C02029: Doctor of Philosophy

33847 PhD Thesis: Software Engineering

August 2023

Methods for Improved Analysis and Implementation of Algorithms for Local Hamiltonians

Mauro E.S. Morales

Under the supervision of Michael Bremner and Marika Kieferova.

School of Computer Science

Faculty of Engineering and Information Technology

University of Technology Sydney

NSW - 2007, Australia

Certificate of Original Authorship

I, Mauricio Enrique Morales Soler, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and IT at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Signature of Author

Date: August 2023

Abstract

Methods for Improved Analysis and Implementation of Algorithms for Local Hamiltonians

by

Mauro E.S. Morales

Doctor of Philosophy

University of Technology Sydney

One of the main applications of quantum computers is expected to be computing properties of physical systems. Of particular interest is that of computing ground states and of performing evolution of a system with a given Hamiltonian. Related to these problems are the questions of when can we expect a quantum advantage and how to best implement a quantum algorithm when such advantage is expected. In this thesis we seek to give partial answers to these questions for certain versions of the local Hamiltonian problem and the Hamiltonian evolution problem.

In the first part of this thesis we study a parameterized version of the local Hamiltonian problem where the ground-state of interest can be expressed as a superposition of computational basis states with a fixed Hamming weight. We find that this problem is unlikely to be tractable for quantum computers and moreover find connections with a quantum version of the exponential time hypothesis.

The second part of this thesis provides a quantum sampling scheme based on Fermions which is inspired on the Boson Sampling scheme. The hardness guarantees in our scheme are comparable to those of Random Circuit Sampling, moreover we prove an anticoncentration property for this scheme in an improvement over what is known for Boson Sampling.

In the final part we present results on product formulas for Hamiltonian simulation. This work improves over previous methods to implement higher-order product formulae,

by finding new product formulae which greatly reduce the error and thus require fewer gate counts to implement. Moreover, we compare the performance of these product formulae in practice and find that our product formulae could be a better choice for quantum chemistry.

To my parents for their unending support...

List of Research Outputs

Related to the Thesis :

(1) Michael J. Bremner, Zhengfeng Ji, Ryan L. Mann, Luke Mathieson, Mauro E. S. Morales, and Alexis T. E. Shaw. Quantum parameterized complexity, 2022. [BJM+22]

(2) Michael J. Bremner, Zhengfeng Ji, Xingjian Li, Luke Mathieson, and Mauro E. S. Morales. Parameterized complexity of weighted local hamil- tonian problems and the quantum exponential time hypothesis, 2022. [BJL+22]

(3) Michał Oszmaniec, Ninnat Dangniam, Mauro E.S. Morales, and Zoltán Zimborás. Fermion sampling: A robust quantum computational advantage scheme using fermionic linear optics and magic input states. PRX Quantum, 3:020328, May 2022. [ODMZ22]

(4) Mauro E. S. Morales, Pedro C. S. Costa, Daniel K. Burgarth, Yuval R. Sanders, and Dominic W. Berry. Greatly improved higher-order product formulae for quantum simulation, 2022. [MCB+22]

Others :

(1) Mauro E. S. Morales, Timur Tlyachev, and Jacob Biamonte. Variational learning of Grover's quantum search algorithm. Phys. Rev. A, 98:062333, Dec 2018. [MTB18]

(2) Jacob D. Biamonte, Mauro E. S. Morales, and Dax Enshan Koh. Entanglement scaling in quantum advantage benchmarks. Phys. Rev. A, 101:012349, Jan 2020. [BMK20]

(3) M. E. S. Morales, J. D. Biamonte, and Z. Zimborás. On the universality of the quantum approximate optimization algorithm. Quantum Information Processing, 19(9):291, Aug 2020. [MBZ20]

(4) V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability deficits in quantum approximate optimization. Phys. Rev. Lett., 124:090504, Mar 2020. [APMB20]

Acknowledgements

I would like to thank my primary supervisor, Michael Bremner, for the constant support and encouragement. Whenever I felt too lost in my research he always had good advice which would redirect my efforts in the right direction. I would also like to thank my SQA co-supervisor, Dominic Berry, for the opportunity to work in his group and his great help in research. Also would like to thank my co-supervisor, Marika Kieferova, for her advice during my PhD and her great feedback during the writing of this thesis. I must also acknowledge the support and fundign from the Sydney Quantum Academy.

A special thanks to all my collaborators during my PhD. First, to Zoltan Zimboras who not only was a great collaborator but also a true mentor from whom I learned much about research. Also a big thanks to my long list of collaborators which include Ninnat Dangniam, Michal Oszmaniec, Pedro Costa, Yuval Sanders, Daniel Burgarth, Ryan Mann, Zhengfeng Ji, Luke Mathieson. Having the opportunity of collaborating with all of them was a great honor. I would also like to thank collaborators from past works such as Dax Koh who also helped with feedback for this thesis.

I want to thank my long list of friends, who one way or another have always been present in my life and supported, without them this thesis would have been impossible to finish.

Last, but not least, my heartfelt thanks to my family. It is simply impossible to express in a few lines how much I appreciate everything they have done for me. To my father for always encouraging to pursue my passion and for my mother for always reminding me to never give up.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum computing has emerged as a groundbreaking field, driven by the anticipation of quantum algorithms surpassing classical algorithms in computational power. This motivation raises pivotal questions concerning the potential of quantum computers: (i) What are the limitations on their computational capabilities? (ii) Under which conditions can quantum algorithms exhibit superiority over classical counterparts? (iii) How can we effectively implement quantum algorithms to leverage their advantages? Extensive research has been dedicated to addressing these inquiries, particularly through investigations of problems with connections to physics. In fact, the initial impetus behind proposing quantum computing was rooted in the recognition of the classical intractability to simulate and compute certain properties of quantum systems [Fey86]. Subsequent work established the concept of quantum computation, employing the framework of a quantum Turing machine [Deu85], and fostering the development of quantum complexity theory [BV97].

Within the realm of quantum algorithms, two prominent problems have dominated the study of quantum physics: the local Hamiltonian problem and the Hamiltonian simulation problem. The local Hamiltonian problem involves Hamiltonians with constituent terms that interact solely over a constant number of qubits. Local Hamiltonians are physically relevant as many physical systems in condensed matter physics or quantum

chemistry include local interactions. When studying these physical systems one is often interested in the groundstate and the corresponding energy of the system, as many other properties depend on these. In its general form, this problem has been found to be challenging for quantum computers [KSV02]. More precisely, the local Hamiltonian problem has been shown to likely be intractable for polynomial-time quantum algorithms. This has spurred a series of works studying variations of the local Hamiltonian problem where the interactions are constrained or other parameters are included in the description of the problem (see Section 2.3.2 for more detail). Understanding the complexity of the local Hamiltonian problem is also useful for finding instances where a quantum advantage with respect to classical algorithms might exist, providing a pathway for understanding the utility of quantum computers for problems of physical relevance.

The Hamiltonian simulation problem revolves around the task of evolving a quantum state for a specified duration under a given Hamiltonian. This is a natural task when one is interested in simulating the behaviour of quantum systems in time starting from some initial state. Notably, this problem is considered tractable for quantum computers when dealing with local Hamiltonians. This problem was one of the first ones where a quantum advantage could be expected as it is not believed that there are efficient classical algorithms solving this problem [Fey82]. This problem could find many applications such as in quantum chemistry or condensed matter systems where one would be interested in studying the evolution of a system. Several methods to solve this problem have been proposed, culminating in the development of methods which are optimal with regard to the dependence on some of the parameters in this problem (we briefly review some of these methods in Section 2.4). One of the first methods proposed to solve this problem was that of product formulae [Llo96]. This simple method decomposes the total evolution under a Hamiltonian into separate evolutions of non-commuting parts. Although the complexity of this method is worse in terms of the dependence on the error than more recent algorithms, the implementation is simple enough that it is expected that product formulae will still be useful in early fault-tolerant quantum computers.

Given their potential applications and well-defined formulations, the local Hamiltonian and Hamiltonian simulation problems offer an intriguing context for addressing questions (i), (ii) and (iii) at the beginning of the introduction. In the case of the local Hamiltonian problem, we know that it is intractable for quantum computers when considering 2-local interactions. More precisely, it is complete for the class QMA which is a quantum generalization of the complexity class NP. By considering more restricted classes of Hamiltonians or by considering other ingredients in the instance, it is possible for the problem to become BQP-complete. In fact some recent work shows that this is the case [GL21]. On the other hand, improving on implementations for Hamiltonian simulation algorithms will help in making problems where quantum computers may be useful more readily available.

The work in this thesis is divided in three parts and has as overall theme a focus on giving partial answers to questions (i), (ii) and (iii). Specifically, we will study the likely intractability of a version of the local Hamiltonian where only the eigenstates that are superpositions of fixed Hamming-weight computational basis states are considered. We denote this version of the problem as the Weighted Local Hamiltonian problem. This work is in line with previous research that seeks to determine what kind of ingredients make the local Hamiltonian problem tractable or intractable (for some history on this, see Section 2.3.2). Then the Fermion sampling scheme is introduced. it is shown that a sampling scheme similar to Boson Sampling [AA11] where the inputs are given instead by Fermionic states achieves similar hardness guarantees than Random Circuit Sampling [AAB+19] or Boson Sampling. Finally, improved implementations of product formulae are given with applications for Hamiltonian simulation and quantum chemistry.

The first part of this work is concerned with studying the complexity of the Local Hamiltonian problem. This problem is central in the study of quantum complexity, as it was the first example of a QMA-complete problem and also is physically motivated by the usual problem in physics of finding the ground state energy of a physical system. I present work done with my collaborators Michael Bremner, Zhengfeng Ji, Xingjian Li, Ryan Mann, Luke Mathieson and Alexis Shaw which studies the complexity of

the Local Hamiltonian problem under the parameterized complexity framework by considering the number of excitations in the ground state as a parameter. In [BJM+22], we introduce the main complexity classes in quantum parameterized complexity. This work left several open questions, which include the complexity of the weighted Local Hamiltonian problem and the existence of natural complete problems for the intractable class QW[1]. Part of this work is presented in Chapter 3. In [BJL+22] we tried to answer the two previous questions. We were able to prove the unlikely tractability of the Local Hamiltonian problem by making connections to the Exponential Time Hypothesis (ETH). Moreover, we prove that the parameterized Local Hamiltonian problem is in QW[1]. The question of whether the parameterized Local Hamiltonian problem is complete for QW[1] is still open. This last work is presented in Chapter 4.

In the second part, I present work done in collaboration with Michal Oszmaniec, Ninnat Dangniam and Zoltan Zimboras. We introduce a new sampling problem with quantum advantage based on Fermionic statistics [ODMZ22], this work is presented in Chapter 5. Many quantum advantage proposals have been based on sampling schemes where random quantum circuits generate output probabilities which are hard to sample from when having only access to classical machines. One of the first sampling schemes proposed was that of Aaronson and Arkhipov [AA11], this scheme known as Boson sampling involves the use of random linear optics circuits whose output probabilities are represented as permanents of submatrices of a unitary. The authors show that it is unlikely for classical machines to sample from distributions close to the ones generated by the scheme provided some conjectures are assumed. Recently, Google has implemented a sampling scheme based on Random Circuit Sampling [AAB+19]. In our work, we show that it is possible to implement a similar scheme based on Fermionic statistics. We show that our scheme in fact combines advantages from both Boson sampling and Random Circuit Sampling.

The third part of this thesis involves the improvement of product formulae for Hamiltonian simulation. This is work done with Pedro Costa, Daniel Burgarth, Yuval Sanders and Dominic Berry. One of the main applications for quantum computers is expected

to be that of simulating other physical systems. The Hamiltonian simulation problem involves evolving for some time a given quantum state under a known Hamiltonian, which seems like a natural setting where to expect a quantum advantage with respect to classical computing. One of the first methods to implement such evolutions is that of Lie-Trotter product formulae [Llo96]. In our work [MCB+22] we find new product formulae based on techniques from [Yos90] which allow for more efficient implementations with a lower error. We make comparisons among state-of-art product formulae and study which ones are better suited for quantum chemistry. We show that the new product formulae we find can be more useful for quantum chemistry when compared with previous ones, this is shown by extensive numerics comparing the performance of the product formulae with randomly chosen Hamiltonians. Although there has been recent work achieving optimal dependence on several parameters for algorithms solving the Hamiltonian simulation problem [LC17a], product formulae have remained unexplored and in fact numerics show that the errors achieved through this method are lower than what theory suggests [CST+21].

1.1 Authorship

As stated previously, each part in this thesis includes work with several collaborators. At the beginning of each chapter (except Chapter 2) I outline in what way I was involved in each work. The work included in these chapters has been rewritten except when stated. In the following paragraphs I summarize my work on each chapter.

This chapter and Chapter 2 were completely written by myself except when citing certain definitions or results.

Chapter 3 is based on work in [BJM+22]. This was work done in collaboration with Michael Bremner, Zhengfeng Ji, Ryan Mann, Luke Mathieson and Alexis Shaw. My contribution in [BJM+22] was mainly that of helping in writing Section II and Section III of that paper. I contributed on the discussions for the proper definitions of the QW-hierarchy and in writing the proofs for some facts pertaining FPQT. In this

chapter I have taken most definitions directly from that paper and have been included for completeness. In Section 3.3.1 I include some non-published work regarding a parameterized version of the subset problem for which I am the main contributor.

Chapter 4 is based on work in [BJL+22]. This is work done in collaboration with Michael Bremner, Zhengfeng Ji, Xingjian Li and Luke Mathieson. My contribution in this project consisted in contributing ideas, writing and in the formation of proofs. I contributed to most of the writing which comprises Section 4.3.1, except Lemma 4.12. I also contributed in the writing of the rest of sections of this chapter, where I didn't contribute, I have rewritten those parts or added some detail such as in Section 4.3.3 (on error reduction part) and some parts on the clock construction in Section 4.3.4. Section 4.3.5 was originally written by my coauthors in the paper, I mainly participated of the discussions for this section. I have included Section 4.3.5 in my thesis for completeness. The figures included in this chapter are those of [BJL+22], the original versions were designed by me (except Figs. 4.4 and 4.5) and were later implemented in Tikz by my coauthors.

