides with a grid point and the sine taper has



(a) $m = 3, K = 3$



(b) $m = 2, K = 1$



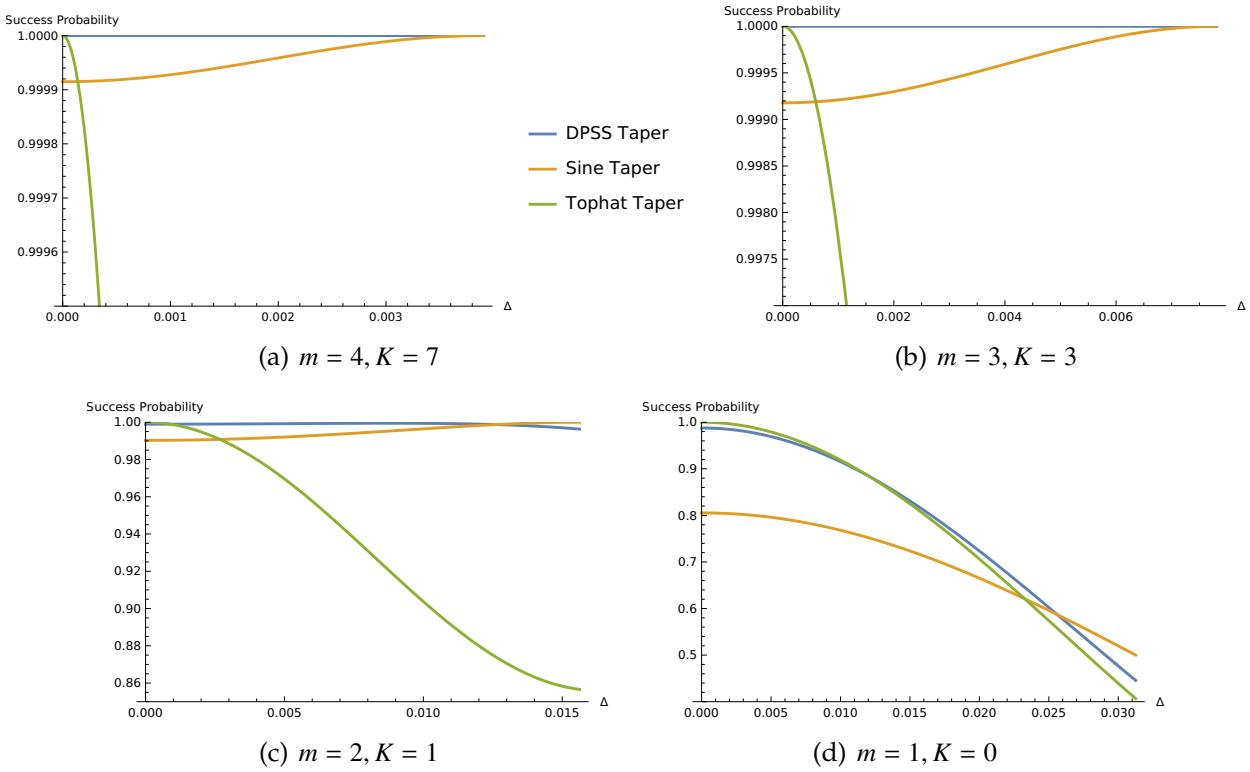
(c) $m = 1, K = 0$

Figure 5.4: Performance of the DPSS taper as a function of Δ from 0 to $1/2N$. (a) The figure is plotted for success probabilities ranging from 0.999999975 to 1. We have omitted labeling the tick with 0.999999975 to avoid cluttering the figure.

the property that it outputs one of the two nearest estimates with unit probability when the true phase is exactly between two grid points. Both of these properties are confirmed by the plots. Although the DPSS kernel does not succeed with unit probability for any Δ it does perform well for the entire range, which is consistent with the fact that it is optimal for the average case.

In Figs. 5.4(a), 5.4(b), and 5.4(c), we again plot the success probability of the DPSS taper as a function of Δ for various values of m. In contrast to Figs. 5.5(a), 5.5(b), 5.5(c), and 5.5(d), we do not plot the other tapers; instead we display a horizontal line at $1 - 4\varepsilon_{\text{ave}}$, where ε_{ave} is the average error associated with the DPSS taper

(alternatively, $1-\varepsilon_{\text{ave}}$ is the largest eigenvalue of the DPSS kernel). We see that the success probability is above $1-4\varepsilon_{\text{ave}}$ for all Δ. Thus, our numerics suggest that the DPSS taper has a worst-case error that is at most four times its average-case error. Since the worst-case error cannot be smaller than the average-case, the worst-case error of DPSS is at most a factor of four worse than the best possible worst-case error. This means that although we have not been able to find the optimal taper for the worst case, the DPSS taper cannot be too far from it.



(a) $m = 4, K = 7$

(b) $m = 3, K = 3$

(c) $m = 2, K = 1$

(d) $m = 1, K = 0$

Figure 5.5: Comparison between the DPSS taper, the sine taper, and the tophat taper.

5.6 Discussion

Quantum algorithms that require coherent phase estimation, such as HHL algorithm [HHL09] and Quantum Metropolis Sampling [TOV+11], also require that the estimate of the phase be uncomputed at some point. If this were not the case, incoherent phase estimation would be sufficient as discarding the ancilla register after coherent QPE renders it equivalent to incoherent QPE. Because tQPE outputs multiple estimates for a particular phase (unlike median boosted QPE that outputs the closest two estimates with high probability), one might worry that the uncomputation step might fail. In Appendix 5.G we show that this concern is unfounded: the error of any coherent QPE algorithm due to uncomputation is of the same magnitude as the error due to the approximation of the phase. For a precise statement of this result, see Theorem 75.

We note that the authors of Ref. [RDD17] studied continuous QPE and formulated a different optimization problem than ours. Interestingly, their optimal taper for continuous QPE also resulted in DPSS. So, it would be valuable to explore the similarities and differences between these two optimization problems, which we leave for future work.

We remark that the asymptotic scaling of K for the DPSS taper compares favorably against the Kaiser taper, which is known to also achieve a similar asymptotic scaling of m as our DPSS taper [BSG+22]. There are other additional factors inside the logarithm for the Kaiser taper that the DPSS taper does not have. Furthermore, for the case of the Kaiser taper, its K value scales as $1/2\pi \log(1/\varepsilon) + O(\log\log(1/\varepsilon))$ with respect to ε in the asymptotic regime as shown in Appendix D of Ref. [BSG+22]. As shown in the proof for Thm. 68, the DPSS taper scales as $\lceil \log(1/\varepsilon) \rceil$ in the asymptotic regime, which implies that the DPSS taper has better frequency concentration around θ than the Kaiser taper in the regime where N is large and δ is small. This corresponds well with existing classical signal processing

literature, where the DPSS taper is known to have maximal frequency concentration in the central lobe out of all taper functions, and the Kaiser taper has originally been designed as an approximation to the optimal DPSS taper.

In addition to the asymptotic analysis, we also provide the explicit expressions for m in the non-asymptotic regime. This will be important for the practical implementation of the QPE algorithm in the early fault-tolerant regime. Finally, we would like to mention that we approach the QPE problem from a different angle, i.e., by framing optimization problems. Due to the optimality of the DPSS taper, all tapers must use $\Omega(\log\log(1/\varepsilon))$ additional qubits to boost the average success probability to at least $1-\varepsilon$. This means that there cannot be any further improvements when using tapers for QPE. This result complements Thm. 1.3 of [MdW23], which provides the same lower bound on the ε-dependence for any algorithm solving QPE in the worst case. However, while Thm. 1.3 of [MdW23] is general and holds for any QPE algorithm, our lower bound only applies to tQPE with arbitrary tapers.

Due to the flexibility of our formulation, it allows us to find tapers satisfying different optimization criteria. For instance, we find that a sinusoidal taper minimizes the average case error under the additional constraint that for $m = 0$ the error probability is zero whenever the true phase happens to be exactly in between two estimates. This taper has been obtained before in the literature as a result of a different optimization problem [LP96, BDM99]: it is the taper that minimizes a cost function that penalizes estimates that are further away from the true value as $4\sin^2((\theta-\theta_{\text{est}})/2)$. We have arrived at this taper from a completely different perspective.

It might be possible to construct tapers with slightly worse average case error but better worst case error by hybridizing some of the eigenvectors of the DPSS kernel with

large eigenvalues. This is similar to how we constructed the sine taper by hybridizing two QP-DPSS tapers. Our numerics suggests that the worst case error of DPSS taper is at most $4\varepsilon_{\text{ave}}$. It is an open question how close the worst case error can be made by hybridizing DPSS's.

5.7 Acknowledgements

The authors thank Sam Slezak for useful discussions related to the error analysis in App. 5.G and anonymous reviewers for their suggestions that helped improve the presentation of this work. DP and SJST were participants in the 2023 Quantum Computing Summer School at LANL, sponsored by the LANL Information Science & Technology Institute. DP spent the following fall semester as a GRA at LANL during which he was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory (LANL) under project number 20230049DR. This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Science Center (QSC). YS was funded by the QSC to perform the analytical and numerical analysis and to write the manuscript along with the other authors. ATS acknowledges initial support from the LANL ASC Beyond Moore's Law project and subsequent support from the Laboratory Directed Research and Development program of Los Alamos National Laboratory (LANL) under project number 20230049DR.

BIBLIOGRAPHY

[AKN98] Dorit Aharonov, Alexei Kitaev, and Noam Nisan. Quantum circuits with mixed states. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 20–30, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276708.

[ALL+21] Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang. Quantum-accelerated multilevel monte carlo methods for stochastic differential equations in mathematical finance. *Quantum*, 5:481, 2021. doi:10.22331/q-2021-06-24-481.

[BBG+13] Robert Beals, Stephen Brierley, Oliver Gray, Aram W Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, and Mark Stather. Efficient distributed quantum computing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2153):20120686, 2013. doi:10.1098/rspa.2012.0686.

[BDM99] Vladimir Bužek, Radoslav Derka, and Serge Massar. Optimal quantum clocks. *Physical Review Letters*, 82(10):2207–2210, 1999. doi:10.1103/PhysRevLett.82.2207.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. doi:10.1090/conm/305/05215.

[BSG+22] Dominic W Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush. Quantifying quantum advantage in topological data analysis. 2022. arXiv:2209.13581.

[BW00] Dominic W Berry and Howard M Wiseman. Optimal states and almost optimal adaptive measurements for quantum interferometry. *Physical Review Letters*, 85(24):5098–5101, 2000. doi:10.1103/PhysRevLett.85.5098.

[CBB20] François Chapeau-Blondeau and Etienne Belin. Fourier-transform quantum phase estimation with quantum phase noise. *Signal Processing*, 170:107441, May 2020. doi:10.1016/j.sigpro.2019.107441.

[CEMM98] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series*

A: Mathematical, Physical and Engineering Sciences, 454(1969):339–354, 1998. doi:10.1098/rspa.1998.0164.

[CF+48] George Ashley Campbell, Ronald M Foster, et al. *Fourier Integrals for Practical Applications*. D. van Nostrand, NY, 1948.

[CKBG23] Chi-Fang Chen, MJ Kastoryano, FGSL Brandao, and A Gilyén. Quantum thermal state preparation. 2023. arXiv:2303.18224.

[CKG23] Chi-Fang Chen, Michael J Kastoryano, and András Gilyén. An efficient and exact noncommutative quantum gibbs sampler. 2023. arXiv:2311.09207.

[Cop94] D Coppersmith. An approximate Fourier transform useful in quantum factoring. *IBM Research Report*, pages RC–19642, 1994. arXiv:quant-ph/0201067.

[CSS18] Anirban Narayan Chowdhury, Yiğit Subaşı, and Rolando D Somma. Improved implementation of reflection operators. 2018. arXiv:1803.02466.

[DL23] Zhiyan Ding and Lin Lin. Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *PRX Quantum*, 4(2):020331, May 2023. doi:10.1103/PRXQuantum.4.020331.

[ESP21] Alexander Engel, Graeme Smith, and Scott E Parker. Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms. *Physics of Plasmas*, 28(6):062305, 2021. doi:10.1063/5.0040313.

[FGML17] François Fillion-Gourdeau, Steve MacLean, and Raymond Laflamme. Efficient state initialization by a quantum spectral filtering algorithm. *Physical Review A*, 95(4):042331, 2017. doi:10.1103/PhysRevA.95.042331.

[GPS24] Sean Greenaway, William Pol, and Sukin Sim. A case study against QSVT: Assessment of quantum phase estimation improved by signal processing techniques. 2024. arXiv:2404.01396.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. doi:10.1145/3313276.3316366.

[GSP21] Valentin Gebhart, Augusto Smerzi, and Luca Pezzè. Bayesian quantum multiphase estimation algorithm. *Physical Review Applied*, 16(1):014035, 2021. doi:10.1103/PhysRevApplied.16.014035.

[HH00] L. Hales and S. Hallgren. An improved quantum fourier transform algorithm and applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, FOCS 2000. IEEE Comput. Soc, November 2000. doi:10.1109/SFCS.2000.892139.

[HHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. doi:10.1103/PhysRevLett.103.150502.

[HNS02] Peter Hoyer, Jan Neerbek, and Yaoyun Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002. doi:10.1007/s00453-002-0976-3.

[HTR09] Simon Haykin, David J Thomson, and Jeffrey H Reed. Spectrum sensing for cognitive radio. *Proceedings of the IEEE*, 97(5):849–877, 2009. doi:10.1109/JPROC.2009.2015711.

[Kai66] James F Kaiser. Digital filters. In *System Analysis by Digital Computer*, pages 218–285. Wiley New York, NY, 1966.

[Kit95] A Yu Kitaev. Quantum measurements and the Abelian stabilizer problem. 1995. arXiv:quant-ph/9511026.

[Kla03] Hartmut Klauck. Quantum time-space tradeoffs for sorting. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 69–76, 2003. doi:10.1145/780542.780553.