Chapter 5 is based on published work [ODMZ22] written in collaboration with Michal Oszmaniec, Ninnat Dangniam and Zoltan Zimboras. We introduce a new sampling scheme for quantum advantage based on Fermionic linear optics supplied with magic states which allow to prove hardness guarantees comparable to other schemes. My contribution to this work consisted in the contribution of ideas and writing several parts in [ODMZ22]. In this chapter I have rewritten or expanded parts in [ODMZ22] which have been included here, moreover, several parts where I did not contribute have not been included but I cite the corresponding results when used. In the paper [ODMZ22], I helped writing sections of Appendix E.3 which correspond to Appendix A.2 in this thesis, I also was involved in proofs of earlier versions of this part, but the final version includes many parts written by my coauthors and thus has been included as an appendix. The part corresponding to Sections 5.6 and 5.7 was written by my coauthors with some contributions by myself, but the version presented in this thesis is rewritten and expanded by me, in fact I obtain different error bounds than in our paper as I have followed a

different path in certain calculations. Although these error bounds are not improved, these changes help making the calculations clearer. In the paper I contributed to an earlier version of the proof of Theorem 5.9 and in this thesis I have detailed some of the calculations that are present in the paper. The figures in Appendix A were done by my coauthors but all the figures in this chapter (excluding the appendix) were done by me. The theorem, lemma and corollary statements that appear in this thesis are directly taken from [ODMZ22] unless otherwise specified, but as mentioned previously the explanations and discussions have been expanded by myself when appropriate. Finally, I contributed with some numerics which have not been included in this thesis but have been presented in [ODMZ22].

Chapter 6 is based on work in [MCB+22]. This is work done in collaboration with Pedro C.S. Costa, Daniel K. Burgarth, Yuval R. Sanders and Dominic W. Berry. In this work I contributed to the idea, writing and numerics. All the theorems, lemmas and corollaries included are stated as in [MCB+22] and also the figures. Not much is changed from the original publication as much of the writing was done by myself with corrections from my coauthors. Those parts written by coauthors have not been included in this chapter.

Chapter 2

Background

In this chapter I lay out background information relevant to the work presented in later parts of this thesis. Section 2.1 gives a brief introduction to quantum mechanics, Section 2.2 contains basic definitions used in theoretical computer science for classical and quantum computation and Section 2.4 gives a small summary on key methods for Hamiltonian simulation.

2.1 Quantum mechanics

The theory of quantum mechanics is one of the fundamental pillars of modern physics. It describes physical phenomena at the atomic and subatomic scale by relying heavily on the branch of mathematics known as linear algebra (we will assume knowledge of it in the following presentation). In this section we give the basics of this theory which are involved in the theory of quantum computing. In particular, we only work with finite-dimensional systems. A more in-depth introduction can be found in [NC00] or [KSV02] and a more mathematical introduction (albeit not focused on quantum computing) can be found in [Hal13].

The specification of quantum mechanical systems involves four parts: the description of the states of a system, the description of how a system evolves, how different systems

are combined and finally the description of a measurement in a system. In the next paragraphs we detail each of these parts. The state of a quantum system is described by a unit-normed vector in a d-dimensional Hilbert space with underlying vector spaces $\mathcal{H} = \mathbb{C}^d$. Such vectors are termed qudits. The norm in $\mathcal{H}$ is defined as the Euclidean norm. More precisely, given a vector $|\psi\rangle \in \mathcal{H}$ which can be decomposed into an orthonormal basis $\{|i\rangle\}_{i=1}^d$ with complex coefficients $\{\alpha_i\}_i$ as $|\psi\rangle = \sum_{i=1}^d \alpha_i |i\rangle$, then $\||\psi\rangle\| = \sqrt{\sum_{i=1}^d |\alpha_i|^2}$. In particular a qubit corresponds to a unit-normed vector $|\psi\rangle \in \mathbb{C}^2$ and we fix the standard basis which we denote $|0\rangle$ and $|1\rangle$. We denote by $\langle\psi|$ the Hermitian conjugate of $|\psi\rangle$ and by $\langle\psi|\psi\rangle$ the inner product of the state with itself. The norm of a vector $|\psi\rangle \in \mathcal{H}$ is given by $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$. Moreover, we define $|\psi\rangle\langle\phi| = |\psi\rangle \otimes \langle\phi|$ as the outer product of two vectors. Note that $\Pi = |\psi\rangle\langle\psi|$ is a projector onto the subspace spanned by $|\psi\rangle$.

The evolution is given by the action of a unitary operator U acting on $\mathcal{H}$. Since we are dealing with finite-dimensional systems, any unitary U can be written in terms of a Hermitian operator (a Hamiltonian) H such that $U = e^{-iH}$. We denote $U^\dagger$ as the Hermitian conjugate of U.

Suppose now we are given two quantum systems with Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$ in states $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$. To consider the state of the whole system combining systems 1 and 2, then we need to only consider the tensor product space $\mathcal{H}_1 \otimes \mathcal{H}_2$ with state $|\psi_1\rangle \otimes |\psi_2\rangle$ which sometimes we only write as $|\psi_1\rangle\,|\psi_2\rangle$ or $|\psi_1\psi_2\rangle$.

Finally, if a system is in a state $|\psi\rangle = \sum_{i=1}^d \alpha_i |a_i\rangle$ where $|a_i\rangle$ is some orthogonal basis of $\mathcal{H}$ and $\alpha_i \in \mathbb{C}$, then when measuring the system in this basis we get result $|a_i\rangle$ with probability $|\alpha_i|^2$.

It is useful to define probabilistic mixtures of quantum states. To describe these states we define the density operator or density matrix. If $\{p_j\}_{j=1}^k$ is a probability distribution over states $\{|\psi_j\rangle\}_{j=1}^k$ then the density operator is given by $\rho = \sum_{j=1}^k p_j |\psi_j\rangle\langle\psi_j|$. More generally, a density operator ρ is a linear operator over $\mathcal{H}$ satisfying the following properties: (i) $\mathrm{Tr}\{\rho\} = 1$, (ii) $\rho^\dagger = \rho$, (iii) $\langle\psi|\rho|\psi\rangle \geq 0$ for all states $|\psi\rangle$, i.e. ρ is positive semi-definite. We write this as $\rho \geq 0$. A unitary U will act on ρ as $U\rho U^\dagger$, but

we can define more general operations over density matrices. A quantum channel $\mathcal{E}$ is an operator on density matrices (or linear operators over $\mathcal{H}$ more generally), which fulfills the following properties: (i) linearity, i.e., $\mathcal{E}(\alpha\rho_1 + \beta\rho_2) = \alpha\mathcal{E}(\rho_1) + \beta\mathcal{E}(\rho_2)$ for all $\alpha, \beta \in \mathbb{C}$ and all density matrices ρ_1 and ρ_2. (ii) Complete positivity. $\mathcal{E}$ is completely positive if $\mathcal{I}_R \otimes \mathcal{E}$ is positive for any arbitrary system R, where $\mathcal{I}$ is the identity channel. An operator $\mathcal{E}$ is positive if $\mathcal{E}(\rho)$ is positive semi-definite for all ρ positive semi-definite operators. (iii) Trace preservation. $\mathrm{Tr}(\rho) = \mathrm{Tr}(\mathcal{E}(\rho))$ for any linear operator ρ.

Some norms we will be using for a linear operator A acting on the Hilbert space $\mathcal{H}$ are the spectral norm $\|A\| = \sup_{|\psi\rangle\neq 0} \frac{\|A|\psi\rangle\|}{\||\psi\rangle\|}$ where the vector norm used is the Euclidean norm and the max norm $\|A\|_{\max} = \max_{j,k} \left|A_{jk}\right|$ where A_{jk} is the (j,k) entry of A.

2.2 Theory of computation

The theory of computation concerns itself with defining what is computation itself through rigorous means by defining a model of computation and studies which problems can be solved in different computational models and the corresponding resource requirements. We consider an alphabet Σ, the elements of this alphabet are called symbols and in this thesis we usually consider the binary alphabet $\{0,1\}$. A string w over the alphabet Σ is a finite sequence of symbols from Σ. The number of symbols in w is called the length of the string (or bitstring when considering the binary alphabet). The set of strings of length n is denoted Σ^n and the set of all strings over Σ is denoted $\Sigma^* = \cup_{i\in\mathbb{N}}\Sigma^i$. A problem can be described in many different ways as presented in Section 2.3, the most basic notion is given by a language $L \subseteq \{0,1\}^*$, where $\{0,1\}^*$ is the set of all binary strings. Roughly stated, we want to determine which strings belong to L through algorithmic means. One of the most known computational models used in classical computing is that of a Turing machine which formalizes the notion of an algorithm. A precise definition of the Turing machine can be found in [AB09]. The output of a Turing machine M on an input state x is denoted as $M(x)$.

Another widely known computational model is that of Boolean circuits. In this model, the input is processed by a sequence of logical gates which compute some Boolean function. A Boolean circuit is defined as a directed acyclic graph where each of the vertices are labeled as follows. Each vertex of in-degree 0 is labeled as an input node. Every vertex with in-degree 1 is labeled as a *negation* vertex and every vertex of in-degree ≥ 2 is labeled either as an AND vertex or an OR vertex, these vertices with in-degree greater than 1 are also called gates. There is a unique vertex with out-degree 0 which is denoted as the output node. This graph represents a sequence of computations through a logical circuit where the gates are given by the AND, OR and negation logical operator. In the previous definition we have assumed a fixed gate set based on these logical operators but it is equivalent in computational power to circuits based on a different universal basis. We denote the in-degree of the logical gates as fan-in and the out-degree as fan-out. In terms of computational power, the Turing machine and the circuit model are equivalent as the can simulate each other with some polynomial overhead. Sometimes it is useful to consider the notion of oracle machines, these are Turing machines which can enter a special state which allows to decide in one step whether some string that the Turing machine holds belongs to some fixed language L. Such Turing machines M with access to an oracle of the language L are denoted as M^L. Details on the definitions for these models can be found in [AB09], we will not go into the detail of these since we will be mainly working with quantum circuits in this thesis. Sometimes we require the use of quantum oracles and the definition of these will be given when needed. When using circuits as a model for computation it is natural to consider a uniformity condition which. We say a circuit family $\{C_n\}_{n\in\mathbb{N}}$ is a uniform family of circuits if there is a Turing machine M running in polynomial time such that for each n, $M(x)$ outputs a descriptions of circuit C_n, where the length of bitstring x is n.

2.2.1 The quantum circuit model

To represent computations done with quantum systems, we use the quantum circuit model. A circuit (or more precisely, a family of circuits) represents a quantum algorithm. A quantum computation is described over a certain number n of qubits or qudits (a register), the quantum circuit is given by a unitary U acting over this register. The register is initialized in the state $|0^n\rangle$. It is common to give a decomposition of U into one and two qubit gates. As in the classical case, it is natural to consider uniform family of quantum circuits $\{U_n\}_{n\in\mathbb{N}}$ which are generated as outputs of a classical Turing machine running in polynomial time.

2.3 Complexity theory

The main goal in complexity theory is to classify the resources required when solving problems through algorithmic means as defined by a computational model. We give a very quick summary of some concepts in complexity theory which we use in this thesis, for a deeper introduction to this topic a book such as [AB09, Gol08] is suggested. Most commonly, the problems studied are so-called decision problems.

Definition 2.1 (Decision problem [Gol08]). Let $L \subseteq \{0,1\}^*$. The decision problem of L consists in given an input $x \in \{0,1\}^*$, output 1 or 0 deciding if $x \in L$. The map $f : \{0,1\}^* \to \{0,1\}$ solves the decision problem of L (or decides L) if for $x \in L$, $f(x) = 1$ and for $x \notin L$, $f(x) = 0$.

Sometimes we are just interested in a computational problem that gives answers to a well-defined subset of all possible inputs. This corresponds to the notion of promise problems.

Definition 2.2 (Promise problem [Gol08]). Let $A \subseteq \{0,1\}^*$ such that $A = L_{yes} \cup L_{no}$ and $L_{yes} \cap L_{no} = \emptyset$. The promise problem of A consists in given $x \in A$, output 1 if $x \in L_{yes}$ and 0 if $x \in L_{no}$. There is no output required if $x \notin A$. We call A the promise.

On the other hand, one could consider computational problems of a different form. Particularly relevant for this thesis are sampling problems.

Definition 2.3 (Sampling problem [Aar14]). Let $\mathcal{D} = \{\mathcal{D}_x\}$ be a family of probability distributions with $x \in \{0,1\}^*$ and where $\mathcal{D}_x$ is a probability distribution over bitstrings $y \in \{0,1\}^{\text{poly}(|x|)}$ for some polynomial. The sampling problem of $\mathcal{D}$ consists in given an input $x \in \{0,1\}^*$, output $y \in \{0,1\}^{\text{poly}(|x|)}$ with probability $\mathcal{D}_x(y)$.

The resources considered in complexity theory are usually time and space (though once could consider others such as amount of randomness, number of communication interactions, etc...). We quickly review the main complexity classes that are relevant for this thesis. The class P defined below captures the notion of tractable problems for deterministic algorithms.

Definition 2.4 (Polynomial-Time decidable (P) [Gol08]). The class of decision problems solvable by a polynomial-time Turing Machine (TM). More precisely, there is a Turing machine M which on input $x \in \{0,1\}^n$ runs poly(n) steps and $M(x) = 1$ if and only if $x \in L$.

An alternative definition of P is given by those decision problems such that there is a uniform family of circuits $\{C_n\}_n$ such that $C_{|x|}(x) = 1$ if an only if $x \in L$, where $|x|$ is the number of bits in x.

The class NP below captures the notion of problems that have efficient procedures to check whether a purported solution is correct. Later we will define the notion of complete problems, which correspond to those problems to which all problems in a class can be reduced to. Problems that are complete for NP are considered intractable for classical algorithms.

Definition 2.5 (Non-Deterministic Polynomial-Time decidable (NP) [Gol08]). The class of decision problems solvable by a non-deterministc polynomial-time Turing Machine (NTM). Equivalently, it is the class of problems for which instances that

are accepted have proofs that can be verified in poly-time by a TM and non-accepted instances don't have such proofs.

Sometimes we will be interested in complexity classes defined in terms of oracles. We denote as P^L as the class of problems decidable by polynomial-time TM with oracle access to L, also define NP^L as the class of problem decidable by non-deterministic polynomial-time TM with oracle access to L. An important notion in complexity theory is that of a reduction which allows to relate the complexity of different computational problems.

Definition 2.6 ([AB09]). A language $L \subseteq \{0,1\}^*$ is polynomial-time Karp reducible to language $L' \subseteq \{0,1\}^*$ if there is a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\forall x \in \{0,1\}^*, x \in L \iff f(x) \in L'$.

For a complexity class C, we say that L' is C-hard if $\forall L \in C$, L reduces to L'. L' is C-complete when $L' \in C$ and L' is C-hard. A well known problem in NP (in fact NP-complete) is that of κ-SAT. A Boolean formula ϕ over variables $x_1, \cdots, x_n$ consists of a finite expression which includes logical operators acting over these variables (for a rigorous definition see [AB09]). A formula ϕ is said to be satisfiable if there is an assignment of the variables $x_1, \cdots, x_n \in \{0,1\}$ such that $\phi(x_1, \cdots, x_n) = 1$, otherwise ϕ is said to be unsatisfiable. The formula ϕ is said to be in CNF form if it is written as an AND of OR's.

Definition 2.7 (κ-SAT[AB09]).

Instance: A Boolean formula ϕ in CNF form such that each clause has at most k variables.

Problem: Decide whether ϕ is satisfiable.

We point out two classes defined from NP class. One of them is defined in analogy to NP focused on counting problems known as $\#\mathsf{P}$. The second is a generalization of NP which is known as the polynomial hierarchy.

Definition 2.8 (#P). The class of counting problems associated to counting the number of accepting paths in a NTM.

Definition 2.9 (Polynomial Hierarchy (PH)). Consider

$$\begin{aligned}\Delta_0 = \Sigma_0 = \Pi_0 &= \mathsf{P}\\ \Delta_{i+1} &= \mathsf{P}^{\Sigma_i}\\ \Sigma_{i+1} &= \mathsf{NP}^{\Sigma_i}\\ \Pi_{i+1} &= \mathrm{co}\mathsf{NP}^{\Pi_i}\end{aligned}$$

The polynomial hierarchy is defined as

$$\mathsf{PH} = \bigcup_{k\in\mathbb{N}} \Delta_k = \bigcup_{k\in\mathbb{N}} \Sigma_k \tag{2.1}$$

Theorem 2.10 (Toda's Theorem [Tod91]). $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

The P vs NP problem is the central problem of complexity theory. It is believed that $\mathsf{P} \neq \mathsf{NP}$ and thus that NP-complete problems are intractable for deterministic classical algorithms. Therefore, we would not expect polynomial-time algorithms for problems like 3-SAT. Nonetheless, this leaves open the question about the complexity about the best possible algorithm for 3-SAT. More formally, consider the following constants.

Definition 2.11. For any $k \geq 3$, define

$$s_k := \inf\left\{\delta : \text{K-SAT has an } O\left(2^{\delta N}\right)\text{-time algorithm}\right\}$$

where N is the number of variables in the K-SAT instance and with algorithm we refer to classical deterministic algorithms.

Finding s_3 would give us the best possible complexity for solving 3-SAT. If $\mathsf{P} = \mathsf{NP}$ or there is some subexponential algorithm with runtime for 3-SAT (for example $O\left(2^{\sqrt{N}}\right)$), then clearly $s_3 = 0$. Note that the s_i form a monotone increasing sequence, i.e., $s_i \leq s_{i+1}$ for $i \geq 3$. We define the exponential time hypothesis as follows.

Conjecture 2.12 (Exponential Time Hypothesis [IP01]). *With Definition 2.11,* $s_3 > 0$. *Equivalently,* $\exists\ \delta > 0$ *such that* 3-SAT *cannot be solved in time* $O\left(2^{\delta N}\right)$ *with classical deterministic algorithms.*

This conjecture is of course stronger than $\mathsf{P} \neq \mathsf{NP}$. Although the conjecture makes reference to the SAT problem, in fact it has consequences over many other problems. The connection to other computational problems is made through the so called sparsification lemma [IPZ01]. This allows one to show that several problems have no subexponential algorithms if one believes Conjecture 2.12.

Lemma 2.13 (Sparsification Lemma [IPZ01]). *Let* $k \in \mathbb{N}$ *and* $\epsilon > 0$. *There is a constant* $C = C(k, \epsilon)$ *and an algorithm* $\mathcal{A}$ *such that*

- *Given a* k-SAT *formula* ϕ *with* N *variables,* $\mathcal{A}(\phi)$ *outputs formulae* $\phi_1, \cdots, \phi_t$.
- ϕ *is satisfiable* $\iff \exists\ i \in \{1, \cdots, t\}$ *such that* ϕ_i *is satisfiable.*
- *We have* $t \leq 2^{\epsilon N}$ *and* $\mathcal{A}$ *runs in time* $O\left(2^{\epsilon N}\text{poly}(N)\right)$.
- *Each* ϕ_i *is a* k-SAT *formula with* N *variables and* $M_i \leq CN$, *where* M_i *is the number of clauses in* ϕ_i.

Some authors define a slightly weaker version of Conjecture 2.12 which we state in the following.

Conjecture 2.14 (Exponential time Hypothesis, slightly weaker version (ETH)[DF13]). *The problem* 3-SAT *cannot be solved in time* $2^{o(n)}$.

Clearly, if there is a $2^{o(n)}$ algorithm then $s_3 = 0$, thus Conjecture 2.12 implies Conjecture 2.14. This last version is also relevant in parameterized complexity (which we introduce in Section 2.3.1) where it allows to prove the intractability of several problems assuming ETH. In this thesis we will focus on generalizations of this last definition for ETH.

2.3.1 Parameterized complexity

Having presented the most studied complexity classes in Section 2.3 we now introduce parameterized complexity theory, which allows for a classification of the hardness of problems at a finer scale. As mentioned in the introduction, part of the work in this thesis involved creating a framework for a quantum version of parameterized complexity theory. Here we will present the basics of the classical theory.

First, let's begin by defining the notion of a parameterization and of a parameterized problem.

Definition 2.15 (Parameterization [DF13]). A parameterization of a finite alphabet Σ is a mapping $\kappa : \Sigma^* \to \mathbb{Z}^+$ that is polynomial-time computable. The trivial parameterization κ_{trivial} is the parameterization with $\kappa_{\text{trivial}}(x) = 1$ for all $x \in \Sigma^*$.

Definition 2.16 (Parameterized problem [DF13]). A parameterized problem over a finite alphabet Σ is a pair (L, κ) where $L \subseteq \Sigma^*$ is a set of strings over Σ and κ is a parameterization of Σ. We say that a parameterized problem (L, κ) over the alphabet Σ is *trivial* if either $L = \varnothing$ or $L = \Sigma^*$.

Let us consider a simple example to clarify the notion of a parameterized problem. We define the parameterized satisfiability problem as follows.

Definition 2.17 (P-SAT [FG06]).

Instance: A Boolean formula ϕ over k Boolean variables.

Parameter: A natural number k denoting the number of variables.

Problem: Decide whether ϕ is satisfiable.

In this case, the parameterization κ from Definition 2.15 gives the number of Boolean variables in the instance given by the Boolean formula ϕ. We can consider other problems like the vertex cover problem in a parameterized setting. A vertex cover for a graph

$G = (V, E)$ is a collection of vertices $V' \subseteq V$ such that for all edges $e = (v_1, v_2) \in E$, either $v_1 \in V'$ or $v_2 \in V'$.

Definition 2.18 (k-VERTEX COVER [DF13]).

Instance: A graph $G = (V, E)$ and a natural number k.

Parameter: A natural number k.

Problem: Decide whether G has a vertex cover of size $\leq k$.

In this case the parameterization simply gives the size of the vertex cover one seeks in the problem description.

With these definitions in place, we define a tractable algorithm when parameters are included in the problem description.

Definition 2.19 (Fixed-Parameter Tractable (FPT) [DF13]). A parameterized language (L, κ) is said to be fixed-parameter tractable (FPT) if and only if there is an algorithm $\mathcal{A}$, a constant c and a computable function f such that, algorithm $\mathcal{A}(x)$ runs in time at most $f(\kappa(x))|x|^c$ and $x \in L \iff \mathcal{A}(x)$ accepts.

We can think of the FPT class as a parameterized version of P. Although the runtime is still inefficient when the parameter k is not fixed, the combinatorial explosion has been isolated in the $f(k)$ factor. Trivially, the problem P-SAT is in FPT since we can decide the problem for any formula ϕ with m clauses in time $O(2^k m)$, where k is the number of variables. A more interesting example is given by k-VERTEX COVER. By brute force, the runtime to solve this problem is $O(n^k)$ where n is the number of vertices in the graph and k is the size of the vertex cover. This runtime is not enough to put the problem in FPT as the dependence on the parameter is in the exponent. Several techniques have been developed to obtain better runtimes in parameterized complexity such as bounded search tree methods and kernelization (for more details on these techniques see [DF13, FG06]). It was shown by Fellows and Langston [FL87] that the Robertson-Seymour graph minor theorem can be used to solve the k-VERTEX COVER

in time $O(f(k)n^3)$ which puts the problem in FPT. Note that k-VERTEX COVER is an NP-complete problem, yet it can be considered tractable in the parameterized setting.

The notion of a parameterized reduction will be important when considering intractability. We give a definition of a fpt-reduction in what follows.

Definition 2.20 (Fixed-parameter tractable reduction [FG06]). Let (L, κ) and (L', κ') be parameterized problems over alphabets Σ and Σ' respectively. An fpt-reduction from (L, κ) to (L', κ') is a mapping $R : \Sigma^* \to (\Sigma')^*$ such that:

(1) For all $x \in \Sigma^*$ we have $x \in L \iff R(x) \in L'$

(2) R is computable by an fpt-algorithm with respect to κ. That is, there is a computable function f and a polynomial p such that $R(x)$ is computable in $f(\kappa(x))p(|x|)$.

(3) There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that $\kappa'(R(x)) \leq g(\kappa(x))$ for all $x \in \Sigma^*$.

The first two conditions are quite natural for an fpt-reduction. Condition (3) is required to make FPT closed under fpt-reductions. For more detail on this, see Lemma 2.2 in [FG06].

There are several ways to consider intractable classes in the parameterized setting and to generalize NP in parameterized complexity. One class which is considered intractable is para-NP.

Definition 2.21 (para-NP **[FG06]).** A parameterized problem (L, κ) over the alphabet Σ is in para-NP if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, a polynomial p and a nondeterministic algorithm that, given $x \in \Sigma^*$, decides if $x \in L$ in at most $f(\kappa(x))p(|x|)$ steps.

While para-NP seems like a natural definition for a generalization of NP, it suffers certain problems. The issue is that we want to capture intractability in the parameterized setting, or more simply stated, we want to capture when is it possible to have runtimes

of the form $f(k)\,\mathrm{poly}(n)$ instead of $n^{f(k)}$ as in the k-VERTEX COVER problem. For example, we will see below that problems like k-INDEPENDENT SET are unlikely to have tractable runtimes, yet it is complete for a class which is inside para-NP. This motivates the definition of further complexity classes. For details on this, see Section 2.2 in [FG06]. Another complexity class considered intractable is that of XP.

Definition 2.22 (XP [FG06]). Let (L, κ) be a parameterized problem over the alphabet Σ. Then (L, κ) belongs to the class XP if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given $x \in \Sigma^*$, decides if $x \in L$ in at most $|x|^{f(\kappa(x))} + f(\kappa(x))$.

In contrast to FPT, the dependence on the parameter appears in the exponent of the size of the instance, which is considered intractable in parameterized complexity. Clearly, $\mathsf{FPT} \subseteq \mathsf{XP}$. In this thesis we mostly focus in the so-called Weft-hierarchy (W-hierarchy) due to its relevance to Chapter 4 and its connection to ETH. We begin by defining W[P].

Definition 2.23 (W[P] [DF13]). A parameterized problem (L, κ) over the alphabet Σ is in W[P] if there is a verification procedure $\{\mathcal{V}_{n,k}\}_{n,k\in\mathbb{Z}^+}$ such that the following conditions are satisfied.

(1) There is a computable function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ and a polynomial $p \in \mathbb{N}[X]$, such that, for every $x \in \Sigma^*$, $\mathcal{V}_{|x|,\kappa(x)}$ on input x runs in time at most $f(\kappa(x)) \cdot p(|x|)$ on a deterministic Turing machine.

(2) For every $x \in \Sigma^*$,

- If $x \in L$, then there exists a bit string y comprising at most $f(\kappa(x)) \cdot \log |x|$ bits, such that $\mathcal{V}_{|x|,\kappa(x)}(x, y)$ accepts.
- If $x \notin L$, then for every bit string y comprising at most $f(\kappa(x)) \cdot \log |x|$ bits, $\mathcal{V}_{|x|,\kappa(x)}(x, y)$ rejects.