[KRD21] Santhosh Karnik, Justin Romberg, and Mark A Davenport. Improved bounds for the eigenvalues of prolate spheroidal wave functions and discrete prolate spheroidal sequences. *Applied and Computational Harmonic Analysis*, 55:97–128, 2021. doi:10.1016/j.acha.2021.04.002.

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. 2013. arXiv:1307.0411.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal

component analysis. *Nature Physics*, 10(9):631–633, 2014. doi:10.1038/nphys3029.

[LP61] Henry J Landau and Henry O Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty—II. *Bell System Technical Journal*, 40(1):65–84, 1961. doi:10.1002/j.1538-7305.1961.tb03977.x.

[LP62] Henry J Landau and Henry O Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty—III: The dimension of the space of essentially time- and band-limited signals. *Bell System Technical Journal*, 41(4):1295–1336, 1962. doi:10.1002/j.1538-7305.1962.tb03279.x.

[LP96] A Luis and J Peřina. Optimum phase-shift estimation and the quantum description of the phase difference. *Physical Review A*, 54(5):4564–4570, 1996. doi:10.1103/PhysRevA.54.4564.

[LT22] Lin Lin and Yu Tong. Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum*, 3(1):010318, February 2022. doi:10.1103/PRXQuantum.3.010318.

[MdW23] Nikhil S Mande and Ronald de Wolf. Tight bounds for quantum phase estimation and related problems. 2023. arXiv:2305.04908.

[MGB22] Sam McArdle, András Gilyén, and Mario Berta. Quantum state preparation without coherent arithmetic. *arXiv preprint arXiv:2210.14892*, 2022. arXiv:2210.14892.

[MP99] Partha P Mitra and Bijan Pesaran. Analysis of dynamic brain imaging data. *Biophysical Journal*, 76(2):691–708, 1999. doi:10.1016/S0006-3495(99)77236-X.

[MRTC21] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021. doi:10.1103/PRXQuantum.2.040203.

[NLY23] Hongkang Ni, Haoya Li, and Lexing Ying. On low-depth algorithms for quantum phase estimation. *Quantum*, 7:1165, November 2023. doi:10.22331/q-2023-11-06-1165.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA.

Quantum Information & Computation, 9(11):1053–1068, November 2009. doi:10.26421/qic9.11-12-8.

[OTT19] Thomas E O'Brien, Brian Tarasinski, and Barbara M Terhal. Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments. *New Journal of Physics*, 21(2):023022, February 2019. doi:10.1088/1367-2630/aafb8e.

[PLVI87] Jeffrey Park, Craig R Lindberg, and Frank L Vernon III. Multitaper spectral analysis of high-frequency seismograms. *Journal of Geophysical Research: Solid Earth*, 92(B12):12675–12684, 1987. doi:10.1029/JB092iB12p12675.

[PTSS24] Dhrumil Patel, Shi Jie Samuel Tan, Yigit Subasi, and Andrew T. Sornborger. Optimal coherent quantum phase estimation via tapering, 2024. arXiv:2403.18927.

[PW93] Donald B Percival and Andrew T Walden. *Spectral Analysis for Physical Applications*. Cambridge University Press, 1993.

[Ral20] Patrick Rall. Quantum algorithms for estimating physical quantities using block encodings. *Physical Review A*, 102(2):022408, 2020. doi:10.1103/PhysRevA.102.022408.

[Ral21] Patrick Rall. Faster coherent quantum algorithms for phase, energy, and amplitude estimation. *Quantum*, 5:566, October 2021. doi:10.22331/q-2021-10-19-566.

[RDD17] Wojciech Rządkowski and Rafał Demkowicz-Dobrzański. Discrete-to-continuous transition in quantum phase estimation. *Physical Review A*, 96(3):032319, 2017. doi:10.1103/PhysRevA.96.032319.

[RIK22] Gumaro Rendon, Taku Izubuchi, and Yuta Kikuchi. Effects of cosine tapering window on quantum phase estimation. *Physical Review D*, 106(3):034503, 2022. doi:10.1103/PhysRevD.106.034503.

[Rog20] Alessandro Roggero. Spectral-density estimation with the Gaussian integral transform. *Physical Review A*, 102(2):022409, 2020. doi:10.1103/PhysRevA.102.022409.

[SBM05] Vivek V Shende, Stephen S Bullock, and Igor L Markov. Synthesis of quantum

logic circuits. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 272–275, 2005. doi:10.1145/1120725.1120847.

[SHF13] Krysta M Svore, Matthew B Hastings, and Michael Freedman. Faster phase estimation. 2013. arXiv:1304.0741.

[Sho94] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, 1994. IEEE. doi:10.1109/SFCS.1994.365700.

[Sle64] David Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainty—IV: Extensions to many dimensions; generalized prolate spheroidal functions. *Bell System Technical Journal*, 43(6):3009–3057, 1964. doi:10.1002/j.1538-7305.1964.tb01037.x.

[Sle78] David Slepian. Prolate spheroidal wave functions, Fourier analysis, and uncertainty—V: The discrete case. *Bell System Technical Journal*, 57(5):1371–1430, 1978. doi:10.1002/j.1538-7305.1978.tb02104.x.

[Som19] Rolando D Somma. Quantum eigenvalue estimation via time series analysis. *New Journal of Physics*, 21(12):123025, 2019. doi:10.1088/1367-2630/ab5c60.

[SP61] David Slepian and Henry O Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty—I. *Bell System Technical Journal*, 40(1):43–63, 1961. doi:10.1002/j.1538-7305.1961.tb03976.x.

[SY12] Andrew Sornborger and Takeshi Yokoo. A multivariate, multitaper approach to detecting and estimating harmonic response in cortical optical imaging data. *Journal of Neuroscience Methods*, 203(1):254–263, 2012. doi:10.1016/j.jneumeth.2011.09.018.

[Tho82] David J Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982. doi:10.1109/PROC.1982.12433.

[TOV+11] Kristan Temme, Tobias J Osborne, Karl G Vollbrecht, David Poulin, and Frank Verstraete. Quantum Metropolis sampling. *Nature*, 471(7336):87–90, 2011. doi:10.1038/nature09770.

[WBC22] Kianna Wan, Mario Berta, and Earl T. Campbell. Randomized quantum algorithm for statistical phase estimation. *Physical Review Letters*, 129(3):030503, July 2022. doi:10.1103/PhysRevLett.129.030503.

[WBD+21] Lewis Wright, Fergus Barratt, James Dborin, George H Booth, and Andrew G Green. Automatic post-selection by ancillae thermalization. *Physical Review Research*, 3(3):033151, 2021. doi:10.1103/PhysRevResearch.3.033151.

[WBL12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109(5):050505, 2012. doi:10.1103/PhysRevLett.109.050505.

[XC84] Wen Yuan Xu and Christodoulos Chamzas. On the periodic discrete prolate spheroidal sequences. *SIAM Journal on Applied Mathematics*, 44(6):1210–1217, 1984. doi:10.1137/0144086.

[ZKD+17] Zhihui Zhu, Santhosh Karnik, Mark A Davenport, Justin Romberg, and Michael B Wakin. The eigenvalue distribution of discrete periodic time-frequency limiting operators. *IEEE Signal Processing Letters*, 25(1):95–99, 2017. doi:10.1109/lsp.2017.2751578.

APPENDIX

5.A Preparation of the DPSS Taper

In this section, we focus on preparing the DPSS taper $|\phi\rangle$. We use a modification of the method for preparing broadband Gaussian states presented in [CSS18] to prepare the taper state. The key idea is that the DPSS taper in the frequency domain, i.e., $|\hat{\phi}\rangle = U_{\text{QFT}}|\phi\rangle$, is narrow-band. By narrow-band, we mean that it is highly concentrated on just a few grid points, implying that the state, $|\hat{\phi}\rangle$, has appreciable amplitudes on very few of its basis states. Therefore, we can prepare an approximate version of $|\hat{\phi}\rangle$ (say $|\hat{\phi}^*\rangle$) using standard methods presented in [CSS18] with very little overhead. After that, we can perform an inverse quantum Fourier transform on $|\hat{\phi}^*\rangle$ to obtain $|\phi^*\rangle$, which closely approximates the DPSS taper $|\phi\rangle$.

Before we provide an explicit description of the taper state preparation protocol, we show in the lemma below that we can prepare an approximate DPSS taper state that is ε-close to the DPSS taper state using $N' = O\left(\log^2(1/\varepsilon)\right) \ll N$ parameters where ε is an average failure probability of the tQPE algorithm using the DPSS taper. In this case, we refer to the non-trivial amplitudes of the state $|\hat{\phi}\rangle$ as the N' parameters that are needed to characterize our approximate DPSS taper state.

Lemma 71 *The number of parameters, N', necessary to prepare a state, $|\phi^*\rangle$, that is $O(\sqrt{\varepsilon})$-close in Euclidean distance to $|\phi\rangle$ i.e., $\||\phi^*\rangle - |\phi\rangle\|_2 \leq 2\sqrt{\varepsilon}$ satisfies $N' = 2\left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil + 1$.*

Proof. Recall from Thm. 65 and Thm. 66 that

$$\varepsilon \leq 10\exp\left[-\frac{2K-6}{\frac{2}{\pi^2}\log(100K+25)}\right], \tag{5.A.1}$$

where ε is the average probability outside of the central lobe of the most-concentrated DPSS taper in the frequency domain. Because the DPSS taper in the frequency domain has $2K+1$ grid points in the central lobe, the number of parameters N' needed to describe the frequency amplitudes in the central lobe is then $2K+1$. Note that N' may be determined with a standard classical numerical computation on a lattice of size N with central lobe bandwidth parameter K (see section 8.3 of [PW93] for the classical computation of DPSSs on a lattice, the N' parameters are then the frequency coefficients within the bandwidth $2K+1$ of the discrete Fourier transform of this function). From the derivation of the proof of Thm. 66, we observe that $K = \left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil$. Since $N' = 2K+1$, we obtain $N' = 2\left\lceil 175\left(\log\left(10/\varepsilon\right)+1\right)^2\right\rceil + 1$.

Let $P_{N'}$ be the projector into the subspace spanned by the computational basis states corresponding to the central lobe of the DPSS taper in the frequency domain. We define a truncation of the DPSS taper as

$$|\hat{\phi}^*\rangle = \frac{P|\hat{\phi}\rangle}{\|P|\hat{\phi}\rangle\|_2}. \tag{5.A.2}$$

By the properties of the DPSS taper we have that $\left\|\langle\hat{\phi}|\hat{\phi}^*\rangle\right\|_2^2 \geq 1-\varepsilon$, which implies

$$|\hat{\phi}^*\rangle = \sqrt{1-\varepsilon}|\hat{\phi}\rangle + \sqrt{\varepsilon}|\hat{\phi}^\perp\rangle. \tag{5.A.3}$$

Since QFT is unitary the same holds for the time domain tapers, i.e., $|\phi^*\rangle = \sqrt{1-\varepsilon}|\phi\rangle + \sqrt{\varepsilon}|\phi^\perp\rangle$. We know that when we run the algorithm with the perfect DPSS taper $|\phi\rangle$, the success probability is lower bounded by $1-\varepsilon$. By linearity, the success probability of the algorithm ran with truncated DPSS taper $|\phi^*\rangle$ is lower bounded by $(1-\varepsilon)^2 \geq 1-2\varepsilon$. Hence, by letting $\varepsilon \to \varepsilon/2$ we can achieve ε error probability even with imperfect taper preparation. ■

We note that the lemma stated above is similar to Lemma 1 in [CSS18] which states that a Gaussian taper with the reciprocal standard deviation, $1/\sigma$, as the width of its central

lobe, could also be used in place of a DPSS taper to concentrate phase measurements. However, it would not have the optimal concentration of probability in the central lobe that we observe for the DPSS taper.

We now provide an explicit description of the taper state preparation method that closely follows App. B of [CSS18]. We first begin by classically computing the N' amplitudes that characterize the central lobe of $|\hat{\phi}\rangle$ using classical numerical methods. Next, we use these N' amplitudes to prepare a state on a register of $\log_2 N'$ qubits using the preparation method described in [SBM05]. Subsequently, we append $p - \log_2 N'$ qubits initialized to $|0\rangle^{\otimes(p-\log_2 N')}$ to the ancilla register to form a p-qubit ancilla register. The N' amplitudes form an off-center lobe in the frequency domain, so we need to center these amplitudes on the p-qubit register to form a central lobe. In App. C of [CSS18], the authors provide an explicit circuit involving CNOT and X gates that performs the desired centering by permuting and mapping the original N' basis states $\{|j\rangle \mid 0 \leq j \leq N' - 1\}$ to the new N' basis states $\{|j\rangle \mid (N - N')/2 \leq j \leq (N + N')/2 - 1\}$ in the p-qubit ancilla register. Lastly, we perform the centered Fourier transform operator $U_{\text{C-QFT}}$ that was proposed in App. B of [CSS18].

Given $\varepsilon > 0$, where ε is the failure probability of the tQPE algorithm, let us define $U_{\text{C-QFT}}$ as such:

$$U_{\text{C-QFT}} \coloneqq \pi U_{\text{QFT}} \pi, \tag{5.A.4}$$

where π is a p-qubit cyclic permutation operator that maps $|j\rangle$ to $|j - N/2 \bmod N\rangle$, and U_{QFT} is the p-qubit Fourier transform operator. The cyclic permutation operator π can be implemented with $U_{\text{QFT}} Z U_{\text{QFT}}^{-1}$, where Z is a diagonal operator defined as $\text{diag}(Z) \coloneqq [1, -1, 1, -1, \ldots, 1, -1]$ as elaborated in App. B of [CSS18]. Therefore, the overall action of the centered Fourier transform operator $U_{\text{C-QFT}}$ can be understood as a shifted QFT that

accounts for the new central lobe, and it outputs the approximate DPSS taper state $|\phi^*\rangle$ that is $O(\varepsilon)$-close to $|\phi\rangle$ in the ancilla register of the tQPE quantum circuit. Readers can find more details about the derivation in Sec. IV-D and App. B of [CSS18].

Now, we analyze the complexity of the taper state preparation protocol. Ref. [SBM05] introduced a technique for preparing a state on a register of $\log N'$ qubits with a gate complexity of $O(N')$ from N' amplitudes. Note that their technique requires some classical computation, which we disregard, along with the classical computation of the N' amplitudes of the central lobe of $|\hat{\phi}\rangle$. For the quantum circuit required to center the amplitudes, it has been shown in App. C of [CSS18] that the number of two-qubit gates required is $\log\left(\frac{N-N'}{2}\right) = O(\log(N-N'))$. Lastly, we provide the gate complexity of $U_{\text{C-QFT}}$. We begin by providing its alternative expression:

$$U_{\text{C-QFT}} = \pi U_{\text{QFT}} \pi = \left(U_{\text{QFT}} Z U_{\text{QFT}}^{-1}\right) U_{\text{QFT}} \left(U_{\text{QFT}} Z U_{\text{QFT}}^{-1}\right) = U_{\text{QFT}} Z U_{\text{QFT}} Z U_{\text{QFT}}^{-1}. \tag{5.A.5}$$

Notice that we require three applications of QFT which has been shown in [Cop94] to have gate complexity $O\left(p^2\right) = O\left(\log^2 N\right)$. This gives us an overall gate complexity of $O\left(N' + \log(N-N') + \log^2 N\right) = O\left(N' + \log^2 N\right)$ for the state preparation protocol. Using the fact that $N' = O\left(\log^2(1/\varepsilon)\right)$ and $N = 2^p = 2^m \cdot 2^\ell = 2^{O(\log\log 1/\varepsilon)} \cdot 2^{O(\log 1/\delta)}$, we can state the gate complexity of the state preparation protocol C_ϕ in terms of δ and ε:

$$C_\phi = O\left(\log^2(1/\varepsilon) + \left[\log\log(1/\varepsilon) + \log(1/\delta)\right]^2\right) \tag{5.A.6}$$

We also note that the state preparation protocol does not require any additional ancilla qubits other than the p-qubits for the taper state register.

Having discussed how we can prepare an approximate DPSS taper $|\phi^*\rangle$ that is $O(\varepsilon)$-close to the DPSS taper $|\phi\rangle$, we now proceed to provide guarantees for the performance of $|\phi^*\rangle$. To do that we first write the approximate DPSS taper in terms of the DPSS taper:

$$|\phi^*\rangle = \sqrt{1 - O(\varepsilon^2)}|\phi\rangle + O(\varepsilon)\left|\phi^\perp\right\rangle, \tag{5.A.7}$$

where $\langle\phi|\phi^{\perp}\rangle = 0$.

Now, we consider the success probability of our tQPE algorithm when we use the approximate taper. We assume that the probability of error of the exact taper $|\phi\rangle$ is ε, that is

$$|\phi\rangle|\psi_\theta\rangle \xrightarrow{QPE} \sqrt{1-\varepsilon}\sum_{i:\left|\tilde{\theta}_i-\theta\right|\leq\delta}|\tilde{\theta}_i\rangle|\psi_\theta\rangle + \sqrt{\varepsilon}\sum_{i:\left|\tilde{\theta}_i-\theta\right|>\delta}|\tilde{\theta}_i\rangle|\psi_\theta\rangle. \tag{5.A.8}$$

For $\left|\phi^{\perp}\right\rangle$ we assume the worst possibility, i.e. that it returns a δ-close phase estimate with zero probability. Then the approximate taper returns a δ-close estimate with probability at least

$$P^{*}_{\text{success}} \geq \left(1-O\left(\varepsilon^2\right)\right)(1-\varepsilon) = 1-\varepsilon-O\left(\varepsilon^2\right)+O\left(\varepsilon^3\right) = 1-O(\varepsilon). \tag{5.A.9}$$

5.B Ideal Case

The optimal taper for tQPE is the one that maximizes the probability of outputting the value of a phase estimate of the form k/N that is δ-close to θ. To simplify the analysis we have set $\delta = 2^{-(l+1)}$ for some integer ℓ. In this case the $2K+1$ discrete frequencies closest to θ are indeed δ-close to the true phase for $K = 2^{m-1}$, where $p = \ell + m$ is the total number of ancilla qubits. At most a single discrete frequency that is δ-close is left out, and that frequency has the smallest probability of being output by the algorithm. Thus without introducing much error we formulate the optimization problem in terms of maximizing the probability of outputting the closest $2K+1$ discrete estimates, see (5.5.3). Since this is a constrained problem, we used the Lagrangian formulation to solve it. For ease of reference, we restate the corresponding Lagrangian (5.5.4) from the main text below:

$$\mathcal{L}(|\phi\rangle,\lambda) = \sum_{j=-K}^{K}\left|\hat{\phi}\left(\Delta+\frac{j}{N}\right)\right|^2 + \lambda\left(\sum_{n=0}^{N-1}|\phi[n]|^2-1\right). \tag{5.B.1}$$

To find the optimal taper for the original optimization problem, given by (5.5.3), we need to find the stationary point of $\mathcal{L}$ that maximizes the objective function of (5.5.3). But before doing so, we expand the expression on the right-hand side of the above equation by plugging in the definition of $\hat{\phi}$, given by (5.4.6):

$$\mathcal{L}(\phi, \lambda) = \frac{1}{N} \sum_{j=-K}^{K} \sum_{n,n'=0}^{N-1} \phi[n]\phi^*[n'] e^{2\pi i \left(\Delta + \frac{j}{N}\right)(n-n')} + \lambda \left(\sum_{n=0}^{N-1} \phi[n]\phi^*[n] - 1 \right) \tag{5.B.2}$$

$$= \frac{1}{N} \sum_{j=-K}^{K} \sum_{n,n'=0}^{N-1} e^{2\pi i \Delta (n-n')} \phi[n]\phi^*[n'] e^{-2\pi i (j/N)(n'-n)} + \lambda \left(\sum_{n=0}^{N-1} \phi[n]\phi^*[n] - 1 \right). \tag{5.B.3}$$

Then, to find the stationary points of $\mathcal{L}$, we set all partial derivatives to zero. First, we differentiate $\mathcal{L}$ with respect to $\phi^*[m]$ for all $m \in \{0, \ldots, N-1\}$ then set the expression to zero:

$$\frac{\partial \mathcal{L}}{\partial \phi^*[m]} = \frac{1}{N} \sum_{j=-K}^{K} \sum_{n=0}^{N-1} e^{2\pi i \Delta (m-n)} \phi[n] e^{-2\pi i (j/N)(m-n)} + \lambda \phi[m] = 0 \tag{5.B.4}$$

$$\implies \frac{1}{N} \sum_{j=-K}^{K} \sum_{n=0}^{N-1} e^{2\pi i \Delta (n-m)} \phi[n] e^{-2\pi i (j/N)(m-n)} = \lambda \phi[m] \tag{5.B.5}$$

$$\implies \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i \Delta (n-m)} \left(\sum_{j=-K}^{K} e^{-2\pi i (j/N)(m-n)} \right) \phi[n] = \lambda \phi[m] \tag{5.B.6}$$

$$\implies \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i \Delta (n-m)} \left(\frac{\sin(\pi(m-n)(2K+1)/N)}{\sin(\pi(m-n)/N)} \right) \phi[n] = \lambda \phi[m] \tag{5.B.7}$$

As is evident, the last equation is an eigenvalue equation with eigenvalue λ and eigenvector $|\phi\rangle$. We rewrite this equation more concisely as follows:

$$\mathcal{B}_\Delta |\phi\rangle = \lambda |\phi\rangle, \tag{5.B.8}$$

where we define the matrix $\mathcal{B}_\Delta$ as

$$\mathcal{B}_\Delta[m, n] := \frac{1}{N} e^{2\pi i \Delta (n-m)} \left(\frac{\sin(\pi(m-n)(2K+1)/N)}{\sin(\pi(m-n)/N)} \right). \tag{5.B.9}$$

Next, we differentiate $\mathcal{L}$ with respect to the Lagrange multiplier, λ, and also set it to zero:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{n=0}^{N-1} \phi[n]\phi^*[n] - 1 = 0 \tag{5.B.10}$$

$$\implies \sum_{n=0}^{N-1} \phi[n]\phi^*[n] = 1, \tag{5.B.11}$$

reproducing the normalization constraint.

Now, by plugging all the stationary points $(|\phi\rangle, \lambda)$ that satisfy the conditions (5.B.7) and (5.B.11) into the objective function of the original optimization problem (5.5.3), we find:

$$\sum_{j=-K}^{K} \left|\hat{\phi}\left(\Delta + \frac{j}{N}\right)\right|^2 = \lambda. \tag{5.B.12}$$

In other words, this means that the stationary point $(|\phi\rangle, \lambda)$ that maximizes the objective function is the eigenvector of B_Δ with the maximum eigenvalue λ, and this value corresponds to the success probability of the algorithm.

5.C Relationship Between P-DPSS Tapers and Ideal Case Optimal Tapers

We compare the operator B_Δ defined in (5.B.9) to the bandlimiting operator $\mathcal{B}_K : \mathbb{C}^N \to \mathbb{C}^N$ described in [ZKD+17]. Suppose $K \in \mathbb{N}$ and $2K + 1 < N$. The bandlimiting operator, $\mathcal{B}_K$, is defined such that it zeros out the discrete frequencies of a signal $\phi \in \mathbb{C}^N$ that lie outside of the range $\{-K \equiv N - K \pmod N, \ldots, 0, \ldots, K\}$. Formally, this operator does the following:

$$\left(\mathcal{B}_K(\phi)\right)[m] \coloneqq \frac{1}{\sqrt{N}} \sum_{n \in S} \hat{\phi}[n] e^{\frac{2\pi i m n}{N}}, \tag{5.C.1}$$

where $m \in \{0, \ldots, N-1\}$ and $S = \{0, 1, \ldots, K\} \cup \{N-K, \ldots, N-1\}$. In the time domain this can be written as

$$\left(\mathcal{B}_K\left(\phi\right)\right)[m] = \frac{1}{N}\sum_{n=0}^{N-1} \frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\sin\left(\pi(m-n)/N\right)}\phi[n]. \tag{5.C.2}$$

Note that this expression only differs from the matrix description stated in (5.B.7) by a factor of $e^{2\pi i\Delta(m-n)}$ in each summand. We now introduce the timelimiting operator $\mathcal{T}_N : \mathbb{C}^N \to \mathbb{C}$ that is defined similarly as the following:

$$\left(\mathcal{T}_N\left(\phi\right)\right)[n] := \begin{cases} \phi[n], & n \in \{0, 1, \ldots, N-1\} \\ 0, & \text{else.} \end{cases} \tag{5.C.3}$$

In [ZKD+17], it was shown that the P-DPSSs are the eigensequences of the operator $\mathcal{T}_N\mathcal{B}_K\mathcal{T}_N$. In other words, the P-DPSSs are discrete sequences that contain a particular eigensequence that not only fulfills the time-limiting constraints of having non-zero values only on the indices $0, 1, \ldots, N-1$, but also has maximal frequency concentration within our window of interest, $2K+1$. Notice that the time-limiting property and the property of having maximal frequency concentration around $2K+1$ lattice points are the same properties that we want the optimal tapers to possess. These similarities make it interesting for us to study the eigenvalue spectrum of the P-DPSSs because it may suggest an upper bound on what we can do with our ideal tapers. The bounds for the eigenvalues of the P-DPSS's with respect to the operator $\mathcal{T}_N\mathcal{B}_K\mathcal{T}_N$, proven in [ZKD+17], are stated in the following theorem, which we restate with our notation.

Theorem 72 *[ZKD+17, Theorem 1] Suppose $K, N \in \mathbb{N}$ and $W = \frac{2K+1}{2N} < \frac{1}{2}$. Also, suppose that $\lambda^{(i)}$ is the i^{th} largest eigenvalue of $\mathcal{T}_N\mathcal{B}_K\mathcal{T}_N$. Then for any $\varepsilon \in \left(0, \frac{1}{2}\right)$, we have*

$$\lambda^{(2\lfloor NW\rfloor - \lceil R(N,\varepsilon)\rceil)} \geq 1 - \varepsilon, \tag{5.C.4}$$

where

$$R(N,\varepsilon) = \left(\frac{4}{\pi^2}\log(8N)+6\right)\log\left(\frac{16}{\varepsilon}\right) + 2\max\left\{\frac{-\log\left(\frac{\pi}{32}\left(\left(\frac{N}{N-1}\right)^2-1\right)\varepsilon\right)}{\log\left(\frac{N}{N-1}\right)}, 0\right\}. \tag{5.C.5}$$

Although Thm. 72 is stated for the matrix that corresponds to $\mathcal{T}_N\mathcal{B}_K\mathcal{T}_N$, the authors of [ZKD+17] showed that the eigenvalues of $\mathcal{T}_N\mathcal{B}_K\mathcal{T}_N$ and $\mathcal{B}_K$ are the same. Note that the matrix representation of the eigenvalue equation given by (5.B.7) differs from the matrix representation of $\mathcal{B}_K$ by some complex phase. However, we demonstrate in the following proposition that they share the same spectrum by showing that we can adapt the eigensequences of $\mathcal{B}_K$ to obtain the eigensequences that satisfy (5.B.7). In other words, we can use the analytical bounds for the eigenvalues of the P-DPSSs to bound the eigenvalues for our ideal tapers. Below, we refer to ϕ_i^Δ, the eigenvector of the operator described in (5.B.7) with eigenvalue $\lambda^{(i)}$, as the quantum periodic discrete prolate spheroidal sequence (QP-DPSS), where $-\frac{1}{2N} \le \Delta \le \frac{1}{2N}$.