It can be shown easily that $\mathsf{FPT} \subseteq \mathsf{W[P]}$ and $\mathsf{W[P]} \subseteq \mathsf{XP}$ [FG06]. Intuitively, the class corresponds to problems decidable by algorithms running in $f(k)p(n)$ time and

allowed to do $O(f(k)\log n)$ nondeterministic choices. The number of nondeterministic choices is limited in this way because we want to accept those problems that are only inside XP. Said in another way, we can intuitively think of W[P] as the problems where we want to find k items fulfilling certain condition from a set of n elements. We can then encode this choice roughly with $k \cdot \log n$ bits which corresponds to the witness. An example of a problem in W[P] is k-VERTEX COVER, we can guess k times an element in the vertex cover which is encoded by $\log n$ bits and then check that these elements indeed form a cover. It is believed that FPT $\neq$ W[P] since otherwise one could show the existence of subexponential algorithms for circuit satisfiability. For details on this see Chapter 3 of [FG06]. A complete problem for W[P] is given by the WEIGHT-k CIRCUIT SATISFIABILITY problem.

Definition 2.24 (WEIGHT-k CIRCUIT SATISFIABILITY).

Instance: A Boolean circuit C over n input bits and a natural number k.

Parameter: A natural number k.

Problem: Decide whether there exists an n-bit Hamming weight-k string y, such that $C(y)$ accepts.

The idea of parameterizing the Hamming-weight is that it intuitively captures the idea of choosing k elements out of a set of size n. The circuit checks whether this set has some property. For example in the K-VERTEX COVER problem we want a set of size k which is a vertex cover.

An important notion in what follows is that of the circuit weft. Moreover, we give a problem based on the circuit weft which defines the W-hierarchy.

Definition 2.25 (Circuit Weft). Let C be a Boolean circuit. The *weft* of C is the maximum number of gates with fan-in ≥ 2 that act on any path from input to output vertex when C is viewed as a directed acyclic graph.

Finally, the weft-hierarchy (W-hierarchy) is defined as follows.

Definition 2.26 (WEIGHT-k WEFT-t DEPTH-d CIRCUIT SATISFIABILITY).

Instance: A weft-t depth-d Boolean circuit C on n input bits.

Parameter: A natural number k.

Problem: Decide whether there exists an n-bit Hamming weight-k string y, such that $C(y)$ accepts.

Definition 2.27 ($\mathsf{W}[t]$). For $t \in \mathbb{N}$, the class $\mathsf{W}[t]$ consists of all parameterized problems that are FPT reducible to WEIGHT-k WEFT-t DEPTH-d CIRCUIT SATISFIABILITY for some $d \geq t$.

It can be shown that $\mathsf{W}[t] \subseteq \mathsf{W}[\mathsf{P}]$ for every $t \geq 1$. It is an open problem whether the W-hierarchy is strict, i.e., $\mathsf{W}[t]$ is strictly contained in $\mathsf{W}[t+1]$ for all t. The strictness would imply that $\mathsf{FPT} \neq \mathsf{W}[\mathsf{P}]$. In this thesis we mainly work with the quantum version of $\mathsf{W}[1]$ which we define in Chapter 3. An example of a problem which is complete for $\mathsf{W}[1]$ is that of k-INDEPENDENT SET. An independent set for a graph $G = (V, E)$ is defined as a set of vertices V' such that for any $v_1, v_2 \in V'$, $e = (v_1, v_2) \notin E$.

Definition 2.28 (k-INDEPENDENT SET).

Instance: A graph $G = (V, E)$ and a natural number k.

Parameter: A natural number k.

Problem: Decide whether G has independent set of size k.

A proof that this problem is $\mathsf{W}[1]$-complete can be found in [FG06]. Note that both k-VERTEX COVER and k-INTEPENDENT SET are NP-complete problems. In the parameterized setting, we see a separation in the complexity of these two problems since k-VERTEX COVER is in FPT and k-INTEPENDENT SET is $\mathsf{W}[1]$-complete. These kind of separations make parameterized complexity an interesting setting in which one gets a more fine-grained view of the complexity of problems.

It has been shown that FPT = W[1] implies that the ETH is false [DF13], providing a connection between non-parameterized and parameterized complexity. We will see more details about this and analogous results in Chapter 4.

2.3.2 Quantum complexity Theory

In this subsection we define some of the quantum complexity classes used in this thesis. A more in-depth analysis of this classes can be found in [KSV02] and a review of many other classes is given in [Wat08]. Just as in the classical case, a quantum analogue of P would include the problems that we can solve efficiently with quantum computers. This class is known as BQP.

Definition 2.29 (Bounded Quantum Polynomial Time (BQP)). The promise problem $L = (L_{yes}, L_{no})$ is in BQP if and only if there exist a uniform family of quantum circuits $\{Q_n\}_{n\in\mathbb{N}}$ where each Q_n accepts n input qubits and has one output qubit, such that

- if $x \in L_{yes}$, $\Pr\big(Q_{|x|}(x) \text{ accepts}\big) \geq 2/3$
- if $x \in L_{no}$, $\Pr\big(Q_{|x|}(x) \text{ accepts}\big) \leq 1/3$

where $|x|$ denotes the length of x.

The most studied quantum version of NP is defined in terms of a verifier with access to quantum polynomial-time computations and an all-powerful prover who must supply a quantum state.

Definition 2.30 (Quantum Merlin-Arthur (QMA)). Let $L = (L_{yes}, L_{no})$ be a promise problem. We say $L \in$ QMA if and only if there exists a polynomial p and a uniform family of circuits $\{Q_n\}_{n\in\mathbb{N}}$ where each Q_n takes $n + p(n)$ inputs and has one output qubit such that

- If $x \in L_{yes}$, $\exists\, |\psi\rangle$ a $p(n)$ qubit state, $\Pr(Q(x, |\psi\rangle) \text{ accepts}) \geq 2/3$
- If $x \in L_{no}$, $\forall\, |\psi\rangle$ a $p(n)$ qubit state, $\Pr(Q(x, |\psi\rangle) \text{ accepts}) \leq 1/3$

We call the first condition completeness and the second the soundness of the protocol.

Many QMA-complete problems have been found and a high-level summary of these is given in [Boo12]. One of the most important is the local Hamiltonian problem [KSV02] which plays a role analogous to k-SAT in quantum complexity theory.

Local Hamiltonian

The local Hamiltonian problem is usually considered a natural computational problem. The conventional wisdom says that a common problem that physicists and chemists encounter when studying quantum systems is that of computing the ground state energies of a given Hamiltonian with locality properties.

Definition 2.31 (ℓ-LOCAL HAMILTONIAN).

Instance: An ℓ-local Hamiltonian $H \coloneqq \sum_i H_i$ on n qubits that comprises at most a polynomial in n many terms $\{H_i\}$, which each act non-trivially on at most ℓ qubits and have operator norm $\|H_i\| \leq 1$. Numbers a, b such that $0 \leq a < b$ and $b - a \geq 1/\mathrm{poly(n)}$.

Yes: There exists an n-qubit quantum state $|\psi\rangle$, such that $\langle\psi|H|\psi\rangle \leq a$.

No: For every n-qubit quantum state $|\psi\rangle$, $\langle\psi|H|\psi\rangle \geq b$.

As mentioned the ℓ-LOCAL HAMILTONIAN problem is QMA-complete when ℓ is sufficiently large, which was originally proven by Kitaev [KSV02]. The original proof showed that the 5-LOCAL HAMILTONIAN problem is QMA-complete. This result was later improved to showing that 3-LOCAL HAMILTONIAN is QMA-complete [KR03] and further improved to showing 2-LOCAL HAMILTONIAN as QMA-complete [KKR05]. This problem includes a wide variety of Hamiltonians encountered in physics, nonetheless, many times the Hamiltonians of interest have further constraints. Some of these constraints may include a particular interaction graph (for example a grid) or interactions constrained to be of a certain form (for example as in the XY-model).

In the case of 2D grid local Hamiltonian, it was shown in [OT08] that the problem remains QMA-complete. In a series of works [BL08, CM16, PM17] the QMA-completeness of several variations of the local Hamiltonian based on physical models was established. In [LCV07], Hamiltonians which are quartic polynomials in the creation and destruction operators of fermionic creation and destruction operators were considered and proven to be QMA-complete. The case for bosons was shown QMA-complete in [WMN10].

In the previous summary we can see that the tendency has been to modify the Hamiltonian in order to get a sense of the hardness of problems that are actually encountered in the physical sciences. Recent work has also focused on modifying the problem itself by adding certain parameters or including additional promises. In quantum chemistry one is interested in finding the groundstate energy of a Hamiltonian whose description also includes the choice of a basis. In [OIWF22], the authors show that when this basis is fixed, the electronic structure Hamiltonian is QMA-complete for a fixed particle number. The authors leave open several questions, such as for example the hardness of the problem when the basis is promised certain error or the hardness of finding an appropriate basis.

Further work inspired in quantum chemistry is that of [GL21], where the authors add to the local Hamiltonian problem the promise of having access to a quantum state which is close to the groundstate in overlap. The authors prove that this problem is in fact BQP-complete for 6-local Hamiltonians. This result was later improved to 2-local Hamiltonians [GL21, CFW22].

In Chapter 4 we study a parameterized version of the local Hamiltonian where the weight of the groundstate is considered as a parameter. We consider this as a continuation of a line of work which seeks to study versions of the local Hamiltonian problem with applications to concrete problems.

2.4 Hamiltonian simulation

One of the first applications for quantum computers where an advantage might be expected is in the simulation of other quantum systems [Fey82, Fey86]. A basic problem in simulating a quantum systems involves evolving a quantum state with a given Hamiltonian for some prescribed time. This problem is known as the Hamiltonian simulation problem which we detail below.

Problem 1 (Hamiltonian Simulation). Given a Hamiltonian H over n qubits (a $2^n \times 2^n$ Hermitian matrix), a time $t > 0$ and an error parameter $\epsilon > 0$, output a quantum circuit U of poly(n) size such that $\left\|U - e^{iHt}\right\| \leq \epsilon$.

Many different methods have been proposed to solve this problem. In this thesis in Part III we will work with the product formula method but for completeness we will give a summary of other methods in the literature. Although in terms of error dependence product formulae are worse than more recent methods, an advantage is that the implementation is easier for near-term devices. Moreover, some problems just require a constant error, such as quantum chemistry where a chemical accuracy must be reached. For this reasons, studying product formulae is still relevant for quantum simulation. It is important to note that all these methods or algorithms assume some way to access the given Hamiltonian as its size as a matrix is exponential in the number of qubits. We will assume that the Hamiltonian H from Problem 1 is d-sparse, meaning that each column has no more than d non zero entries. We define the following oracles [CB12].

Definition 2.32 (Oracle access to entries of H). Given Hamiltonian H over n qubits, with entries H_{jk} for $j, k \in \{1, 2, \cdots, n\}$. Let the oracle U_H be defined as

$$U_H \left|j, k\right\rangle \left|z\right\rangle = \left|j, k\right\rangle \left|z \oplus H_{jk}\right\rangle, \tag{2.2}$$

where $\oplus$ is the bitwise XOR operator.

Definition 2.33 (Oracle access to non-zero entries of H). Let H be a Hamiltonian over n qubits. Let U_f be an oracle such that

$$U_f |j,k\rangle = |j, f(j,k)\rangle , \tag{2.3}$$

for $j \in \{1, \cdots, n\}$, $k \in \{1, \cdots, d\}$ and where $f(j,k)$ gives the row index of the kth non zero element in column j of H.

2.4.1 Product formulae

The Lie-Trotter product formula (PF) is commonly used in quantum algorithms for Hamiltonian simulation, where one can approximate the Hamiltonian evolution of $H = A + B$ in terms of exponentials of A and B when these operators do not commute. It is well known that

$$\exp(-iHt) = \exp(-iAt)\exp(-iBt) + O\left(t^2\right). \tag{2.4}$$

This particular product formula was applied in the context of quantum computing in [Llo96]. More general PF can be constructed by including more exponentials and parameters which give higher order errors. A summary of the methods to obtain higher-order PF is given in Section 6.2.1 in part III of this thesis.

In [BACS07, CK11] an estimation of the number of calls to the oracle U_f is given to implement an order k integrator. The number of such calls is given by

$$O\left(\frac{5^{2k} d^2 (d + \log^* n) \|H\| t (\|H\| t)^{1/2k}}{\epsilon^{1/2k}}\right). \tag{2.5}$$

This estimation is obtained assuming that higher-order integrators are obtained using the Suzuki method which we introduce in the paragraph with Eq. (6.11) which explains the appearence of the factor 5^{2k}. Alternative methods to obtain higher order integrators exist such as Yoshida's method [Yos90], we introduce and expand this method in Chapter 6.

2.4.2 Taylor series

Since the evolution under a Hamiltonian H is given by $U = e^{-iHt}$, we can consider the truncated Taylor expansion of the exponential $U \simeq \sum_{j=0}^{K} \frac{(-iHt)^j}{j!}$ [BCC+15]. This sum can be implemented using the linear combination of unitaries (LCU) method. In terms of oracle calls, the complexity is given by

$$O\left(d^2\|H\|t\frac{\log\left(d^2\|H\|t/\epsilon\right)}{\log\log(d^2\|H\|t/\epsilon)}\right). \tag{2.6}$$

A 2-qubit gate count is also provided in [BCC+15]

$$O\left(nd^2\|H\|t\frac{\log^2(d^2\|H\|t/\epsilon)}{\log\log(d^2\|H\|t/\epsilon)}\right). \tag{2.7}$$

And the number of ancillae used as

$$O\left(\frac{\log^2(d^2\|H\|t/\epsilon)}{\log\log(d^2\|H\|t/\epsilon)}\right). \tag{2.8}$$

2.4.3 Quantum walks

Previous methods presented so far depended on decomposing the total evolution time into short segments, this has the problem that the scaling on the sparsity is quadratic in the Taylor series approach and cubic in the product formula approach. The methods introduced here based on quantum walks allow for a linear scaling on the sparsity d. We refer to two methods based on quantum walks in this subsection.