Proposition 73 *Suppose $\{(\lambda^{(i)},\psi_i)\}_i$ represents the set of eigenvalues and eigensequences of $\mathcal{B}_K$. Then, for some $-\frac{1}{2N} \le \Delta \le \frac{1}{2N}$, $\{(\lambda^{(i)},\phi_i^\Delta)\}_i$ is the set of eigenvalues and eigensequences of B_Δ given in (5.B.9), where*

$$\phi_i^\Delta[n] = e^{2\pi i\Delta n}\psi_i[n]. \tag{5.C.6}$$

Proof. We first restate the eigenvalue equation described in (5.B.7) with the proposed eigensequence, $\phi_i^\Delta[n]$.

$$\begin{aligned}
&\frac{1}{N}\sum_n e^{2\pi i\Delta(m-n)}\left(\frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\sin\left(\pi(m-n)/N\right)}\right)\phi_i^\Delta[n] \\
&= \frac{1}{N}\sum_n e^{2\pi i\Delta(m-n)}\left(\frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\sin\left(\pi(m-n)/N\right)}\right)e^{2\pi i\Delta n}\psi_i[n] &(5.C.7)\\
&= \frac{e^{2\pi i\Delta m}}{N}\sum_n\left(\frac{\sin\left(\pi(m-n)(2K+1)/N\right)}{\sin\left(\pi(m-n)/N\right)}\right)\psi_i[n] &(5.C.8)
\end{aligned}$$

$$= e^{2\pi i \Delta m} \lambda^{(i)} \psi_i[m] \tag{5.C.9}$$

$$= \lambda^{(i)} \phi_i^{\Delta}[m] \tag{5.C.10}$$

where the second to last equality follows from the fact that ψ_i is an eigensequence of $\mathcal{B}_K$ with eigenvalue $\lambda^{(i)}$. This concludes the proof. ■

Note that the eigensequences $\{\phi_i^{\Delta}\}_i$ are what we refer to as QP-DPSSs in the main text. It is clear from the development above that transforming the tophat taper in the way shown in (5.C.6) would give us a taper that would succeed with probability 1. In fact, it is known that the P-DPSSs have up to $2K + 1$ eigensequences with eigenvalue 1. While such a transformation would be amazing, we remark that it requires explicit knowledge of Δ which is not something that we can reasonably assume to possess (i.e., it would assume knowledge of the phase that we are trying to determine). Nevertheless, we provide some further discussion below to allow the reader to understand the deeper connections between P-DPSSs and optimal quantum tapers. In particular, we note that it may be possible to construct optimal tapers with greater efficiency by taking linear combinations of optimal tapers due to the possibility of the existence of a much more efficient description of the tapers. In fact, if one is willing to give up on the goal of achieving a success probability of 1, there are possibly other tapers with a much more efficient representation. From the development above in Thm. 72 and Prop. 73, we have the following result.

Corollary 74 *Given $N = 2^{l+m}$, where l is the number of qubits needed to encode the target phase and m is the number of extra qubits necessary to guarantee that with probability at least $1 - \varepsilon$, the desired precision will be met. Using the QP-DPSS taper with the largest eigenvalue, we have*

$$m = O\left(\log\log 1/\delta + \log\log 1/\varepsilon\right). \tag{5.C.11}$$

Proof. We are interested in the largest eigenvalue, i.e., $\lambda^{(0)}$. Therefore, from (5.C.4), we

first set $2\lfloor NW \rfloor - \lceil R(N,\varepsilon) \rceil = 0$ to obtain $\lceil R(N,\varepsilon) \rceil = 2\lfloor NW \rfloor \leq 2NW$. For all $N \geq 2$, the first term in (5.C.5) dominates and therefore, (5.C.5) reduces to

$$R(N,\varepsilon) = O\left(\log\left(\frac{1}{\varepsilon}\right)\log N\right). \tag{5.C.12}$$

This naturally implies that

$$NW = O\left(\log\left(\frac{1}{\varepsilon}\right)\log N\right). \tag{5.C.13}$$

Recall that m, the additional qubits required for a given precision, can be expressed as $\log(NW)$ because $W = \frac{2^m}{2^{l+m}}$ is the fraction of the lattice for which the taper has significant support. This gives

$$\begin{aligned}
2^m &= O\left(\log\left(\frac{1}{\varepsilon}\right)\log N\right) && (5.C.14)\\
&= O\left((m+l)\log(1/\varepsilon)\right) && (5.C.15)\\
&= O\left(m\log 1/\varepsilon + \log 1/\delta \log 1/\varepsilon\right) && (5.C.16)\\
\implies m &= O\left(\log\left(m\log 1/\varepsilon + \log 1/\delta \log 1/\varepsilon\right)\right) && (5.C.17)\\
&= O\left(\log\left(\log 1/\delta \log 1/\varepsilon\right) + \frac{m\log 1/\varepsilon}{\log 1/\delta \log 1/\varepsilon}\right) && (5.C.18)\\
&= O\left(\log\left(\log 1/\delta \log 1/\varepsilon\right) + \frac{m}{\log 1/\delta}\right) && (5.C.19)\\
\implies m - O\left(\frac{m}{\log 1/\delta}\right) &= O\left(\log\left(\log 1/\delta \log 1/\varepsilon\right)\right) && (5.C.20)\\
\implies m &= \frac{O\left(\log\left(\log 1/\delta \log 1/\varepsilon\right)\right)}{1 - O\left(\frac{1}{\log 1/\delta}\right)} && (5.C.21)\\
&= O\left(\log\log 1/\delta + \log\log 1/\varepsilon\right). && (5.C.22)
\end{aligned}$$

The second equality follows by substituting $N = 2^{l+m}$. The fifth equality follows from the fact that $\log(x+y) \leq \log(x) + y/x$ for all $x, y > 0$. This concludes the proof. ■

5.D Average-Case

For ease of reference, we restate the Lagrangian corresponding to the average case from the main text below:

$$\mathcal{L}_{\text{avg}}\left(|\phi\rangle, \lambda\right) = \mathbb{E}_{\Delta\sim\mathcal{D}}\left[\sum_{j=-K}^{K}\left|\hat{\phi}\left(\Delta+\frac{j}{N}\right)\right|^2\right] - \lambda\left(\sum_{n=0}^{N-1}|\phi[n]|^2 - 1\right). \tag{5.D.1}$$

As mentioned before, we focus on the case where Δ is uniformly distributed over the interval $[-\frac{1}{2N}, \frac{1}{2N}]$, i.e., $\mathcal{D}$ is the uniform distribution over this interval. It is reasonable to make this assumption since it is unlikely that the phase values are influenced in a way such that they lie away or close to the lattice points in a systematic way. However, it is possible to generalize the subsequent analysis for other probability distributions, such as the Gaussian distribution.

Please note that the analysis below follows a similar approach outlined in App. 5.B. Now, consider the following:

$$\mathcal{L}_{\text{avg}}\left(|\phi\rangle, \lambda\right) \tag{5.D.2}$$

$$= \int_{-\frac{1}{2N}}^{\frac{1}{2N}} d\Delta\, N \sum_{j=-K}^{K}\left|\hat{\phi}\left(\Delta+\frac{j}{N}\right)\right|^2 - \lambda\left(\sum_{n=0}^{N-1}|\phi[n]|^2 - 1\right) \tag{5.D.3}$$

$$= \frac{1}{N}\int_{-\frac{1}{2N}}^{\frac{1}{2N}} d\Delta\, N \sum_{j=-K}^{K}\sum_{n,n'=0}^{N-1}\phi[n]\phi^*[n']e^{2\pi i(\Delta+\frac{j}{N})(n-n')} - \lambda\left(\sum_{n=0}^{N-1}\phi[n]\phi^*[n] - 1\right) \tag{5.D.4}$$

$$= \sum_{n,n'=0}^{N-1}\phi[n]\phi^*[n']\int_{-\frac{1}{2N}}^{\frac{1}{2N}} d\Delta\, e^{2\pi i\Delta(n-n')}\sum_{j} e^{-2\pi i(j/N)(n'-n)} - \lambda\left(\sum_{n=0}^{N-1}\phi[n]\phi^*[n] - 1\right) \tag{5.D.5}$$

$$= \sum_{n,n'=0}^{N-1}\phi[n]\phi^*[n']\frac{\sin\left(2\pi\left(1/2N\right)\left(n'-n\right)\right)}{\pi(n'-n)}\frac{\sin\left(\pi(n'-n)(2K+1)/N\right)}{\sin\left(\pi(n'-n)/N\right)} - \lambda\left(\sum_{n=0}^{N-1}\phi[n]\phi^*[n] - 1\right) \tag{5.D.6}$$

$$= \sum_{n,n'=0}^{N-1}\phi[n]\phi^*[n']\underbrace{\frac{\sin\left(\pi(n'-n)(2K+1)/N\right)}{\pi(n'-n)}}_{=:C(n,n')} - \lambda\left(\sum_{n=0}^{N-1}\phi[n]\phi^*[n] - 1\right) \tag{5.D.7}$$

Then, to find stationary points of $\mathcal{L}_{\text{avg}}$, we set all of its partial derivatives to zero. First, we differentiate $\mathcal{L}_{\text{avg}}$ with respect to $\phi^*[m]$ for all $m \in \{0, \ldots, N-1\}$ and set it to zero:

$$\frac{\partial \mathcal{L}_{\text{avg}}}{\partial \phi^*[m]} = \sum_{n=0}^{N-1} \phi[n] C(n,m) - \lambda \phi[m] = 0 \tag{5.D.8}$$

$$\implies \sum_{n=0}^{N-1} \phi[n] C(n,m) = \lambda \phi[m]. \tag{5.D.9}$$

Note that the above equation is an eigenvalue equation, and its eigenvectors are DPSSs, previously studied in classical signal processing [Sle78]. Next, differentiating $\mathcal{L}$ with respect to the Lagrange multiplier λ and setting it to zero leads to the normalization constraint for the taper:

$$\frac{\partial \mathcal{L}_{\text{avg}}}{\partial \lambda} = -\sum_{n=0}^{N-1} \phi[n]\phi^*[n] + 1 = 0 \tag{5.D.10}$$

$$\implies \sum_{n=0}^{N-1} \phi[n]\phi^*[n] = 1, \tag{5.D.11}$$

Next, by plugging all the stationary points $(|\phi\rangle, \lambda)$ that satisfy the conditions (5.D.9) and (5.D.11) into the objective function of $\mathcal{L}_{\text{avg}}$, we get the following:

$$\mathbb{E}_{\Delta \sim \mathcal{D}} \left[\sum_{j=-K}^{K} \left| \hat{\phi}\left(\Delta + \frac{j}{N}\right) \right|^2 \right] = \lambda. \tag{5.D.12}$$

In other words, this means that the stationary point $(|\phi\rangle, \lambda)$ that maximizes the objective function is the eigenvector of C with the maximum eigenvalue λ. This corresponds to the DPSS with the maximum eigenvalue, which additionally corresponds to the success probability of the algorithm.

5.E Classical Signal Analysis Derivation of DPSS as Optimal Tapers

Following [Sle78, KRD21], the discrete time Fourier transform (DTFT), $\hat{x} \in L_2([-\frac{1}{2}, \frac{1}{2}])$, of a discrete signal, $x \in l_2(\mathbb{Z})$, is given by

$$\hat{x}(f) \coloneqq \sum_{n=-\infty}^{\infty} x[n] e^{-2\pi i f n}, \quad f \in \left[-\frac{1}{2}, \frac{1}{2}\right], \tag{5.E.1}$$

with the inverse transform given by

$$x[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{x}(f) e^{2\pi i f n} df \ . \tag{5.E.2}$$

From these definitions, we can see that $x, x' \in l_2(\mathbb{Z})$ satisfy the Parseval-Plancherel equality, $\langle x, x' \rangle_{l_2(\mathbb{Z})} = \langle \hat{x}, \hat{x}' \rangle_{L_2([-1/2,1/2])}$. Here, we denote $\langle \cdot, \cdot \rangle_{\mathcal{A}}$ as the inner product defined on the vector space $\mathcal{A}$. We say that $x \in l_2(\mathbb{Z})$ is time-limited to $n \in \{0, \ldots, N-1\}$ if $x[n] = 0$ for all $n \in \mathbb{Z} \backslash \{0, \ldots, N-1\}$. Furthermore, we say that $x \in l_2(\mathbb{Z})$ is band-limited to $|f| \leq W$ if $\hat{x}(f) = 0$ for $|f| > W$, where $W \in (0, 1/2)$.

The aim here is to find discrete functions $x[n]$ that are time-limited to $n \in \{0, \ldots, N-1\}$ and are maximally band-limited to the frequency band $|f| \leq W$. We can formulate this as an optimization problem in the following way:

$$\max_{x \in l_2(\mathbb{Z})} \quad \int_{-W}^{W} |\hat{x}(f)|^2 df \tag{5.E.3}$$

$$\text{subject to} \quad \|x\|_{l_2(\mathbb{Z})}^2 = 1, \tag{5.E.4}$$

$$x[n] = 0 \text{ for all } n \in \mathbb{Z} \backslash \{0, \ldots, N-1\}. \tag{5.E.5}$$

We can simplify things conceptually by defining the following two operators: the time-limiting operator $\mathcal{T}_N$ as

$$(\mathcal{T}_N x)[n] \coloneqq \begin{cases} x[n] & \text{if } \ n \in \{0, \ldots, N-1\} \\ 0 & \text{if } \ n \in \mathbb{Z} \backslash \{0, \ldots, N-1\} \end{cases} \tag{5.E.6}$$

and the band-limiting operator $\mathcal{B}_W$ as

$$(\mathcal{B}_W x)[n] := \sum_{l=-\infty}^{\infty} \frac{\sin[2W(l-n)]}{\pi(l-n)} x[l] \quad \text{for} \quad n \in \mathbb{Z} \, . \tag{5.E.7}$$

Note that the DTFT of the band-limiting operator is $\widehat{\mathcal{B}_W x}(f) = \hat{x}(f)$ for $|f| \leq W$ and $\widehat{\mathcal{B}_W x}(f) = 0$ for $|f| > W$.

With these definitions, we see that the integral in (5.E.3) can be represented as

$$\int_{-W}^{W} |\hat{x}(f)|^2 df = \langle \hat{x}, \widehat{\mathcal{B}_W x} \rangle_{L_2([-\frac{1}{2},\frac{1}{2}])} \, . \tag{5.E.8}$$

From the Parseval-Plancherel theorem, we have

$$\langle \hat{x}, \widehat{\mathcal{B}_W x} \rangle_{L_2([-\frac{1}{2},\frac{1}{2}])} = \langle x, \mathcal{B}_W x \rangle_{l_2(\mathbb{Z})} \, . \tag{5.E.9}$$

This implies that

$$\begin{aligned} \langle x, \mathcal{B}_W x \rangle_{l_2(\mathbb{Z})} &= \langle \mathcal{T}_N x, \mathcal{B}_W \mathcal{T}_N x \rangle_{l_2(\mathbb{Z})} & (5.E.10) \\ &= \langle x, \mathcal{T}_N \mathcal{B}_W \mathcal{T}_N x \rangle_{l_2(\mathbb{Z})} \, , & (5.E.11) \end{aligned}$$

with the last expression, (5.E.11), holding because $\mathcal{T}_N$ is self-adjoint.

From (5.E.3), (5.E.8), and (5.E.11), we can see that the eigenvector, ϕ_0, corresponding to the maximum eigenvalue, λ_0, of the matrix given by

$$[\mathcal{T}_N \mathcal{B}_W \mathcal{T}_N]_{l,n} = \frac{\sin(2\pi W(l-n))}{\pi(l-n)}; \quad l, n \in \{0, \ldots, N-1\} \tag{5.E.12}$$

solves the maximization problem. This classical signal processing result gives the same optimal tapers that we find above for the quantum average-case optimal taper.

5.F Relationship Between Tapers and the Fourier Convolution Theorem

Here, we aim to show a conceptually useful way to understand tapering, and hence tQPE, originating in the classical signal processing community. We begin by deriving the Fourier convolution theorem in a form useful for our discussion below. Starting with two functions of time, ψ and x, we have that their pointwise product (denoted by '$\cdot$')

$$\begin{aligned}
\psi(t)\cdot x(t) &= \left(\int_{-\infty}^{\infty}\hat{\psi}(g)e^{-2\pi igt}dg\right)\cdot\left(\int_{-\infty}^{\infty}\hat{x}(f)e^{-2\pi ift}df\right) && (5.F.1)\\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{\psi}(g)\hat{x}(f)e^{-2\pi i(g+f)t}dgdf && (5.F.2)\\
&= \int_{-\infty}^{\infty}\underbrace{\left(\int_{-\infty}^{\infty}\hat{\psi}(g)\hat{x}(h-g)dg\right)}_{(\hat{\psi}*\hat{x})}e^{-2\pi iht}dh\,, && (5.F.3)
\end{aligned}$$

where $*$ denotes a convolution.

Switching notation and denoting the continuous Fourier transform as $\mathcal{F}$, we see from above

$$\begin{aligned}
\psi(t)\cdot x(t) &= \mathcal{F}^{-1}\{\mathcal{F}\{\psi\}*\mathcal{F}\{x\}\} && (5.F.4)\\
&\rightarrow \mathcal{F}\{\psi(t)\cdot x(t)\} = \mathcal{F}\{\psi\}*\mathcal{F}\{x\}\,. && (5.F.5)
\end{aligned}$$

Note that here, to be consistent with the signs of the complex exponentials, we use the opposite convention for our transform than is usual.

For tQPE, the amplitudes of the state encoded on the ancilla register before the inverse QFT are $\phi[n]e^{2\pi i\theta n}$. Here, $\theta\in[0,2\pi]$, but the taper, ϕ is a time-limited sequence with discrete support at equally spaced times, $n\Delta t$. We can represent this sequence as a continuous distribution, $\phi(t)\cdot\text{Ш}(t)$, in the limit as $n\rightarrow\infty$, and $\Delta t\rightarrow 0$, where $\text{Ш}(t)=\sum_{n=0}^{N-1}\delta(t-n\Delta t)$ is a

Dirac comb. In this limit, $\phi[n]e^{2\pi i\theta n} \to \left((\phi(t)\cdot \text{Ш}(t))\cdot e^{2\pi i\theta t}\right) = \left(\left(\phi(t)\sum_{n=0}^{N-1}\delta(t-n\Delta t)\right)\cdot e^{2\pi i\theta t}\right) = \left(\left(\sum_{n=0}^{N-1}\phi(t)\delta(t-n\Delta t)\right)\cdot e^{2\pi i\theta t}\right)$. To apply the Fourier convolution theorem, we take $x(t) = e^{2\pi i\theta t}$ giving $\mathcal{F}\{x\}(f) = \delta(\theta - f)$, and the discrete and finite in time distribution, $\psi(t) = \sum_{n=0}^{N-1}\phi(t)\delta(t-n\Delta t)$. This then gives us

$$\begin{aligned}\mathcal{F}\{\psi\}(f) &= \int_{-\infty}^{\infty}\sum_{n=0}^{N-1}\phi(t)\delta(t-n\Delta t)e^{-2\pi itf}dt &(5.F.6)\\ &= \sum_{n=0}^{N-1}\int_{-\infty}^{\infty}\phi(t)e^{-2\pi itf}\delta(t-n\Delta t)dt &(5.F.7)\\ &= \sum_{n=0}^{N-1}\phi[n]e^{-2\pi in\Delta tf}\ . &(5.F.8)\end{aligned}$$

Rescaling $f\Delta t \to f$ and similarly for θ, this gives us,

$$\begin{aligned}\mathcal{F}\{\psi(t)\cdot x(t)\}(f) &= \{\mathcal{F}\{\psi\} * \mathcal{F}\{x\}\}(f) &(5.F.9)\\ &= \left\{\left(\sum_{n=0}^{N-1}\phi[n]e^{-2\pi inf}\right) * \delta(\theta - f)\right\}(f) &(5.F.10)\\ &= \sum_{n=0}^{N-1}\phi[n]e^{-2\pi in(\theta-f)} &(5.F.11)\\ &= \mathcal{F}\{\phi\}(\theta - f) &(5.F.12)\end{aligned}$$

Switching back to our previous notation, for tQPE, we have

$$\mathcal{F}\{\psi(t)\cdot x(t)\} = \hat{\phi}(\theta - f)\ , \tag{5.F.13}$$

the coefficient in (5.4.7). tQPE samples this with the QFT at discrete frequencies, k, as $\text{QFT}^\dagger[\psi(t)\cdot x(t)] = \hat{\phi}_k(\theta - k/N)$.

5.G Error Analysis for QPE with Uncomputation

In this section, we bound the error in applications of QPE that require uncomputation of the estimated phase. The analysis below applies to all coherent QPE algorithms. In applications that require coherent QPE, like HHL and Quantum Metropolis Sampling, QPE is used in order to approximately implement a unitary of the form:

$$cT_{CA} \coloneqq \sum_j T(\theta_j)_C \otimes |\psi_j\rangle\langle\psi_j|_A, \tag{5.G.1}$$

where A and C are two registers (see the left-hand side of Figure 5.6) and $T(\theta)$ is a unitary operator parametrized by $\theta \in [0, 2\pi)$. Furthermore, $\{|\psi_j\rangle\}_j$ are the eigenvectors of some unitary U with the corresponding eigenvalues $\{e^{2\pi i\theta_j}\}_j$. We will assume that T is L_T-Lipschitz continuous for some $L_T > 0$:

$$\|T(\theta) - T(\theta')\| \leq L_T \min(|\theta - \theta'|, 2\pi - |\theta - \theta'|). \tag{5.G.2}$$

We approximately implement the unitary cT using the following algorithm (see the right-hand side of Figure 5.6):

1. Add an ancillary register B and initialize it in the all-zeros state. That is, $|0\rangle^{\otimes p}$ for some $p \in \mathbb{N}$.
2. Apply the unitary Q (any QPE algorithm; this includes the tQPE algorithm) on the registers A and B such that the state of the system is in register A and the phase estimate is stored in register B. Here, Q is defined as follows:

$$Q_{BA} \coloneqq \sum_{j,x,y} f_{jxy}|x\rangle\langle y|_B \otimes |\psi_j\rangle\langle\psi_j|_A, \tag{5.G.3}$$

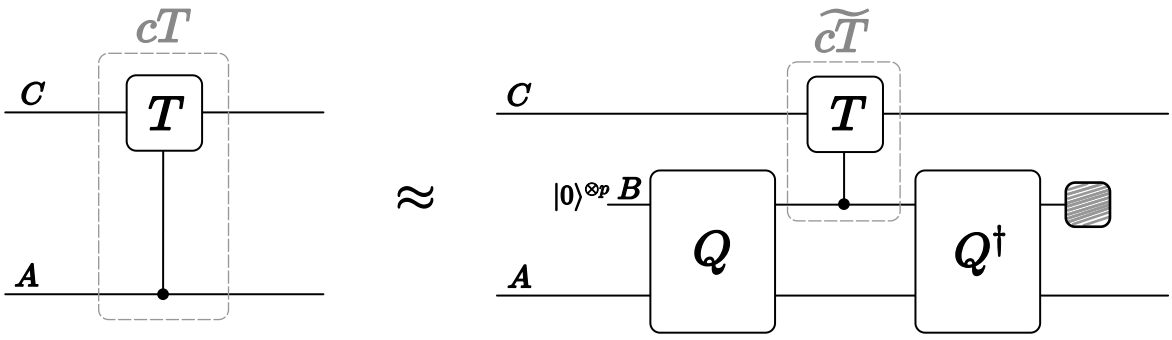


Figure 5.6: The unitary cT (left) is approximated by the quantum channel (right) inside the dashed box. Note that Q can be any QPE algorithm. For tQPE, it would include the state preparation unitary for the taper on the ancilla register.

where f_{jxy} is some probability density that depends on the QPE algorithm one employs. This probability density satisfies the following property for some $\delta, \varepsilon > 0$:

$$\sum_{x:|\theta_j - (x-y)/2^p| > \delta} |f_{jxy}|^2 \leq \varepsilon. \tag{5.G.4}$$

3. Apply the following controlled unitary on registers B and C with the control on register B:

$$\widetilde{cT}_{CB} \coloneqq \sum_x T(x/2^p)_C \otimes |x\rangle\langle x|_B, \tag{5.G.5}$$

4. Apply $Q^\dagger$ on the registers A and B.

5. Trace out register B.

Note that although cT and $\widetilde{cT}$ act in a similar way on the C register, the former controls that action on the A register and the latter on the B register. Since the control on the B register is with respect to the computational basis, it is straightforward to implement.

Now, before proceeding with the error analysis of the above algorithm (Theorem 75),

we define the quantum channel versions of the unitaries cT, Q, and $\widetilde{cT}$:

$$\mathcal{T}(\cdot) \coloneqq cT(\cdot)cT^{\dagger} \tag{5.G.6}$$

$$\mathcal{Q}(\cdot) \coloneqq Q(\cdot)Q^{\dagger} \tag{5.G.7}$$

$$\widetilde{\mathcal{T}}(\cdot) \coloneqq \widetilde{cT}(\cdot)\widetilde{cT}^{\dagger}. \tag{5.G.8}$$

We also define an appending channel $\mathcal{A}$, which essentially appends the all-zeros state to the input state, to represent Step 1 of the above algorithm:

$$\mathcal{A}(\rho_D) \coloneqq \rho_D \otimes |0\rangle\langle 0|_B. \tag{5.G.9}$$

Note that in the above equation and in what follows, we use a shorthand $|0\rangle\langle 0|_B \equiv |0\rangle\langle 0|_B^{\otimes p}$ for simplicity.

With the above notions in place, we now present our main error analysis in the following theorem.

Theorem 75 *The following holds:*

$$\frac{1}{2}\left\|\mathcal{T}_{CA} - \mathrm{Tr}_B \circ \mathcal{Q}^{\dagger}_{BA} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA}\right\|_{\diamond} \leq 2L_T\delta + 4\varepsilon. \tag{5.G.10}$$

Here, Tr_B *denotes the partial-trace channel, where the register* B *is traced out.*

Proof. Consider that

$$\begin{aligned}
&\frac{1}{2}\left\|\mathcal{T}_{AC} - \mathrm{Tr}_B \circ \mathcal{Q}^{\dagger}_{BA} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA}\right\|_{\diamond} \\
&\qquad = \sup_{\rho_{RCA}} \frac{1}{2}\left\|\left(\mathcal{I}_R \otimes \mathcal{T}_{AC}\right)(\rho_{RCA}) - \left(\mathcal{I}_R \otimes \mathrm{Tr}_B \circ \mathcal{Q}^{\dagger}_{BA} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA}\right)(\rho_{RCA})\right\|_1,
\end{aligned} \tag{5.G.11}$$

where R is a reference system of arbitrary dimension and $\mathcal{I}_R$ is the identity channel acting on R. Expanding the first term of the right-hand side of the above equation, we get

$$\left(\mathcal{I}_R \otimes \mathcal{T}_{CA}\right)(\rho_{RCA})$$

$$= (I_R \otimes T_{CA})\rho_{RCA}(I_R \otimes T_{AC}^\dagger) \tag{5.G.12}$$

$$= \left(I_R \otimes \sum_{j_1} T(\theta_{j_1})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A\right)\rho_{RCA}\left(I_R \otimes \sum_{j_2} T^\dagger(\theta_{j_2})_C \otimes |\psi_{j_2}\rangle\langle\psi_{j_2}|_A\right) \tag{5.G.13}$$

$$= \left(\sum_{j_1} I_R \otimes T(\theta_{j_1})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A\right)\rho_{RCA}\left(\sum_{j_2} I_R \otimes T^\dagger(\theta_{j_2})_C \otimes |\psi_{j_2}\rangle\langle\psi_{j_2}|_A\right). \tag{5.G.14}$$

This implies that the Kraus operator, which we denote by V_{RCA}, of the above channel is

$$V_{RCA} = \sum_{j_1} I_R \otimes T(\theta_{j_1})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A. \tag{5.G.15}$$

Therefore, this channel can be rewritten as

$$(\mathcal{I}_R \otimes \mathcal{T}_{CA})(\rho_{RCA}) = V_{RCA}\rho_{RCA}V_{RCA}^\dagger. \tag{5.G.16}$$