Phase estimation on quantum walks. In [CB12] an algorithm based on oracles in Definition 2.32 and Definition 2.33 is given which uses a discrete-time quantum walk to simulate evolution. The query complexity of this algorithm is given by

$$O\left(\frac{\|H\|t}{\sqrt{\epsilon}} + d\|H\|_{max}t\right). \tag{2.9}$$

As mentioned earlier, the dependence on d is linear. Moreover, this method achieves linear dependence on the evolution time t. This method has the problem that the dependence on the error ϵ is much worse than, for example, the Taylor series method.

On the other hand, the quantum walk achieves a linear dependence on the sparsity d, which improves over other methods that only get quadratic dependence.

Linear combination of quantum walks. By implementing the quantum walk with the LCU technique [BCK15], it is possible to improve on the previous methods. In [BCK15] the authors give a method to achieve a good scaling in both sparsity and the error ϵ. The number of queries to oracles in Definition 2.32 and Definition 2.33 is given by

$$O\left(d\|H\|_{max}t\frac{\log(d\|H\|_{max}t/\epsilon)}{\log\log(d\|H\|_{max}t/\epsilon)}\right), \tag{2.10}$$

and the number of 2-qubit gates is

$$O\left([n+\log^{5/2}(d\|H\|_{max}t/\epsilon)]d\|H\|_{max}t\frac{\log(d\|H\|_{max}t/\epsilon)}{\log\log(d\|H\|_{max}t/\epsilon)}\right). \tag{2.11}$$

2.4.4 Quantum signal processing

The quantum signal processing technique [LC17a] achieves optimal dependence on the parameters by using techniques from optimal quantum control. The query complexity is given by

$$O\left(d\|H\|_{max}t+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right). \tag{2.12}$$

The gate count is given by

$$O(n+m\text{polylog}(m)), \tag{2.13}$$

where m is the number of bits used to specify matrix elements in the Hamiltonian.

Although quantum signal processing achieves optimal dependence on all parameters, there is still much active research on improving the performance of Lie-Trotter product formulae. The reason is that product formulae have lower requirements in terms of gate implementation, in particular for certain Hamiltonians where locality is restricted [CST+21]. Moreover, for some applications like quantum chemistry the error ϵ is actually a constant (for example when considering some chemical error) and thus the $(1/\epsilon)^{2k}$ scaling of product formulae is not a problem when compared with better scalings such as that in quantum signal processing. In Chapter 6 we will give a more detailed

analysis for product formulae with random Hamiltonians which have similar form as those that appear in chemistry.

Part I

Parameterized Complexity and the Weighted Local Hamiltonian

Chapter 3

Quantum Parameterized Complexity

This chapter is based on work in [BJM+22]. This was work done in collaboration with Michael Bremner, Zhengfeng Ji, Ryan Mann, Luke Mathieson and Alexis Shaw. We introduce some of the main complexity classes in quantum parameterized complexity and set the stage to study a parameterized version of the local Hamiltonian problem in Chapter 4.

My contribution in [BJM+22] was mainly that of helping with writing Section II and Section III in that work. I contributed in the discussions for the proper definitions of the QW-hierarchy and in writing the proofs for some facts pertaining FPQT. In this chapter I have taken most definitions directly from our work and have been included for completeness in this thesis. In Section 3.3.1 I include some non-published work of my own regarding a parameterized version of the subset problem.

3.1 Introduction

Parameterized complexity seeks to classify the complexity of computational problems when the instance description includes parameters describing the instance itself. In Section 2.3.1 we gave a summary of the relevant definitions in parameterized complexity. In this chapter we give a quantum generalization of parameterized complexity,

focused on generalizations for FPT, W[P] and the W-hierarchy. This quantum version of parameterized complexity is motivated by the search for problems where there might be quantum advantages. We are interested in understanding whether there are tractable algorithms in the quantum parameterized setting that are not known either in the classical case, nor under the usual definition of BQP. A second motivation to develop a quantum parameterized complexity theory is that of having a fine-grained understanding of problems that are known to be QMA-complete. As remarked at the end of Section 2.3.1, problems that are NP-complete can have different complexity for parameterized complexity classes. This fact motivates the study of a parameterized version of the local Hamiltonian problem in Chapter 4.

In [ABNO22], the parameterized complexity of verifying QMA problems was studied. In particular, the parameter considered is the number of T gates used in the verification circuit for circuit satisfiability problems. The authors find lower bounds on the T-count for such problems assuming ETH. In our work, we have given generalizations of several classical parameterized complexity classes, we seek to give a theory of parameterization in general and not just in verification. A summary of our results, including computational problems in each of these classes is given in Fig. 3.1. A summary of the classes studied in [BJM+22] is given in Fig. 3.1. In this chapter we work mainly with the classes FPQT, W[P] and the QW-hierarchy. Although the definitions for FPQT and W[P] are analogous to the classical case, we find that a quantum versions of the W-hierarchy has some issues which also appear when trying to define a probabilistic version. The main issue with the QW is that of success amplification; recall that the W-hierarchy is defined in terms of constant depth circuits, this implies that the success amplification would be required to be done in constant depth. We will introduce a parameterized version of the subset problem from [BGKS16] where this will appear as an issue.

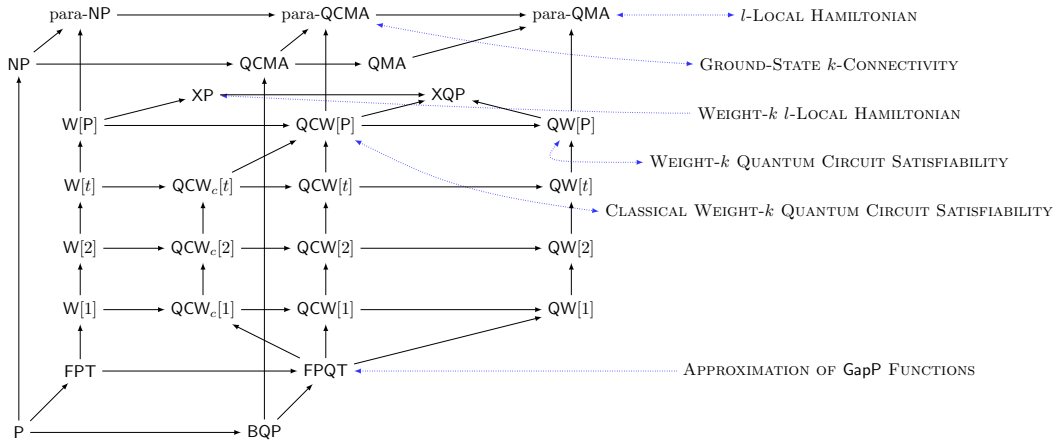

Figure 3.1: Complexity classes and problems discussed in our paper [BJM+22].

3.2 Tractability in quantum parameterized complexity

In parameterized complexity, a notion of tractability is given by the fixed-parameter tractable class known as FPT [FG06]. A quantum version of this class is defined in what follows.

Definition 3.1 (FPQT algorithm). Let (L, κ) be a parameterized problem over the alphabet Σ. An algorithm $\mathcal{A}$ is a FPQT *algorithm for* (L, κ) if the following conditions are satisfied.

(1) There is a computable function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ and a polynomial $p \in \mathbb{N}[X]$, such that, for every $x \in \Sigma^*$, the size of an FPT-uniform quantum circuit that computes $\mathcal{A}$ on input x is at most $f(\kappa(x)) \cdot p(|x|)$.

(2) For every $x \in \Sigma^*$,

- If $x \in L$, then $\mathbf{Pr}[\mathcal{A}(x) \text{ accepts}] \geq \frac{2}{3}$.
- If $x \notin L$, then $\mathbf{Pr}[\mathcal{A}(x) \text{ accepts}] \leq \frac{1}{3}$.

Definition 3.2 (FPQT). The class FPQT consists of all parameterized problems that have an FPQT algorithm.

This definition for FPQT is simply a generalization of the definition given in Definition 2.19 for FPT. As is natural for quantum algorithms, we consider quantum

circuits with uniformity condition based on FPT rather than P. The error probability of $1/3$ is completely arbitrary and can be replaced by any constant non-zero probability less than $1/2$.

In [BJM+22] we have proved some further facts about FPQT. As the main purpose of this chapter is to introduced the basic quantum parameterized classes and present the main issue with the definition in the QW-hierarchy, we will simply state these results without proof (proofs can be found in [BJM+22]).

Proposition 3.3. $\mathsf{FPT} = \mathsf{FPQT}$ *if and only if* $\mathsf{P} = \mathsf{BQP}$.

Proposition 3.4. $\mathsf{FPT}^{\mathsf{FPQT}} = \mathsf{FPQT}$.

Proposition 3.5. $\mathsf{FPT}^{\mathsf{BQP}} = \mathsf{FPT}^{\mathsf{FPQT}}$.

This last proposition offers an alternative characterization for FPQT since it implies $\mathsf{FPT}^{\mathsf{BQP}} = \mathsf{FPQT}$. By considering a BQP-complete problem in the oracle we can define FPQT-complete problems. Note as well that Proposition 3.3 indicates that there are likely quantum advantages in the parameterized setting when compared to classical algorithms.

The following proposition about FPQT will help us prove some later facts

Proposition 3.6. *If* $\mathsf{FPQT} \subseteq \mathsf{XP}$ *then* $\mathsf{P} = \mathsf{BQP}$.

Proof. If $\mathsf{FPQT} \subseteq \mathsf{XP}$ then any BQP-complete problem with trivial parameterization is contained in XP. Implying that $\mathsf{P} = \mathsf{BQP}$. ∎

It will also be important to define a notion of reduction in the quantum parameterized setting.

Definition 3.7. Let (L, κ) and (L', κ') be parameterized problems over the alphabets Σ and Σ' respectively. A *FPQT reduction* from (L, κ) to (L', κ') is a mapping $R : \Sigma^* \to (\Sigma')^*$ such that the following conditions are satisfied.

(1) For all $x \in \Sigma^*$, $x \in L \iff R(x) \in L'$.

(2) R is computable by an FPQT algorithm with respect to the parameter κ, (i.e. $R(x)$ is computable using an FPT-uniform collection of circuits of size $f(\kappa(x) \cdot p(|x|)$ with high probability).

(3) There is a computable function $g : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that $\kappa'(R(x)) \leq g(\kappa(x))$ for all $x \in \Sigma^*$.

3.3 Intractability in quantum parameterized complexity

The class NP can be generalized in several ways in the parameterized setting, the same is true when we consider quantum parameterized complexity. In what follows we give the definitions of some of these quantum generalizations which will be fundamental in proving intractability for quantum parameterized algorithms, in Fig. 3.1 we show the known containments for this classes (for more detail on their definition, see [BJM+22]). In [BJM+22] we considered several classes such as para-QMA which is a generalization of para-NP where the verifier is a quantum circuit of size at most $f(\kappa(x))p(|x|)$ for some computable function f and polynomial p. We also consider para-QCMA where the witness is classical. Moreover, we generalize W[P] and the W-hierarchy to its quantum versions QW[P] and the QW-hierarchy together with QCW which assumes a classical witness. In this chapter we only give the definitions for QW[P] and the QW-hierarchy as these will be relevant for Chapter 4.

The main challenge in defining the quantum versions of these classes is that of preserving the relations of tractability and intractability in the parameterized setting. Note that in Definition 2.23 the class W[P] allows up to $f(k) \log n$ nondeterministic choices for a parameter k and instance size n. In the quantum case, this is generalized by including as input in the verifier a quantum state over $f(k) \log n$ qubits. The quantum generalization of the W-hierarchy is not as direct, the quantum generalization of weft requires choosing a gate acting over many qubits carefully and there are also issues with

success amplification due to the constraints on depth of the verifying circuit. Before going into more detail about this, we give the definition for $\mathsf{QW}[\mathsf{P}]$.

Definition 3.8 ($\mathsf{QW}[\mathsf{P}]$). A parameterized problem (L, κ) over the alphabet Σ is in $\mathsf{QW}[\mathsf{P}](c, s)$ if there is a quantum verification procedure $\{\mathcal{V}_{n,k}\}_{n,k\in\mathbb{Z}^+}$ such that the following conditions are satisfied.

(1) There is a computable function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ and a polynomial $p \in \mathbb{N}[X]$, such that, for every $x \in \Sigma^*$, the size of an FPT-uniform quantum circuit that computes $\mathcal{V}_{|x|,\kappa(x)}$ on input x is at most $f(\kappa(x)) \cdot p(|x|)$.

(2) For every $x \in \Sigma^*$,

- If $x \in L$, then there exists a quantum state $|\psi\rangle$ comprising at most $f(\kappa(x)) \cdot \log|x|$ qubits, such that $\mathbf{Pr}[\mathcal{V}_{|x|,\kappa(x)}(x, |\psi\rangle) \text{ accepts}] \geq c$.
- If $x \notin L$, then for every quantum state $|\psi\rangle$ comprising at most $f(\kappa(x)) \cdot \log|x|$ qubits, $\mathbf{Pr}[\mathcal{V}_{|x|,\kappa(x)}(x, |\psi\rangle) \text{ accepts}] \leq s$.

The class $\mathsf{QW}[\mathsf{P}]$ is defined to be $\mathsf{QW}[\mathsf{P}](\frac{2}{3}, \frac{1}{3})$.

A complete problem for this class is the weighted version of quantum circuit satisfiability. Before introducing the problem, we define the notion of weight.

Definition 3.9 (Weight of a quantum state). A quantum state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ on n qubits is said to have *weight* k if $\alpha_x = 0$ for all x not of Hamming weight k.

With this definition of weight, we generalize the Hamming weight of a bitstring which was used in the definition of a complete problem for $\mathsf{W}[t]$ in Definition 2.24. We give a natural quantum generalization of the weighted circuit satisfiability problem.

Definition 3.10 (WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY).

Instance: A quantum circuit C on n witness qubits and $\text{poly}(n)$ ancilla qubits. Two positive numbers $a, b \in (0, 1)$, such that $b - a > \frac{1}{\text{poly}(n)}$.

Parameter: A natural number k.

Yes: There exists an n-qubit weight-k quantum state $|\psi\rangle$, such that $\mathbf{Pr}[C(|\psi\rangle) \text{ accepts}] \geq b$.

No: For every n-qubit weight-k quantum state $|\psi\rangle$, $\mathbf{Pr}[C(|\psi\rangle) \text{ accepts}] \leq a$.

The weighted quantum circuit satisfiability is in fact $\mathsf{QW}[\mathsf{P}]$-complete.

Proposition 3.11. WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY *is* $\mathsf{QW}[\mathsf{P}]$-complete *under FPQT reductions.*

Proof. Firstly, we show that WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY is in $\mathsf{QW}[\mathsf{P}]$. Let C be a quantum circuit on n qubits, k a natural number, and $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ a computable function. Further let $S_{n,k}$ denote the set of all n-bit strings with Hamming weight k and let ε be a binary enumeration of the elements of $S_{n,k}$. An n-qubit weight-k quantum state $|\psi\rangle = \sum_{x \in S_{n,k}} \alpha_x |x\rangle$ can be described using $f(k) \cdot \log(n)$ qubits by the quantum state $|\psi_\varepsilon\rangle = \sum_{x \in S_{n,k}} \alpha_x |\varepsilon(x)\rangle$. Let $\mathcal{M}_{n,k}$ be a verification procedure for deciding whether the a weight of an n-qubit quantum state is k. The verification procedure $\mathcal{V}_{n,k}$ constructs the state $|\psi\rangle$ from $|\psi_\varepsilon\rangle$ and accepts if and only if $C(\mathcal{M}_{n,k} |\psi\rangle)$ accepts. Applying the gap amplification scheme of Marriott and Watrous [MW05] to this procedure completes the claim.

We now prove that WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY is $\mathsf{QW}[\mathsf{P}]$-hard. Let (L, κ) be a problem in $\mathsf{QW}[\mathsf{P}]$ with verification procedure $\{\mathcal{V}_{n,k}\}_{n,k \in \mathbb{Z}^+}$. Further let $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ be a computable function and define $k_x := \kappa(x)$. For input $x \in \Sigma^*$, we shall construct a quantum circuit C_x that is satisfiable by a weight-k_x quantum state if and only if $\mathcal{V}_{|x|,\kappa(x)}(x)$ is satisfiable. The circuit C_x takes as input n qubits and firstly decides whether the input state has weight k_x using the verification procedure $\mathcal{M}_{n,k_x}$. Finally, the circuit inputs the quantum state into the verifier $\mathcal{V}_{|x|,\kappa(x)}$. Therefore, C_x is satisfiable by a weight-k_x quantum state if and only if $\mathcal{V}_{|x|,\kappa(x)}(x)$ is satisfiable. This completes the proof. ∎

It is natural to ask whether the parameterized version of l-LOCAL HAMILTONIAN which we denote as the weighted local Hamiltonian problem is QW[P]-complete. However, as we shall see this problem is in XP, and thus if it were QW[P]-complete then FPQT $\subseteq$ XP which would imply P = BQP. This makes unlikely that the weighted local Hamiltonian is QW[P]-complete under FPQT reductions.

Definition 3.12 (WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b)).

Instance: An ℓ-local Hamiltonian $H \coloneqq \sum_i H_i$ on n qubits that comprises at most a polynomial in n many terms $\{H_i\}$, which each act non-trivially on at most ℓ qubits and have operator norm $\|H_i\| \leq 1$.

Parameter: A natural number k.

Yes: There exists an n-qubit weight-k quantum state $|\psi\rangle$, such that $\langle\psi|H|\psi\rangle \leq a$.

No: For every n-qubit weight-k quantum state $|\psi\rangle$, $\langle\psi|H|\psi\rangle \geq b$.

Proposition 3.13. WEIGHT-k l-LOCAL HAMILTONIAN *is in* XP.

Proof. Let $S_{n,k}$ denote the set of all n-bit strings with Hamming weight k and let ε be an enumeration of the elements of $S_{n,k}$. We define the matrix H_ε such that $\langle\varepsilon(x)|H_\varepsilon|\varepsilon(y)\rangle \coloneqq \langle x|H|y\rangle$ for all $x, y \in S_{n,k}$, and for an n-qubit weight-k quantum state $|\psi\rangle \coloneqq \sum_{x\in S_{n,k}} \alpha_x |x\rangle$, we define the quantum state $|\psi_\varepsilon\rangle \coloneqq \sum_{x\in S_{n,k}} \alpha_x |\varepsilon(x)\rangle$. Then, for any n-qubit weight-k quantum state $|\psi\rangle$, we have $\langle\psi|H|\psi\rangle = \langle\psi_\varepsilon|H_\varepsilon|\psi_\varepsilon\rangle$. Therefore, it is sufficient to compute the smallest eigenvalue $\lambda_{\min}(H_\varepsilon)$ of H_ε. However, since the dimension of H_ε is $n^{O(k)}$ and each of its entries can be computed in time $n^{O(1)}$, we can compute $\lambda_{\min}(H_\varepsilon)$ in time $n^{O(k)}$. Hence, WEIGHT-k l-LOCAL HAMILTONIAN is in XP. This completes the proof. ∎

Inside QW[P] there lies a whole hierarchy of intractable classes known as the QW-hierarchy. First, we define the weft of a circuit and then we define a corresponding circuit satisfiability problem which we use to define each rung in the QW-hierarchy.

Definition 3.14 (Quantum circuit weft). Given a quantum circuit C comprising generalised Toffoli gates, one and two-qubit gates, and unbounded classical fan-out. The *weft* of C is the maximum number of Toffoli gates that act on any path from input qubit to output qubit.

We remark that the fanout gate allowed in a weft-1 quantum circuit is classical. In a quantum circuit, a fanout gate is called classical if all of the target qubits are initialized to the $|0\rangle$ state and no other gates acted on them before the fanout gate. After the fanout gate, a unitary gate can only act on the fanout qubits by using them as controls. The equivalence between this definition of classical fanout gates and the standard definition follows from the principle of delayed measurements. Because quantum fanout gates are very powerful and can simulate big Toffoli and threshold gates [HS05], they should be avoided when defining weft-t quantum circuits. In the classical case, there is no restriction in the use of fanout, if we allowed the same with the quantum fanout gate then we would increase the power of the constant depth quantum circuits without increasing the weft.

To define the QW-hierarchy we proceed similarly as in [MW05] for the class QMA. For functions $c, s : \mathbb{N} \to [0, 1]$ we define the following problem.

Definition 3.15 (WEIGHT-k WEFT-t DEPTH-d QUANTUM CIRCUIT SATISFIABILITY (c, s)).

Instance: A weft-t depth-d quantum circuit C on n witness qubits and $\mathrm{poly}(n)$ ancilla qubits.

Parameter: A natural number k.

Yes: There exists an n-qubit weight-k quantum state $|\psi\rangle$, such that $\Pr[C(|\psi\rangle) \text{ accepts}] \geq c$.

No: For every n-qubit weight-k quantum state $|\psi\rangle$, $\Pr[C(|\psi\rangle) \text{ accepts}] \leq s$.

Definition 3.16 ($\mathsf{QW}_{c,s}[t]$). For $t \in \mathbb{N}$, the class $\mathsf{QW}_{c,s}[t]$ consists of all parameterized problems that are FPQT reducible to WEIGHT-k WEFT-t DEPTH-d QUANTUM CIRCUIT SATISFIABILITY(c, s) for some constant depth $d \geq t$.

Due to the constant depth requirement of weft-t quantum circuits, it is not clear if this class has the error reduction property. This was not an issue for $\mathsf{QW}[\mathsf{P}]$, for example when proving Proposition 3.11, since the circuits involved are of polynomial size. Because of the constraint in the depth for QW circuits, we don't have access to techniques such as those in [MW05]. For this reason, the definition of $\mathsf{QW}_{c,s}[t]$ includes the completeness and soundness as part of its definition. These classes are most relevant when c and s have a polynomial gap, i.e., $c - s > 1/\text{poly}(n)$. Based on this, we define the QW-hierarchy as

Definition 3.17. Define $\mathsf{QW}[t]$ as

$$\mathsf{QW}[t] := \bigcup_{\substack{c,s \\ c-s>1/\text{poly}(n)}} \mathsf{QW}_{c,s}[t].$$

The following containments are straightforward.

Proposition 3.18. *For any $t \in \mathbb{N}$ and any $a, b : \mathbb{N} \to [0, 1]$, $\mathsf{W}[t] \subseteq \mathsf{QW}_{a,b}[t]$, $\mathsf{QW}_{a,b}[t] \subseteq \mathsf{QW}_{a,b}[t + 1]$*

Proof. To prove the first part, note that $\mathsf{W}[t] \subseteq \mathsf{QW}_{0,1}[t]$. Moreover, note that if $\forall n \in \mathbb{N}$, $b(n) \geq b'(n)$ and $a(n) \leq a'(n)$ then $\mathsf{QW}_{a,b}[t] \subseteq \mathsf{QW}_{a',b'}[t]$. We conclude that $\mathsf{W}[t] \subseteq \mathsf{QW}_{0,1}[t] \subseteq \mathsf{QW}_{a,b}[t]$. The second part is trivial. ∎

Proposition 3.19. *For any $t \in \mathbb{N}$ and $a, b : \mathbb{N} \to [0, 1]$ such that $b(n) \geq \frac{1}{2} + \frac{1}{\text{poly}(n)}$, $a(n) \leq \frac{1}{2} - \frac{1}{\text{poly}(n)}$ we have that $\mathsf{QW}_{a,b}[t] \subseteq \mathsf{QW}[\mathsf{P}]$.*

The complexity classes $\mathsf{QW}_{a,b}[t]$, for $t \geq 1$ and fixed $a, b : \mathbb{N} \to [0, 1]$, define the $\mathsf{QW}_{a,b}$ hierarchy, while note that $\mathsf{FPQT} \subseteq \mathsf{QW}_{a,b}[0]$. We prove the following.

Proposition 3.20. *For any $t \in \mathbb{N}$ and any $a, b : \mathbb{N} \to [0, 1]$, if $\mathsf{W}[t] = \mathsf{QW}_{a,b}[t]$ then $\mathsf{P} = \mathsf{BQP}$.*

Proof. If $\mathsf{W}[t] = \mathsf{QW}_{a,b}[t]$ for some $a, b : \mathbb{N} \to [0,1]$ then $\mathsf{FPQT} \subseteq \mathsf{QW}_{a,b}[t] = W[t] \subseteq \mathsf{XP}$, and so $\mathsf{P} = \mathsf{BQP}$ by Proposition 3.6. ∎

Proposition 3.21. *For any $t \in \mathbb{N}$ and any $a, b : \mathbb{N} \to [0,1]$, if $\mathsf{QW}_{a,b}[t] \subseteq \mathsf{XP}$ then $\mathsf{P} = \mathsf{BQP}$.*

Proof. If $\mathsf{QW}_{a,b}[t] \subseteq \mathsf{XP}$ then $\mathsf{FPQT} \subseteq \mathsf{XP}$, and so $\mathsf{P} = \mathsf{BQP}$ by Proposition 3.6. ∎

While it was possible to find complete problems for $\mathsf{QW}[\mathsf{P}]$, it is much harder to do the same for $\mathsf{QW}[1]$ due to the fact that its definition includes constant-weft circuits. In the next section we prove that the Subset state problem defined in [BGKS16] is $\mathsf{QW}[\mathsf{P}]$ complete, but the issue with success amplification (or equivalently, error reduction) in $\mathsf{QW}[t]$ doesn't allow us to prove completeness for $\mathsf{QW}[1]$. Note that if we tried to apply some technique such as the Kitaev clock construction to prove that the WEIGHT-k ℓ-LOCAL HAMILTONIAN problem is $\mathsf{QW}[1]$-hard, we would have the issue that the history state does not have in general weight k. This is because the constant depth circuit we are reducing from does not preserve weight in general.

3.3.1 Subset state problem is $\mathsf{QW}[\mathsf{P}]$-complete

Here we use results from [BGKS16] to prove that the witness for the $\mathsf{QW}[\mathsf{P}]$ class is only required to be a uniform superposition of weight-k states. The proof follows the idea of [BGKS16] by modifying the so called geometric Lemma used in that reference.

Definition 3.22. Let $S \subseteq [d]$. We denote $|S\rangle \in \mathbb{C}^d$ as the state

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle \tag{3.1}$$

When working with qubits, we can identify $i \in S$ with its binary description. We say that $|S\rangle$ is a subset state of weight k when it can be written as a uniform superposition of computational states of weight k.

Lemma 3.23 (geometric Lemma from Grilo et al. [BGKS16]). *For a vector $v \in \mathbb{C}^d$, there exists a subset $S \subseteq [d]$ such that*

$$\frac{1}{\sqrt{|S|}}\left|\sum_{j\in S} v_j\right| \geq \frac{\|v\|}{8\sqrt{\log_2(d)+3}} \tag{3.2}$$

Lemma 3.24. *For a vector $v \in \mathbb{C}^d$ with weight k, there exists a subset $S \subseteq [d]$ such that $|S\rangle$ is weight-k and*

$$\frac{1}{\sqrt{|S|}}\left|\sum_{j\in S} v_j\right| \geq \frac{\|v\|}{8\sqrt{\log_2(d)+3}} \tag{3.3}$$

Proof. Consider the vector space $\mathcal{S}_k \subseteq \mathbb{C}^d$ spanned by the computational basis states of weight k. Consider the isomorhism $T : \mathcal{S}_k \to \mathbb{C}^{\binom{d}{k}}$ which maps the weight k basis states to the canonical orthonormal basis of $\mathbb{C}^k$. Then we apply Lemma 3.23, which gives a subset of the weight-k states such that equation 3.2 is fullfilled. Using T^{-1} we can map back to $\mathcal{S}_k \subseteq \mathbb{C}^d$ which gives the result. ∎

Definition 3.25 (WEIGHT-k BASIS STATE CHECK (WEIGHT-k BSC(α))).