Similarly, we get the Kraus representation of the second channel in (5.G.11). For this, consider the following:

$$\mathcal{I}_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger \circ \widetilde{\mathcal{T}}_{CB} \circ Q_{BA} \circ \mathcal{A}_{CA}(\rho_{RCA})$$

$$= I_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger \circ \widetilde{\mathcal{T}}_{CB} \circ Q_{BA} \circ (\rho_{RCA} \otimes |0\rangle\langle 0|_B) \tag{5.G.17}$$

$$= I_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger \circ \widetilde{\mathcal{T}}_{CB}\left(Q_{BA}\,(\rho_{RCA} \otimes |0\rangle\langle 0|_B)\,Q_{AB}^\dagger\right) \tag{5.G.18}$$

$$= I_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger \circ \widetilde{\mathcal{T}}_{CB}\Bigg(\sum_{j_1,x_1,y_1} f_{j_1x_1y_1}|x_1\rangle\langle y_1|_B \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A$$
$$(\rho_{RCA} \otimes |0\rangle\langle 0|_B)\sum_{j_2,x_2,y_2} f^*_{j_2x_2y_2}|y_2\rangle\langle x_2|_B \otimes |\psi_{j_2}\rangle\langle\psi_{j_2}|_A\Bigg) \tag{5.G.19}$$

$$= I_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger \circ \widetilde{\mathcal{T}}_{CB}\Bigg(\sum_{j_1,j_2x_1,x_2} f_{j_1x_10}f^*_{j_2x_20}\langle\psi_{j_1}|\rho_{RCA}|\psi_{j_2}\rangle \otimes |x_1\rangle\langle x_2|_B \otimes |\psi_{j_1}\rangle\langle\psi_{j_2}|_A\Bigg) \tag{5.G.20}$$

$$= I_R \otimes \mathrm{Tr}_B \circ Q_{BA}^\dagger\Bigg(\sum_{j_1,j_2x_1,x_2} f_{j_1x_10}f^*_{j_2x_20}T(^{x_1}/_{2^p})_C\langle\psi_{j_1}|\rho_{RCA}|\psi_{j_2}\rangle T^\dagger(^{x_2}/_{2^p})_C \otimes |x_1\rangle\langle x_2|_B \otimes |\psi_{j_1}\rangle\langle\psi_{j_2}|_A\Bigg) \tag{5.G.21}$$

$$= I_R \otimes \mathrm{Tr}_B\Bigg(\sum_{j_1,j_2x_1,x_2,y_3,y_4} f_{j_1x_10}f^*_{j_2x_20}f^*_{j_1x_1y_3}f_{j_2x_2y_4}T(^{x_1}/_{2^p})_C\langle\psi_{j_1}|\rho_{RCA}|\psi_{j_2}\rangle T^\dagger(^{x_2}/_{2^p})_C \otimes |y_3\rangle\langle y_4|_B \otimes |\psi_{j_1}\rangle\langle\psi_{j_2}|_A\Bigg) \tag{5.G.22}$$

$$= I_R \otimes \left(\sum_{j_1,j_2 x_1,x_2,y_3} f_{j_1 x_1 0} f^*_{j_2 x_2 0} f^*_{j_1 x_1 y_3} f_{j_2 x_2 y_3} T^{(x_1/2^p)}_C \langle \psi_{j_1} | \rho_{RCA} | \psi_{j_2} \rangle T^{\dagger (x_2/2^p)}_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_2}|_A \right) \tag{5.G.23}$$

$$= \sum_{j_1,j_2} \left(\sum_{x_1,x_2,y_3} f_{j_1 x_1 0} f^*_{j_2 x_2 0} f^*_{j_1 x_1 y_3} f_{j_2 x_2 y_3} \left(I_R \otimes T^{(x_1/2^p)}_C\right) \langle \psi_{j_1} | \rho_{RCA} | \psi_{j_2} \rangle \left(I_R \otimes T^{\dagger (x_2/2^p)}_C\right) \otimes |\psi_{j_1}\rangle\langle\psi_{j_2}|_A \right) \tag{5.G.24}$$

$$= \sum_{y_3} \Bigg(\underbrace{\sum_{j_1,x_1} f_{j_1 x_1 0} f^*_{j_1 x_1 y_3} \left(I_R \otimes T^{(x_1/2^p)}_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A\right)}_{=:K_{y_3}} \Bigg) \rho_{RCA} \Bigg(\underbrace{\sum_{j_2,x_2} f^*_{j_2 x_2 0} f_{j_2 x_2 y_3} \left(I_R \otimes T^{\dagger (x_2/2^p)}_C \otimes |\psi_{j_2}\rangle\langle\psi_{j_2}|_A\right)}_{K^\dagger_{y_3}} \Bigg) \tag{5.G.25}$$

$$= \sum_{y_3} K_{y_3} \rho_{RCA} K^\dagger_{y_3}. \tag{5.G.26}$$

Plugging (5.G.16) and (5.G.26) into (5.G.11) and removing the system labels for simplicity, we get

$$\frac{1}{2} \left\| \mathcal{T}_{AC} - \mathrm{Tr}_B \circ \mathcal{Q}^\dagger_{BA} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA} \right\|_\diamond = \sup_\rho \frac{1}{2} \left\| V\rho V^\dagger - \sum_{y_3} K_{y_3} \rho K^\dagger_{y_3} \right\|_1. \tag{5.G.27}$$

In the perfect QPE case, uncomputation is exact, resulting in register B being perfectly reset to the all-zeros state after applying $Q^\dagger$ (see Figure 5.6). Consequently, in this scenario, there exists only a single Kraus operator (corresponding to $y_3 = 0$), and this Kraus operator is a unitary operator V. For the imperfect QPE case, we expect this operator to remain approximately unitary, as we will demonstrate next.

Continuing,

$$\frac{1}{2} \left\| \mathcal{T}_{AC} - \mathrm{Tr}_B \circ \mathcal{Q}^\dagger_{BA} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA} \right\|_\diamond$$

$$= \sup_\rho \frac{1}{2} \left\| V\rho V^\dagger - K_0 \rho K^\dagger_0 - \sum_{y_3 \neq 0} K_{y_3} \rho K^\dagger_{y_3} \right\|_1 \tag{5.G.28}$$

$$\leq \sup_\rho \frac{1}{2} \left(\left\| V\rho V^\dagger - K_0 \rho K^\dagger_0 \right\|_1 + \left\| \sum_{y_3 \neq 0} K_{y_3} \rho K^\dagger_{y_3} \right\|_1 \right) \tag{5.G.29}$$

$$= \sup_\rho \frac{1}{2} \left\| V\rho V^\dagger - K_0 \rho K^\dagger_0 \right\|_1 + \sup_\rho \frac{1}{2} \left\| \sum_{y_3 \neq 0} K_{y_3} \rho K^\dagger_{y_3} \right\|_1, \tag{5.G.30}$$

where the first inequality follows from the triangle inequality.

Let us begin by bounding the first term of the above equation:

$$\sup_{\rho} \left\| V\rho V^\dagger - K_0 \rho K_0^\dagger \right\|_1 = \sup_{\rho} \left\| V\rho V^\dagger - K_0\rho V^\dagger + K_0\rho V^\dagger - K_0\rho K_0^\dagger \right\|_1 \tag{5.G.31}$$

$$\leq \sup_{\rho} \left\| V\rho V^\dagger - K_0\rho V^\dagger \right\|_1 + \sup_{\rho} \left\| K_0\rho V^\dagger - K_0\rho K_0^\dagger \right\|_1 \tag{5.G.32}$$

$$= \sup_{\rho} \left\| (V - K_0)\rho V^\dagger \right\|_1 + \sup_{\rho} \left\| K_0\rho (V^\dagger - K_0^\dagger) \right\|_1 \tag{5.G.33}$$

$$\leq \sup_{\rho} \|V - K_0\| \|\rho\|_1 \left\| V^\dagger \right\| + \sup_{\rho} \|K_0\| \|\rho\|_1 \left\| V^\dagger - K_0^\dagger \right\| \tag{5.G.34}$$

$$= \|V - K_0\| + \|K_0\| \|V - K_0\| \tag{5.G.35}$$

$$= (1 + \|K_0\|) \|V - K_0\|, \tag{5.G.36}$$

where the first inequality follows from the triangle inequality and the second inequality follows from the Hölder's inequality. Next, we bound the quantities $\|K_0\|$ and $\|V - K_0\|$ from above one by one:

$$\|K_0\| = \left\| \sum_{j_1, x_1} f_{j_1 x_1 0} f^*_{j_1 x_1 0} \left(I_R \otimes T^{(x_1/2^p)}_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A \right) \right\| \tag{5.G.37}$$

$$= \max_{j_1} \left\| \sum_{x_1} |f_{j_1 x_1 0}|^2 \left(I_R \otimes T^{(x_1/2^p)}_C \right) \right\| \tag{5.G.38}$$

$$\leq \max_{j_1} \sum_{x_1} |f_{j_1 x_1 0}|^2 \left\| (I_R \otimes T^{(x_1/2^p)}_C) \right\| \tag{5.G.39}$$

$$\leq 1. \tag{5.G.40}$$

The second equality follows from Lemma 76, the first inequality follows from the triangle inequality, and the last inequality follows from the fact that for every x_1, we have $|f_{j_1 x_1 0}|^2 \leq 1$. We now bound $\|V - K_0\|$ from above:

$$\|V - K_0\|$$

$$= \left\| \sum_{j_1} I_R \otimes T(\theta_{j_1})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A - \sum_{j_1, x_1} f_{j_1 x_1 0} f^*_{j_1 x_1 0} \left(I_R \otimes T^{(x_1/2^p)}_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A \right) \right\| \tag{5.G.41}$$

$$\leq \left\| \sum_{j_1} T(\theta_{j_1})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A - \sum_{j_1,x_1} |f_{j_1 x_1 0}|^2 \left(T(^{x_1}/_{2^p})_C \otimes |\psi_{j_1}\rangle\langle\psi_{j_1}|_A \right) \right\| \tag{5.G.42}$$

$$= \max_{j_1} \left\| T(\theta_{j_1})_C - \sum_{x_1} |f_{j_1 x_1 0}|^2 T(^{x_1}/_{2^p})_C \right\| \tag{5.G.43}$$

$$= \max_{j_1} \left\| \sum_{x_1} |f_{j_1 x_1 0}|^2 T(\theta_{j_1})_C - \sum_{x_1} |f_{j_1 x_1 0}|^2 T(^{x_1}/_{2^p})_C \right\| \tag{5.G.44}$$

$$\leq \max_{j_1} \sum_{x_1} |f_{j_1 x_1 0}|^2 \left\| T(\theta_{j_1})_C - T(^{x_1}/_{2^p})_C \right\| \tag{5.G.45}$$

$$= \max_{j_1} \left(\sum_{x_1 : |\theta_{j_1} - x_1/2^p| \leq \delta} |f_{j_1 x_1 0}|^2 \left\| T(\theta_{j_1})_C - T(^{x_1}/_{2^p})_C \right\| + \sum_{x_1 : |\theta_{j_1} - x_1/2^p| > \delta} |f_{j_1 x_1 0}|^2 \left\| T(\theta_{j_1})_C - T(^{x_1}/_{2^p})_C \right\| \right) \tag{5.G.46}$$

$$\leq \max_{j_1} \left(\sum_{x_1 : |\theta_{j_1} - x_1/2^p| \leq \delta} |f_{j_1 x_1 0}|^2 (L_T \delta) + \sum_{x_1 : |\theta_{j_1} - x_1/2^p| > \delta} |f_{j_1 x_1 0}|^2 (2) \right) \tag{5.G.47}$$

$$\leq \max_{j_1} \left(L_T \delta + 2\varepsilon \right) \tag{5.G.48}$$

$$= L_T \delta + 2\varepsilon, \tag{5.G.49}$$

where the first inequality follows from the submultiplicativity of the spectral norm under tensor products, the second equality follows from Lemma 76, the second inequality follows from the triangle inequality, the third inequality follows from (5.G.2), and the last inequality follows from the following fact about the error probability of the QPE algorithm used:

$$\sum_{x_1 : |\theta_{j_1} - x_1/2^p| > \delta} |f_{j_1 x_1 0}|^2 \leq \varepsilon. \tag{5.G.50}$$

To this end, using the bounds given by (5.G.40) and (5.G.49) in (5.G.36), we get

$$\sup_{\rho} \left\| V \rho V^\dagger - K_0 \rho K_0^\dagger \right\|_1 \leq 2 L_T \delta + 4\varepsilon. \tag{5.G.51}$$

Now that we have bounded the first term of (5.G.30), we focus on bounding its second

term. For this, consider the following:

$$\left\| \sum_{y_3 \neq 0} K_{y_3}^\dagger K_{y_3} \right\| = \left\| K_0^\dagger K_0 - \sum_{y_3} K_{y_3}^\dagger K_{y_3} \right\| \tag{5.G.52}$$

$$= \left\| K_0^\dagger K_0 - I \right\| \tag{5.G.53}$$

$$= \left\| K_0^\dagger K_0 - V^\dagger V \right\| \tag{5.G.54}$$

$$= \left\| K_0^\dagger K_0 - K_0^\dagger V + K_0^\dagger V - V^\dagger V \right\| \tag{5.G.55}$$

$$\leq \left\| K_0^\dagger K_0 - K_0^\dagger V \right\| + \left\| K_0^\dagger V - V^\dagger V \right\| \tag{5.G.56}$$

$$\leq \left\| K_0^\dagger \right\| \left\| K_0 - V \right\| + \left\| K_0^\dagger - V^\dagger \right\| \left\| V \right\| \tag{5.G.57}$$

$$\leq 2 \left\| K_0 - V \right\| \tag{5.G.58}$$

$$\leq 2 L_T \delta + 4\varepsilon, \tag{5.G.59}$$

where the first inequality follows from the triangle inequality, the second inequality follows from the submultiplicativity property of the spectral norm, the third inequality follows from (5.G.40), and the last inequality follows from (5.G.49). Next, we have the following equality:

$$\sup_\rho \left\| \sum_{y_3 \neq 0} K_{y_3} \rho K_{y_3}^\dagger \right\|_1 = \left\| \sum_{y_3 \neq 0} K_{y_3}^\dagger K_{y_3} \right\|. \tag{5.G.60}$$

This follows due to the following fact, where each A_i is some linear operator:

$$\left\| \sum_i A_i^\dagger A_i \right\| = \sup_\rho \operatorname{Tr}\left[\sum_i A_i^\dagger A_i \rho \right] \tag{5.G.61}$$

$$= \sup_\rho \sum_i \operatorname{Tr}\left[A_i^\dagger A_i \rho \right] \tag{5.G.62}$$

$$= \sup_\rho \sum_i \operatorname{Tr}\left[A_i \rho A_i^\dagger \right] \tag{5.G.63}$$

$$= \sup_\rho \operatorname{Tr}\left[\sum_i A_i \rho A_i^\dagger \right] \tag{5.G.64}$$

$$= \sup_\rho \left\| \sum_i A_i \rho A_i^\dagger \right\|_1. \tag{5.G.65}$$

Using the identity given by (5.G.60), and the bound given in (5.G.59), we get

$$\sup_{\rho} \left\| \sum_{y_3 \neq 0} K_{y_3} \rho K_{y_3}^{\dagger} \right\|_1 \leq 2L_T \delta + 4\varepsilon. \tag{5.G.66}$$

Finally, combining the bounds of (5.G.51) and (5.G.66) to bound (5.G.30) from above, we get

$$\frac{1}{2} \left\| \mathcal{T}_{AC} - \mathrm{Tr}_B \circ \mathcal{Q}_{BA}^{\dagger} \circ \widetilde{\mathcal{T}}_{CB} \circ \mathcal{Q}_{BA} \circ \mathcal{A}_{CA} \right\|_{\diamond} \leq 2L_T \delta + 4\varepsilon. \tag{5.G.67}$$

This concludes the proof.