Instance: A classical description x of a unitary V_x implemented by a quantum circuit acting on $m(n)$ qubits and $a(n)$ ancilla qubits, where $n = |x|$ and $m(n)$, $a(n)$ polynomials. A $m'(n)$-bitstring y such that $m'(n) \leq m(n) + a(n)$.

Parameter: A natural number k.

Yes: $\exists S \subseteq [2^{m(n)}]$ of weight-k, $\left\|(\langle y| \otimes I)V_x |S\rangle |0\rangle^{\otimes a(n)}\right\|^2 \geq 1 - \alpha$.

No: $\forall S \subseteq [2^{m(n)}]$ of weight-k, $\left\|(\langle y| \otimes I)V_x |S\rangle |0\rangle^{\otimes a(n)}\right\|^2 \leq \alpha$.

Proposition 3.26. WEIGHT-k BSC(α) *is* QW[P]*-complete for $\alpha = 2^{-r(n)}$ where $r(n)$ is a polynomial.*

Proof. The proof largely follows the one from [BGKS16]. First we prove that WEIGHT-k BSC(α) is in QW[P] by showing that it reduces to WEIGHT-k QUANTUM CIRCUIT

SATISFIABILITY. Let $|\psi\rangle$ be a state with weight k or less. When the verifier circuit receives the witness, it applies V_x and measures in the computational basis to check whether the bitstring y was obtained. Assume we are given a yes-instance, then there is a weight-k state $|S\rangle$ such that the verifier circuit accepts with probability at least $1-\alpha$. If we are given a no-instance, then for all $S \subseteq [2^{m(n)}]$ of weight k or less, then

$$\left\| (\langle y| \otimes I) V_x |S\rangle |0\rangle^{\otimes a(n)} \right\|^2 \leq \alpha$$

We now prove that the probability of acceptance is low for all $|\psi\rangle$ of weight k. By virtue of Lemma 3.24, there is $|S\rangle$ of weight k such that $|\langle\psi|S\rangle| \geq \frac{1}{8\sqrt{m(n)+3}}$. Define

$$O \coloneqq \mathrm{Tr}_A \left[\left(I \otimes |0\rangle\langle 0|^{\otimes a(n)} \right) V_x^\dagger \left(|y\rangle\langle y| \otimes I \right) V_x \left(I \otimes |0\rangle\langle 0|^{\otimes a(n)} \right) \right]. \tag{3.4}$$

We want to bound the expression $\langle\psi|O|\psi\rangle$ which gives the probability of acceptance for witness $|\psi\rangle$. Note that

$$\begin{aligned}
\langle S|O|S\rangle - \langle\psi|O|\psi\rangle &\leq \max_{0\leq C\leq I} |\langle S|C|S\rangle - \langle\psi|C|\psi\rangle| && (3.5)\\
&= \frac{1}{2} \| |S\rangle\langle S| - |\psi\rangle\langle\psi| \|_{tr} && (3.6)\\
&= \frac{1}{2}(2 - |\langle\psi|S\rangle|^2) && (3.7)\\
&\leq 1 - \frac{1}{128(m(n)+3)}. && (3.8)
\end{aligned}$$

Thus,

$$\begin{aligned}
\langle\psi|O|\psi\rangle &= \langle S|O|S\rangle - (\langle S|O|S\rangle - \langle\psi|O|\psi\rangle) && (3.9)\\
&\leq \alpha + \left(1 - \frac{1}{128(m(n)+3)}\right). && (3.10)
\end{aligned}$$

Thus showing the gap is proportional to the inverse of a polynomial. To prove completeness we reduce from WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY. Let C be a quantum circuit and $\{C, I-C\}$ the POVM such that either there is a weight-k witness state such that C accepts with probability at least $\langle\psi|C|\psi\rangle \geq 1 - 2^{-r(n)}$ with r

a polynomial, or for all weight-k witness states the circuit accepts with probability at most $\langle\psi|C|\psi\rangle \leq 2^{-r(n)}$. This reduces to the WEIGHT-k BSC(α) problem by choosing $y = 1$ and $m'(n) = 1$.

More specifically, assume we are given a no instance of WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY, then clearly this corresponds to a no instance of WEIGHT-k BSC(α). Now suppose we are given a yes instance and let $|\psi\rangle$ be the witness that maximizes the acceptance probability of C. Then by Lemma 3.24, there is a weight-k subset state $|S\rangle$ such that

$$|\langle\psi|S\rangle| \geq \frac{1}{8\sqrt{n+3}}. \tag{3.11}$$

Next, note that

$$\begin{aligned}
\langle S|C|S\rangle - \langle\psi|C|\psi\rangle &\leq \max_{0\leq C\leq I} |\langle S|C|S\rangle - \langle\psi|C|\psi\rangle| && (3.12)\\
&= \frac{1}{2}\||\psi\rangle\langle\psi| - |S\rangle\langle S|\| && (3.13)\\
&= 2\sqrt{1 - |\langle\psi|S\rangle|^2} && (3.14)\\
&\leq 2 - |\langle\psi|S\rangle|^2, && (3.15)
\end{aligned}$$

where following [BGKS16] we use the inequality $\sqrt{1-x^2} \leq 1 - x^2/2$. Thus

$$\langle S|C|S\rangle \geq \frac{1}{128(n+3)} - 2^{-r(n)}. \tag{3.16}$$

Using probability amplification we can prove the result. ■

Since we lack probability amplification for $\mathsf{QW}[1]$, we cannot prove an analogous result for this class. Instead, by using the same structure of the previous proof we obtain the following.

Definition 3.27 (WEIGHT-k WEFT-t DEPTH-d BASIS STATE CHECK (WEIGHT-k WEFT-t DEPTH-d BSC(α))).

Instance: A classical description x of a unitary V_x implemented by a quantum circuit of constant depth and weft t acting on $m(n)$ qubits and $a(n)$ ancilla qubits, where

$n = |x|$ and $m(n)$, $a(n)$ polynomials. A $m'(n)$-bitstring y such that $m'(n) \leq m(n) + a(n)$.

Parameter: A natural number k.

Yes: $\exists S \subseteq [2^{m(n)}]$ of weight-k, $\left\| (\langle y| \otimes I) V_x |S\rangle |0\rangle^{\otimes a(n)} \right\|^2 \geq 1 - \alpha$.

No: $\forall S \subseteq [2^{m(n)}]$ of weight-k, $\left\| (\langle y| \otimes I) V_x |S\rangle |0\rangle^{\otimes a(n)} \right\|^2 \leq \alpha$.

We can directly apply the techniques in proof from Proposition 3.26 to prove some results on the WEIGHT-k WEFT-t DEPTH-d BSC(α) problem and the QW-hierarchy. In particular the previous proof shows the following two propositions.

Proposition 3.28. WEIGHT-k WEFT-t DEPTH-d BSC(α) *is in* $\mathsf{QW}_{\alpha,1-\alpha}[t]$ *for* $\alpha = 2^{-r(n)}$ *with* $r(n)$ *a polynomial.*

Proposition 3.29. WEIGHT-k WEFT-t DEPTH-d QUANTUM CIRCUIT SATISFIABILITY$(\alpha, 1-\alpha)$ *reduces to* WEIGHT-k WEFT-t DEPTH-d BSC$(\alpha, \frac{1}{128(n+3)} - \alpha)$ *for* $\alpha = 2^{-r(n)}$ *with* $r(n)$ *a polynomial.*

We see from the previous proof and the propositions that the main obstacle for proving completeness is the lack of an amplification procedure for the QW-hierarchy. In particular, notice that the reduction in Proposition 3.29 includes specific parameters for α which doesn't translate into a completeness for the whole class.

3.4 Summary

In this chapter we have given the definitions for some basic complexity class in quantum parameterized complexity. While the definitions of FPQT and QW[P] were direct from the classical case, we have seen that the QW-hierarchy has some issues regarding error reduction which has implications in finding natural complete problems for QW[1]. One of the motivations for the work presented in Chapter 4 is that of searching for such QW[1]-complete problem.

Chapter 4

Parameterized Complexity of the Weighted Local Hamiltonian Problem

This chapter is based on work in [BJL+22]. This is work done in collaboration with Michael Bremner, Zhengfeng Ji, Xingjian Li and Luke Mathieson. Our work has been presented in The Theory of Quantum Computation, Communication and Cryptography conference (TQC) 2023. We present a parameterized version of the local Hamiltonian and connect the intractability of this problem to a quantum version of the exponential time hypothesis, moreover we show that this problem is QW[1]-complete. The writing in this chapter follows closely [BJL+22]. My contribution in this work consisted in contributing ideas, writing and in the proof of key results. I contributed to most of the writing which comprises Section 4.3.1, except Lemma 4.12. I also contributed in the writing of the rest of sections of this chapter, where I didn't contribute, I have rewritten those parts or added some detail such as in Section 4.3.3 (error reduction section) and some parts on the clock construction in Section 4.3.4. Section 4.3.5 was originally written by my coauthors in the paper, I mainly participated of the discussions for this section and contributed to the proof. I have included Section 4.3.5 in my thesis for completeness. The figures included in this chapter are those of [BJL+22], the original

versions were designed by me (except Figs. 4.4 and 4.5) and were later implemented in Tikz by my coauthors.

4.1 Introduction

Having laid out some of the main classes in quantum parameterized complexity in Chapter 3, we now focus on the complexity of a parameterized version of the Local Hamiltonian problem. The Local Hamiltonian problem is of fundamental importance in quantum complexity theory (see Section 2.3.2). In recent years, there has been some work on modified versions of the Local Hamiltonian which attempt to bring closer problems that quantum chemists or physicists have to tackle in their fields. In [OIWF22], the authors establish the QMA-completeness of a variant of the local Hamiltonian problem considering a fixed basis describing the orbitals of the electronic structure problem, inspired by the problem posed in [WLAG13]. Another work in this direction is that of [GL21], where the authors study the so called GUIDED LOCAL HAMILTONIAN PROBLEM in which the instance description includes a local Hamiltonian H and a state vector u promised to be close to the ground state of H. In this work it is shown that when the Hamiltonian is 6-local then the decision problem is BQP-hard, further work [GHLM22, CFW22] has shown that the problem remains BQP-hard when considering 2-local Hamiltonians.

Central to our work is the weighted version of the local Hamiltonian problem which we prove is in QW[1]. As is mentioned in Chapter 3, this problem is in XP, which is in stark contrast to the WEIGHT-k QUANTUM CIRCUIT SATISFIABILITY problem which is QW[P]-complete and hence cannot be in XP unless P = BQP. By proving that the weighted local Hamiltonian problem is in QW[1] we demonstrate a likely separation between this problem and other parameterized variants of QMA-complete problems such as Quantum Circuit Satisfiability under FPQT reductions.

In our work we link the complexity of the WEIGHT-k ℓ-LOCAL HAMILTONIAN problem to the classical ETH and quantum variants of it, QETH and QCETH. It is

shown that if the WEIGHT-k ℓ-LOCAL HAMILTONIAN problem can be solved in FPT or FPQT (the quantum generalization of FPT introduced in [BJM+22]) then versions of these hypotheses will fail. The weight in this problem refers to the Hamming weight of the states in the promise of the local Hamiltonian problem, either there is a weight-k state with a small eigenvalue, or all weight-k states are above a certain energy. The restriction of the weight on the states considered in the problem finds a physical interpretation when considering the 1s in the computational basis as particle excitations and thus the weight corresponds to fixing the particle number to k.

By connecting the complexity of the WEIGHT-k ℓ-LOCAL HAMILTONIAN to QETH and QCETH we are also giving evidence of the unlikely tractability of QW[1] and of the whole QW-hierarchy. On the other hand, this provides a new way to disprove QETH if the reader does not believe it to be true.

4.2 Preliminaries

In this section we present the computational problems that we will be dealing with in this chapter and also define the QETH and QCETH.

We will reduce the WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b) problem to WEIGHT-k WEFT-t DEPTH-d QUANTUM CIRCUIT SATISFIABILITY (c, s) and show inclusion in QW[1]. For this purpose, we first reduce the WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b) problem to a weight-preserving version of the circuit satisfiability problem. We define a weight-preserving quantum circuit as a quantum circuit that when given an input of some weight k, the output is also of weight-k.

Definition 4.1 (Weight-k Weight-Preserving Quantum Circuit Satisfiability(c, s)).

Instance: A weight-preserving quantum circuit C on n witness qubits, poly(n) ancilla qubits with circuit size poly(n).

Parameter: A natural number k.

Yes: There exists an n-qubit weight-k quantum state $|\psi\rangle$, such that

$$\Pr[C(|\psi\rangle) \text{ accepts}] \geq c.$$

No: For every n-qubit weight-k quantum state $|\psi\rangle$,

$$\Pr[C(|\psi\rangle) \text{ accepts}] \leq s.$$

To show the likely intractability of the local Hamiltonian problem, we introduce the following quantum generalizations of ETH.

Definition 4.2 (Quantum Exponential Time Hypothesis). We define the QETH as follows. For some c, s with $c - s > 1/\text{poly}(n)$, there is no *quantum* algorithm running in time $2^{o(n)}$ that decides for a weft-1 quantum circuit Q of total description size n whether (i) there is an input witness state $|\psi\rangle$ such that $\Pr(Q(|\psi\rangle) \text{ accepts}) \geq c$ or (ii) for all input witness states $|\psi\rangle$, $\Pr(Q(|\psi\rangle) \text{ accepts}) \leq s$, given the promise that one of the two holds.