Eq. (5.G.10) quantifies the error resulting from the uncomputation of QPE combined with the error of approximating the phase in the first place. Let us compare this error to the error introduced solely due the approximation of the phase. For this purpose, we consider the unitary

$$\overline{cT} := \sum_j T(\bar{\theta}_j)_C \otimes |\psi_j\rangle\langle\psi_j|_A, \tag{5.G.68}$$

where $\bar{\theta}_j$ is a single estimate with $|\bar{\theta}_j - \theta_j| \leq \delta$. Let us denote the associated channel by $\overline{\mathcal{T}}$. To implement this unitary one would need access to a QPE algorithm that always outputs a single estimate, which is known not to exist (Here we are ignoring the failure probability, ε, which we assume is negligible compared to $L_T\delta$). Nevertheless, in such a case uncomputation would be exact. It is possible to show that

$$\frac{1}{2} \left\| \mathcal{T} - \overline{\mathcal{T}} \right\|_{\diamond} \leq L_T \delta. \tag{5.G.69}$$

This is due to the following:

$$\frac{1}{2} \left\| \mathcal{T} - \overline{\mathcal{T}} \right\|_{\diamond} \leq \left\| cT - \overline{cT} \right\| \tag{5.G.70}$$

$$= \left\| \sum_j T(\theta_j)_C \otimes |\psi_j\rangle\langle\psi_j|_A - \sum_j T(\bar{\theta}_j)_C \otimes |\psi_j\rangle\langle\psi_j|_A \right\| \tag{5.G.71}$$

$$= \left\| \sum_j \left(T(\theta_j)_C - T(\bar{\theta}_j)_C \right) \otimes |\psi_j\rangle\langle\psi_j|_A \right\| \tag{5.G.72}$$

$$= \max_j \left\| T(\theta_j)_C - T(\bar{\theta}_j)_C \right\| \tag{5.G.73}$$

$$\leq L_T \delta. \tag{5.G.74}$$

The first inequality follows from Lemma 77, the third equality follows from Lemma 76, and the last inequality follows from the L_T-Lipschitz property of the operator T. All in all, from (5.G.10) and (5.G.69), we can say that the additional error of QPE due to the uncomputing is the same order as the error due to the approximation of the phase. ■

5.H Useful Lemmas

Lemma 76 (Norm of direct-sum of operators) *Let $A = A_1 \bigoplus \cdots \bigoplus A_n$, where each A_i is a linear operator and n is some natural number. Then the following is true:*

$$\|A\| = \max_i \|A_i\| . \tag{5.H.1}$$

Proof. For each $i \in \{1, \ldots, n\}$, consider the following singular value decomposition: $A_i = U_i D_i V_i^\dagger$, where U_i and V_i are unitaries and D_i is a diagonal matrix consisting of singular values. Then the singular value decomposition of $A = UDV^\dagger$, where $U = U_1 \bigoplus \cdots \bigoplus U_n$, $D = D_1 \bigoplus \cdots \bigoplus D_n$, and $V = V_1 \bigoplus \cdots \bigoplus V_n$. Using this and the fact that the spectral norm of a matrix is equal to its maximum singular value, we get $\|A\| = \max_i \|A_i\|$. ■

Lemma 77 (Bound on diamond distance between unitary channels) *Let $\mathcal{U}(\cdot) := U(\cdot)U^\dagger$ and $\mathcal{V}(\cdot) := V(\cdot)V^\dagger$ be two unitary channels with unitaries U and V, respectively. Then the following holds:*

$$\frac{1}{2} \|\mathcal{U} - \mathcal{V}\|_\diamond \leq \|U - V\| . \tag{5.H.2}$$

Proof. The following is an alternate simple proof of Lemma 12.6 of [AKN98]. Consider the following:

$$\frac{1}{2}\left\|\mathcal{U}-\mathcal{V}\right\|_{\diamond}$$

$$=\sup_{\rho}\frac{1}{2}\left\|(\mathcal{I}\otimes\mathcal{U})(\rho)-(\mathcal{I}\otimes\mathcal{V})(\rho)\right\|_1 \tag{5.H.3}$$

$$=\sup_{\rho}\frac{1}{2}\left\|(I\otimes U)\rho(I\otimes U^\dagger)-(I\otimes V)\rho(I\otimes V^\dagger)\right\|_1 \tag{5.H.4}$$

$$=\sup_{\rho}\frac{1}{2}\left\|(I\otimes U)\rho(I\otimes U^\dagger)-(I\otimes U)\rho(I\otimes V^\dagger)+(I\otimes U)\rho(I\otimes V^\dagger)-(I\otimes V)\rho(I\otimes V^\dagger)\right\|_1 \tag{5.H.5}$$

$$\leq\sup_{\rho}\frac{1}{2}\left\|(I\otimes U)\rho(I\otimes U^\dagger)-(I\otimes U)\rho(I\otimes V^\dagger)\right\|_1+\frac{1}{2}\left\|(I\otimes U)\rho(I\otimes V^\dagger)-(I\otimes V)\rho(I\otimes V^\dagger)\right\|_1 \tag{5.H.6}$$

$$=\sup_{\rho}\frac{1}{2}\left\|(I\otimes U)\rho(I\otimes U^\dagger-I\otimes V^\dagger)\right\|_1+\frac{1}{2}\left\|(I\otimes U-I\otimes V)\rho(I\otimes V^\dagger)\right\|_1 \tag{5.H.7}$$

$$\leq\sup_{\rho}\frac{1}{2}\left\|(I\otimes U)\right\|\left\|\rho\right\|_1\left\|(I\otimes U^\dagger-I\otimes V^\dagger)\right\|+\frac{1}{2}\left\|(I\otimes U-I\otimes V)\right\|\left\|\rho\right\|_1\left\|(I\otimes V^\dagger)\right\| \tag{5.H.8}$$

$$=\frac{1}{2}\left\|(I\otimes U^\dagger-I\otimes V^\dagger)\right\|+\frac{1}{2}\left\|(I\otimes U-I\otimes V)\right\| \tag{5.H.9}$$

$$=\frac{1}{2}\left\|(I\otimes U-I\otimes V)\right\|+\frac{1}{2}\left\|(I\otimes U-I\otimes V)\right\| \tag{5.H.10}$$

$$\leq\frac{1}{2}\left\|U-V\right\|+\frac{1}{2}\left\|U-V\right\| \tag{5.H.11}$$

$$=\left\|U-V\right\|. \tag{5.H.12}$$

The first inequality follows from the triangle inequality, the second inequality follows from the generalized Hölder's inequality, and the third inequality holds due to the sub-multiplicativity property of the trace norm under tensor products. ■

CHAPTER 6

LIST OF MY PH.D. RESEARCH PROJECTS

During the course of my doctoral research, I have contributed to a variety of projects in quantum algorithms. These collaborations have led to several peer-reviewed publications as well as preprints, spanning both theoretical advances and applications. A complete list of the papers co-authored during my Ph.D. studies is given below:

- J. Chen, H. Westerheim, Z. Holmes, I. Luo, T. Nuradha, **D. Patel**, S. Rethinasamy, K. Wang, M. M. Wilde, "QSlack: A slack-variable approach for variational quantum semi-definite programming", in *Physical Review A*, Aug. 2025, doi:10.1103/lwxq-4myj. [CWH+25]

  We introduce QSlack and CSlack, variational methods for large-scale semi-definite and linear programming based on slack variables and the penalty approach. By making use of the SDP and LP duality theory, we prove that these methods provide a theoretical guarantee that, if one could find global optima of the objective functions, then the resulting values sandwich the true optimal values from both above and below. Benchmark results show that both primal and dual formulations approach ground-truth values with errors typically on the order of 10^{-2}.

- M. Minervini, **D. Patel**, M. M. Wilde, "Quantum natural gradient with thermal-state initialization", in *Physical Review A*, Aug. 2025, doi:10.1103/49b1-fbc3. [MPW25b]

  We derive exact expressions for three quantum-generalized Fisher information matrices (Fisher-Bures, Wigner-Yanase, Kubo-Mori) applicable to variational quantum circuits initialized in thermal states. These expressions can be estimated by combining Hadamard tests, classical sampling, and Hamiltonian simulation. The framework enables quantum natural gradient methods in the mixed-state regime and

yields fundamental limits on parameter estimation under sample access.

- N. Liu, M. Minervini, **D. Patel,** and M. M. Wilde, "Quantum thermodynamics and semi-definite optimization", *arXiv:2505.04514*, May 2025. [LMPW25]

  We show that minimizing free energy (rather than energy) in quantum systems with non-commuting conserved charges leads to a dual chemical-potential maximization that is concave and efficiently solvable by gradient ascent. This thermodynamic insight bridges to SDP solving. We present classical and hybrid quantum-classical algorithms with runtime and convergence guarantees.

- A. N. Sims, **D. Patel**, A. Philip, A. H. Rubin, R. Bandyopadhyay, M. Radulaski, M. M. Wilde, "Digital Quantum Simulations of the Non-Resonant Open Tavis-Cummings Model", *arXiv:2501.18522*, Jan. 2025. [SPP+25]

  The open Tavis–Cummings model describes N quantum emitters coupled to a common cavity mode with losses and decoherence, but becomes classically hard to simulate as N increases. We implement two quantum algorithms to capture its dynamics in the inhomogeneous, non-resonant regime with up to three excitations. We show that the implemented algorithms have gate complexities that scale polynomially, as $O(N^2)$ and $O(N^3)$, while the number of qubits used by these algorithms (space complexity) scales linearly as $O(N)$. One of these algorithms is the sampling-based wave matrix Lindbladization algorithm, for which we propose two protocols to implement its system-independent fixed interaction, resolving key open questions of [PW23b]. We benchmark our results against a classical differential equation solver in a variety of scenarios and demonstrate that our algorithms accurately reproduce the expected dynamics.

- M. Minervini, **D. Patel**, M. M. Wilde, "Evolved Quantum Boltzmann Machines", *arXiv:2501.03367*, Feb. 2025. [MPW25a]

We introduce evolved quantum Boltzmann machines, a variational ansatz built from preparing a thermal state of one Hamiltonian followed by unitary evolution under another. Our framework yields analytical gradients for tasks such as ground-state energy estimation and generative modeling, which can be evaluated with quantum algorithms involving classical sampling, Hamiltonian simulation, and the Hadamard test. We also derive Fisher-Bures, Wigner-Yanase, and Kubo-Mori information matrices for this ansatz and develop natural gradient descent algorithms based on them. Together, these results establish a powerful approach to quantum optimization and learning.

- B. Go, H. Kwon, S. Park, **D. Patel**, M. M. Wilde, "Sample-based Hamiltonian and Lindbladian simulation: Non-asymptotic analysis of sample complexity", *Quantum Science and Technology*, Sept. 2025, doi:10.1088/2058-9565/ae075b. [GKP+25]

  Density matrix exponentiation (DME) and wave matrix Lindbladization (WML) are sample-based quantum algorithms for simulating Hamiltonian and Lindbladian dynamics. We provide non-asymptotic sample complexity analyses, proving optimal bounds for both DME and WML up to constant factors. Specifically, we show $O(t^2/\epsilon)$ scaling for DME (with t the simulated evolution time and ϵ the error tolerance) and $O(t^2d^2/\epsilon)$ for WML, along with matching lower bounds in relevant regimes. These results establish the optimality of both algorithms and clarify gaps in earlier analyses.

- **D. Patel**, M. M. Wilde, "Natural gradient and parameter estimation for quantum Boltzmann machines", *arXiv:2410.24058*, Oct. 2024. [PW24]

  We derive formulas for the Fisher–Bures and Kubo–Mori information matrices of parameterized thermal states and develop quantum algorithms to estimate their elements using classical sampling, Hamiltonian simulation, and the Hadamard test.

These results enable a natural gradient descent algorithm for quantum Boltzmann machines that accounts for the geometry of thermal states. We also establish limits on Hamiltonian parameter estimation from thermal samples and sketch an asymptotically optimal strategy in the single-parameter case.

- **D. Patel**, D. Koch, S. Patel, M. M. Wilde, "Quantum Boltzmann machine learning of ground-state energies", *arXiv:2410.12935*, Oct. 2024. [PKPW24]

  We study quantum Boltzmann machines as a variational ansatz for estimating ground-state energies of Hamiltonians. Unlike standard parameterized circuits, this approach is not known to suffer from the barren-plateau problem. We delineate a hybrid quantum-classical algorithm for this task and rigorously prove that it converges to an ε-approximate stationary point of the energy function optimized over parameter space, while using a number of parameterized-thermal-state samples that is polynomial in ε^{-1}, the number of parameters, and the norm of the Hamiltonian being optimized. Our algorithm efficiently estimates gradients via a novel circuit combining classical sampling, Hamiltonian simulation, and the Hadamard test, resolving an open problem from [AAR+18]. Supporting results include explicit gradient and Hessian formulas and bounds used in the convergence proof.

- **D. Patel**, I. Chakrabarty, A. Roy, N. Ganguly, "Broadcasting of non-locality", *Pramana – Journal of Physics*, Apr. 2024, doi:10.1007/s12043-024-02726-1. [PRCG24]

  We study whether Bell non-locality and steering can be broadcast using local quantum cloning. Using the universal Buzek–Hillary cloner, we show that non-locality cannot be broadcasted through local cloning, analyzed via the Bell–CHSH and CJWR steering inequalities. For more than six measurement settings, some output states remain steerable, but Werner and Bell-diagonal states become unsteerable under optimal cloning. Extending to three-qubit systems, we also find that genuine

tripartite non-locality cannot be broadcast under Svetlichny's inequality.

- **D. Patel**, S. J. S. Tan, Y. Subasi, A. T. Sornborger, "Optimal Coherent Quantum Phase Estimation via Tapering", *arXiv:2403.18927*, Sept. 2024. [PTSS24]

  Chapter 5 is based on this paper.

- Z. Goldfeld, **D. Patel**, S. Sreekumar, M. M. Wilde, "Quantum neural estimation of entropies", in *Physical Review A*, Mar. 2024, doi:10.1103/PhysRevA.109.032431. [GPSW24]

  Entropy measures quantify the amount of information and correlation present in a quantum system. In practice, when the quantum state is unknown and only copies thereof are available, one must resort to the estimation of such entropy measures. Here we propose a variational quantum algorithm for estimating the von Neumann and Rényi entropies, as well as the measured relative entropy and measured Rényi relative entropy. Our approach first parametrizes a variational formula for the measure of interest by a quantum circuit and a classical neural network and then optimizes the resulting objective over parameter space. Numerical simulations of our quantum algorithm are provided, using a noiseless quantum simulator. The algorithm provides accurate estimates of the various entropy measures for the examples tested, which renders it a promising approach for usage in downstream tasks.

- H. Westerheim, J. Chen, Z. Holmes, I. Luo, T. Nuradha, **D. Patel**, S. Rethinasamy, K. Wang, M. M. Wilde, "Dual-VQE: A quantum algorithm to lower bound the ground-state energy", *arXiv:2312.03083*, Dec. 2023. [WCH+23]

  We introduce the dual variational quantum eigensolver (dual-VQE), a hybrid quantum-classical algorithm that produces a lower bound on the ground-state energy of a Hamiltonian, complementing the standard variational quantum eigensolver (VQE) upper bound. Together, VQE and dual-VQE sandwich the true

ground-state energy, providing a built-in quality check. Dual-VQE leverages semidefinite programming duality and we show that it can be implemented with resources comparable to VQE when combined with a convex combination ansatz and generative models. Our simulations on the transverse-field Ising model show convergence to the true value with error on the order of 10^{-2}.

- **D. Patel**, M. M. Wilde, "Wave Matrix Lindbladization I: Quantum Programs for Simulating Markovian Dynamics", *Open Systems & Information Dynamics*, Jun. 2023, doi:10.1142/S1230161223500105. [PW23a]

  Chapter 1 is based on this paper.

- **D. Patel**, M. M. Wilde, "Wave Matrix Lindbladization II: General Lindbladians, Linear Combinations, and Polynomials", *Open Systems & Information Dynamics*, Sept. 2023, doi:10.1142/S1230161223500142. [PW23b]

  Chapter 2 is based on this paper.

- **D. Patel**, P. J. Coles, M. M. Wilde, "Variational Quantum Algorithms for Semidefinite Programming", *Quantum*, Jun. 2024, doi:10.22331/q-2024-06-17-1374. [PCW24]

  Chapter 4 is based on this paper.

- A. Challapalli, J. Konlan, **D. Patel**, G. Li, "Discovery of Cellular Unit Cells With High Natural Frequency and Energy Absorption Capabilities by an Inverse Machine Learning Framework", *Frontiers in Mechanical Engineering*, Nov. 2021, doi:10.3389/fmech.2021.779098. [CKPL21]

  We develop an inverse machine learning framework combining regression and generative neural networks to design cellular unit cells with enhanced dynamic properties. The discovered structures achieve 30–100% higher natural frequency and up to 300% greater energy absorption compared to biomimetic designs such as honey-

comb, plant stems, and trabecular bone. Our experimental and simulation results validate these gains, demonstrating the potential of machine learning for engineering lightweight structures with improved vibration resistance and impact performance.

- A. Challapalli, **D. Patel**, G. Li, "Inverse machine learning framework for optimizing lightweight metamaterials", *Materials & Design*, Oct. 2021, doi:10.1016/j.matdes.2021.109937. [CPL21]

  Structure scouting and design optimization for superior mechanical performance through inverse machine learning is an emerging area of interest. Inverse machine learning can be a substantial approach in structural design to explore complex and massive numbers of geometrical patterns within short periods of time. Here, an inverse design framework using generative adversarial networks (GANs) is proposed to explore and optimize structural designs such as lightweight lattice unit cells. Lightweight lattice structures are widely accepted to have excellent mechanical properties and have found applications in various engineering structures. Using the proposed framework, different lattice unit cells that are 40–120% better in load carrying capacity than octet unit cell are discovered. These new lattice unit cells are analyzed numerically and validated experimentally by testing 3D printed lattice unit cells and lattice cored sandwiches. The proposed inverse design framework can be applied to the design and optimization of other types of load bearing structures.

- **D. Patel**, S. Patro, C. Vanarasa, I. Chakrabarty, A. K. Pati, "Impossibility of Cloning of Quantum Coherence", in *Physical Review A*, Feb. 2021, doi:10.1103/PhysRevA.103.022422. [PPV+21]

  It is well known that it is impossible to clone an arbitrary quantum state. How-

ever, this inability does not lead directly to no cloning of quantum coherence. Here, in this article, we show that it is impossible to clone the coherence of an arbitrary quantum state. In particular, with an ancillary system as machine state, we show that it is impossible to clone the coherence of states whose coherence is greater than the coherence of the known states on which the transformations are defined. Also, we characterize the class of states for which coherence cloning will be possible for a given choice of machine. Furthermore, we find the maximum range of states whose coherence can be cloned perfectly. The impossibility proof also holds when we do not include machine states. Lastly, we generalize the impossibility of cloning of coherence in terms of dimension of the quantum state and coherence measure taken into consideration.

- A. Ganguly, **D. Patel**, R. Shah, S. V. Thankachan, "LF Successor: Compact Space Indexing for Order-Isomorphic Pattern Matching", in *International Colloquium on Automata, Languages, and Programming (ICALP)*, Jul. 2021, doi:10.4230/LIPIcs.ICALP.2021.71. [GPST21]

  Two strings are order isomorphic iff the relative ordering of their characters is the same at all positions. For a given text $T[1, n]$ over an ordered alphabet of size σ, we can maintain an order-isomorphic suffix tree/array in $O(n \log n)$ bits and support (order-isomorphic) pattern/substring matching queries efficiently. It is interesting to know if we can encode these structures in space close to the text's size of $n \log \sigma$ bits. We answer this question positively by presenting an $O(n \log \sigma)$-bit index that allows access to any entry in order-isomorphic suffix array (and its inverse array) in $t_{\mathsf{SA}} = O(\log^2 n / \log \sigma)$ time. For any pattern P given as a query, this index can count the number of substrings of T that are order-isomorphic to P (denoted by occ) in $O((|P| \log \sigma + t_{\mathsf{SA}}) \log n)$ time using standard techniques. Also, it can report the

locations of those substrings in additional $O(\text{occ} \cdot t_{\mathsf{SA}})$ time.

- **D. Patel**, R. Shah, "Inverse Suffix Array Queries for 2-Dimensional Pattern Matching in Near-Compact Space", in *International Colloquium on Automata, Languages, and Programming (ICALP)*, Nov. 2021, doi:10.4230/LIPIcs.ISAAC.2021.60. [PS21]

  We study 2D pattern matching, where a square pattern is searched in a text arranged as a matrix. Classical 2D suffix trees and arrays exist for this task but require $O(N \log N)$ bits of space, which is not optimal. We present the first method to support inverse suffix array queries for 2D pattern matching in near-compact space, using two 1D compressed suffix trees. Our index uses $O(N \log\log N + N \log \sigma)$ bits and answers queries in polylogarithmic time, providing a step toward a full 2D succinct index.

BIBLIOGRAPHY

[AAR+18] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum Boltzmann machine. *Physical Review X*, 8:021050, May 2018. doi:10.1103/PhysRevX.8.021050.

[CKPL21] Adithya Challapalli, John Konlan, Dhrumil Patel, and Guoqiang Li. Discovery of cellular unit cells with high natural frequency and energy absorption capabilities by an inverse machine learning framework. *Frontiers in Mechanical Engineering*, 7, November 2021. doi:10.3389/fmech.2021.779098.

[CPL21] Adithya Challapalli, Dhrumil Patel, and Gouqiang Li. Inverse machine learning framework for optimizing lightweight metamaterials. *Materials & Design*, 208:109937, October 2021. doi:10.1016/j.matdes.2021.109937.

[CWH+25] Jingxuan Chen, Hanna Westerheim, Zoë Holmes, Ivy Luo, Theshani Nuradha, Dhrumil Patel, Soorya Rethinasamy, Kathie Wang, and Mark M. Wilde. Slack-variable approach for variational quantum semidefinite programming. *Physical Review A*, 112(2), August 2025. doi:10.1103/lwxq-4myj.

[GKP+25] Byeongseon Go, Hyukjoon Kwon, Siheon Park, Dhrumil Patel, and Mark M Wilde. Sample-based hamiltonian and lindbladian simulation: Non-asymptotic analysis of sample complexity. *Quantum Science and Technology*, September 2025. doi:10.1088/2058-9565/ae075b.

[GPST21] Arnab Ganguly, Dhrumil Patel, Rahul Shah, and Sharma V. Thankachan. Lf successor: Compact space indexing for order-isomorphic pattern matching. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, July 2021. doi:10.4230/LIPICS.ICALP.2021.71.

[GPSW24] Ziv Goldfeld, Dhrumil Patel, Sreejith Sreekumar, and Mark M. Wilde. Quantum neural estimation of entropies. *Physical Review A*, 109(3):032431, March 2024. doi:10.1103/physreva.109.032431.

[LMPW25] Nana Liu, Michele Minervini, Dhrumil Patel, and Mark M. Wilde. Quantum thermodynamics and semi-definite optimization, May 2025. doi:10.48550/ARXIV.2505.04514.

[MPW25a] Michele Minervini, Dhrumil Patel, and Mark M. Wilde. Evolved quantum boltzmann machines, February 2025. doi:10.48550/ARXIV.2501.03367.

[MPW25b] Michele Minervini, Dhrumil Patel, and Mark M. Wilde. Quantum natural gradient with thermal-state initialization. *Physical Review A*, 112(2), August 2025. doi:10.1103/49b1-fbc3.

[PCW24] Dhrumil Patel, Patrick J. Coles, and Mark M. Wilde. Variational quantum algorithms for semidefinite programming. *Quantum*, 8:1374, June 2024. doi:10.22331/q-2024-06-17-1374.

[PKPW24] Dhrumil Patel, Daniel Koch, Saahil Patel, and Mark M. Wilde. Quantum boltzmann machine learning of ground-state energies, October 2024. doi:10.48550/ARXIV.2410.12935.

[PPV+21] Dhrumil Patel, Subhasree Patro, Chiranjeevi Vanarasa, Indranil Chakrabarty, and Arun Kumar Pati. Impossibility of cloning of quantum coherence. *Physical Review A*, 103(2):022422, February 2021. doi:10.1103/physreva.103.022422.

[PRCG24] Dhrumil Patel, Arup Roy, Indranil Chakrabarty, and Nirman Ganguly. Broadcasting of non-locality. *Pramana*, 98(2), April 2024. doi:10.1007/s12043-024-02726-1.

[PS21] Dhrumil Patel and Rahul Shah. Inverse Suffix Array Queries for 2-Dimensional Pattern Matching in Near-Compact Space. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, volume 212 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:14, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ISAAC.2021.60.

[PTSS24] Dhrumil Patel, Shi Jie Samuel Tan, Yigit Subasi, and Andrew T. Sornborger. Optimal coherent quantum phase estimation via tapering, 2024. arXiv:2403.18927.

[PW23a] Dhrumil Patel and Mark M. Wilde. Wave matrix Lindbladization I: Quantum programs for simulating Markovian dynamics. *Open Systems and Information Dynamics*, 30(02):2350010, June 2023. doi:10.1142/s1230161223500105.

[PW23b] Dhrumil Patel and Mark M. Wilde. Wave matrix lindbladization ii: General lindbladians, linear combinations, and polynomials. *Open Systems and Information Dynamics*, 30(03):2350014, September 2023. doi:10.1142/s1230161223500142.

[PW24] Dhrumil Patel and Mark M. Wilde. Natural gradient and parameter estimation for quantum boltzmann machines, October 2024. doi:10.48550/ARXIV.2410.24058.

[SPP+25] Aidan N. Sims, Dhrumil Patel, Aby Philip, Alex H. Rubin, Rahul Bandyopadhyay, Marina Radulaski, and Mark M. Wilde. Digital quantum simulations of the non-resonant open tavis-cummings model, January 2025. doi:10.48550/ARXIV.2501.18522.

[WCH+23] Hanna Westerheim, Jingxuan Chen, Zoë Holmes, Ivy Luo, Theshani Nuradha, Dhrumil Patel, Soorya Rethinasamy, Kathie Wang, and Mark M. Wilde. Dual-vqe: A quantum algorithm to lower bound the ground-state energy, 2023. doi:10.48550/ARXIV.2312.03083.