Definition 4.3 (Quantum-Classical Exponential Time Hypothesis). We define the QCETH as follows. There is no *quantum* algorithm running in time $2^{o(n)}$ that decides for a weft-1 Boolean circuit C of total description size n, whether there is an input vector x such that $C(x) = 1$.

We have defined QETH as a hypothesis about *some* pair c, s with polynomial gap rather than all such pairs c, s. The reason for this choice is that we want to show that if certain problems are tractable given any inverse polynomial gap, then QETH is false. This will be evident later in this chapter.

4.3 Weighted Local Hamiltonian is in QW[1]

In this section we prove that the weighted version of the Local Hamiltonian problem is in the class QW[1]. We state this as a theorem.

Theorem 4.4. *Given a, b such that $b - a > 1/\mathrm{poly}(n)$, then* WEIGHT-$k$ ℓ-LOCAL HAMILTONIAN(a, b) *is in* $\mathsf{QW}_{c,s}[1]$ *for some c, s such that $c - s > 1/\mathrm{poly}(n)$.*

The proof of Theorem 4.4 consists of a series of reductions. In the **first step**, we reduce the weighted local Hamiltonian problem to a weight-preserving quantum circuit satisfiability problem defined below. This step is discussed in Section 4.3.2. In the **second step** (Section 4.3.3), we prove that strong completeness and soundness error reduction is also possible for the weight-preserving circuits using the quantum singular value transformation. This step is necessary for the reductions in the later steps. In the **third step** we reduce the weight-preserving quantum circuit satisfiability problem to instances of the Local Hamiltonian problem that are *almost spatially sparse*. This notion will be defined below in Section 4.3.4 of the proof of Theorem 4.4. Finally in the **fourth step** (Section 4.3.5), we reduce the weighted almost spatially sparse Hamiltonian to an instance of the weighted constant-depth, weft-1, quantum circuit satisfiability problem. Before proceeding to the proof of these reductions, we prove some preliminary results about weight-preserving quantum circuits which we will require later.

4.3.1 Universality of Weight-Preserving Circuits

In this section, we will show how the classic proof of quantum universality in [BBC+95] can be adapted to show universality of weight-preserving circuits.

Definition 4.5. An operator O acting on $(\mathbb{C}^2)^{\otimes n}$ is weight-preserving if for any k and any computational basis state $|x\rangle$ of weight k, $O|x\rangle$ is a vector in $(\mathbb{C}^2)^{\otimes n}$ of weight exactly k.

Definition 4.6. A circuit C is *weight-preserving* if its corresponding unitary operator is weight-preserving.

We also define the weight-preserving version of one-qubit gates.

Definition 4.7. For any single qubit gate U, define a two-qubit gate

$$\hat{U} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

It is easy to check that $\hat{U}$ is always a weight-preserving gate. Note that When $U = X$, $\hat{U}$ is the SWAP gate, this fact will be used regularly below.

The Fredkin gate (control-SWAP gate) is another example of weight-preserving gate. We will also need the following weight-preserving gate.

$$E = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix}.$$

This phase gate is necessary for universality as otherwise we will not be able to create relative phases between states such as $|00\rangle$ and $|11\rangle$.

Definition 4.8. A set of weight-preserving gates is *weight-universal* if they can (approximately) generate all weight-preserving unitary transformations.

Lemma 4.9. *If a set of single-qubit gates $U_1, U_2, \ldots, U_s$ and CNOTs form a standard universal gate set, then $\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_s$, Fredkin and E gates form a weight-universal gate set when allowed two extra ancilla qubits in the state $|01\rangle$.*

Proof. We follow the steps of [NC00, Chapt. 4]. In this proof the first step is to show that two-level unitary gates are universal and can generate any $d \times d$ unitary from the group $U(d)$. Recall that two-level unitaries are gates which only act on the subspace spanned by two computational basis state, for example for $d = 3$ a two-level unitary could be

$$\begin{pmatrix} a & 0 & b \\ 0 & 1 & 0 \\ c & 0 & d \end{pmatrix}.$$

The authors prove that $d \times d$ unitaries can be obtained using $d(d-1)/2$ two level unitaries. In our case we simply need to recognize that this proof will hold in any chosen weight-k subspace. Hence we can always use the same inductive steps as those in [NC00, Sec. 4.5.1] where non-trivial unitaries are limited to this subspace. This requires at most $\binom{n}{k}\left(\binom{n}{k}-1\right)/2$.

Then, by following the proof in [NC00, Sec. 4.5.2] it can be shown that if we can implement all $\hat{U}$ operators (where U is a single qubit gate), E, and Fredkin, then we can implement any two-level unitary.

Recall that in [NC00, Sec. 4.5.2] the authors use the Gray code, which given two bitstrings generates a sequence of strings that differ by a single bit. That is the hamming weight changes by one in each step of the sequence. This sequence is used to generate a circuit of multiply-controlled single-qubit gates to define an arbitrary two-level unitary.

In our case, we cannot use this construction as it is not weight-preserving. However, note that we have the Fredkin gate in our gate set, which allows controlled swaps, and also note that we are operating in a weight-preserving space. Hence, we only need a sequence of operations that controllably swap qubits in this space and then will ultimately perform $\hat{U}$ gate. Suppose we want to implement a two level operator in the subspace of $|s\rangle = |10001\rangle$ and $|t\rangle = |11000\rangle$. We can consider the following transformations $10001 \longrightarrow 10100 \longrightarrow 11000$. Essentially, we want to place $(k-1)$ of the 1's from $|s\rangle$ in the same positions of $(k-1)$ 1's in $|t\rangle$. The remaining non-swapped 1 of $|s\rangle$ is placed in a position next to the remaining 1 in $|t\rangle$, for instance in the previous example we performed the transformation $10001 \longrightarrow 10100$ placing the last 1 in the third position, next to the second position where the last 1 of $|t\rangle$ is located. This can be implemented in the same way as in [NC00, Sec. 4.5.2] with the difference that now we apply controlled SWAP operators controlled on the rest of the qubits, see Fig. 4.1. Finally the operator $\hat{U}$ acts on qubits 2 and 3 (corresponding to the second and third bits from left to right). This operator is controlled on the rest of the qubits and finally we revert the SWAP operations. For weight-k states we will require at most $2k$ SWAP gates plus the controlled $\hat{U}$.

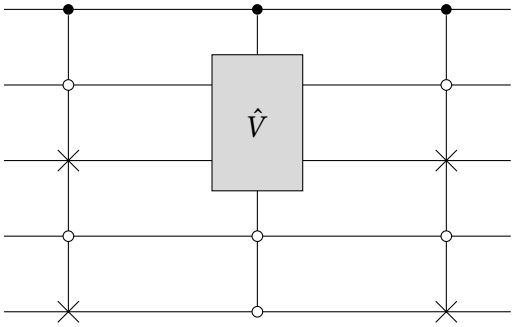

Figure 4.1: Circuit implementing a two-level unitary between states $|s\rangle = |10001\rangle$ and $|t\rangle = |11000\rangle$. The transformation represented by the controlled SWAP gates is $10001 \longrightarrow 10100$. The controlled $\hat{V}$ gate implements the two-level transformation in the subspace spanned by $|s\rangle$ and $|t\rangle$. The black dots denote the control operations activated if the qubit is in the state $|1\rangle$ and white dots denote controls activated when the qubit is in state $|0\rangle$. The crosses indicate SWAP operations.

We now show that we can implement weight-1 two-qubit gates $\hat{V}$ with multiple controls using only weight-1 two-qubit gates, the Fredkin, and E gates. We follow the technique employed in [BBC+95] to prove this. First, by Lemma 5.1 of [BBC+95], it's known that a controlled version of $W \in \mathrm{SU}(2)$ can be implemented by considering $A, B, C \in \mathrm{SU}(2)$ such that $ABC = I$ and $AXBXC = W$. Directly employing the same decomposition in the case where $\hat{W}$ is a weight-1 two-qubit gate, by noting that $\hat{A}\hat{B}\hat{C} = I$ and $\hat{A}(\mathrm{SWAP})\hat{B}(\mathrm{SWAP})\hat{C} = \hat{W}$. Note that in our case the CNOT gates become Fredkin gates. To implement a single control version of $W \in \mathrm{U}(2)$, a controlled phase gate is included, in the weight-preserving case we use the gate E.

To construct a multiple controlled version of a unitary $\hat{W}$ with $W \in \mathrm{U}(2)$, consider the construction from Lemma 6.1 in [BBC+95]. We can create a weight-preserving version of this construction as in Fig. 4.2, which includes two ancilla qubits set to $|0\rangle\,|1\rangle$ and requires finding $\hat{V}$ such that $\hat{V}^2 = \hat{W}$. These qubits can be reused for each gate we want to construct and thus only increases the weight of all the qubits in 1. The intuition behind the circuit is that we use the $|01\rangle$ ancilla to decide if we should apply the controlled $V^\dagger$, since in the original construction there are two CNOTs, we can replace them with SWAPs and the ancilla system. When considering more control qubits, the construction generalizes in the same way, by considering more Fredkin gates acting

on ancillas instead of CNOTs. Note that if we want the controls to be activated by $|0\rangle$ instead of $|1\rangle$, we can simply introduce SWAPs in the ancilla system. With these considerations we can implement any two-level unitary constructed from circuits such as the one in Fig. 4.1 using only weight-1 two qubit gates, the Fredkin gate, and E. If we want to use the discrete set $\hat{U}_1, \cdots, \hat{U}_s$ instead of all weight-1 preserving two qubit gates, then the Solovay-Kitaev theorem applies in this case and thus proves the result. ■

Remark 4.10. The proof above shows that to implement a two-level unitary over the weight-k subspace requires $O(2^n)$ gates from our weight-universal gate set. This exponential comes mainly from the implementation we used for the controlled $\hat{W}$ gate. For our work in this chapter, this exponential dependence is sufficient. We remark that a more efficient construction is possible, with caveat that it includes non-trivial operations outside the weight-k subspace which might be of interest to some readers. In [BBC+95] a more efficient construction is offered which scales like $O(n^2)$. We can adapt our proof to improve the scaling in the same way provided that we don't care how the two-level unitary acts outside the weight-k subspace of dimension 2. This improvement is obtained by noticing that circuits implementing two-level unitaries as in Fig. 4.1 only require k controls since we need to check the position of the 1's. This will imply that outside the weight-k subspace the action of the unitary will be non-trivial, but if we only care about this subspace, then the dependence will be on k rather than n for implementing them. Even more improvements can be obtained using the techniques from Lemma 7.2 and Lemma 7.3 in [BBC+95].

Remark 4.11. Note that when initializing the ancilla qubits of the weight-preserving circuits we will construct in the reductions, we can set at most $f(k)$ of them to $|1\rangle$, where f is some computable function. This guarantees our reduction still contained in FPQT.

The following lemmas will be necessary for our proof of Theorem 4.4.

Lemma 4.12. *Let $n = 2^r$ be an integer power of 2. The W state*

$$|W_n\rangle = \frac{1}{\sqrt{n}} \left(|10\cdots0\rangle + |01\cdots0\rangle + \cdots + |00\cdots1\rangle \right)$$

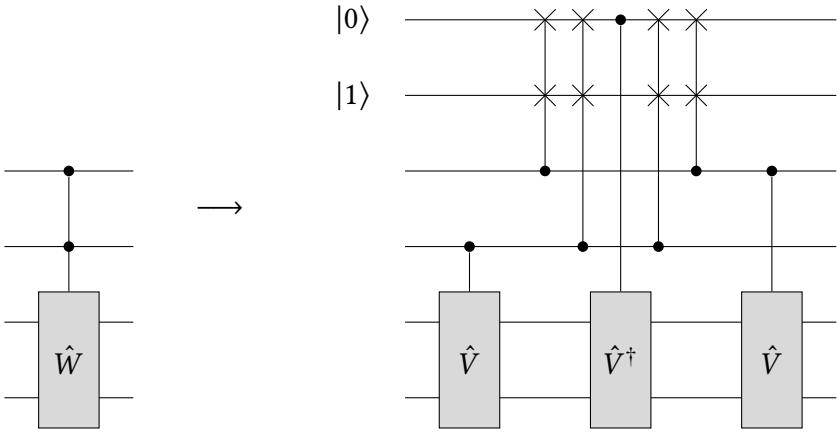

Figure 4.2: Circuit implementing a controlled version of $\hat{W}$ with two controls. This requires two ancillas initiated in the state $|01\rangle$ and can be reused in the construction of other gates. In this circuit $\hat{V}^2 = \hat{W}$.

of n qubits can be computed from $\left|0^{n-1}1\right\rangle$ *by a weight-preserving quantum circuit efficiently.*

Proof. We prove by induction on r that there is such circuits C_n such that $C_n |0^n\rangle = |0^n\rangle$ and $C_n \left|0^{n-1}1\right\rangle = |W_n\rangle$. First for $r = 1$, the result follows by applying the gate

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.1}$$

Assume the claim is proved for $n = 2^{r-1}$ and we shall show the same for $n' = 2^r$. Notice that

$$|W_{n'}\rangle = \frac{1}{\sqrt{2}}\big(|W_n\rangle |0^n\rangle + |0^n\rangle |W_n\rangle\big),$$

which can be prepared by first apply the gate in Eq. (4.1) to the $n + 1$ and the last qubit followed by two C_{n-1} circuits acting on the first and second half of the qubits. ∎

4.3.2 Weight-Preserving Quantum Circuit Satisfiability

In this section, we construct a weight-preserving verification circuit from the local Hamiltonian problem. We emphasize that the Hamiltonian does *not* need to be weight-

preserving and that the resulting circuit is not of constant depth yet. The following lemma summarizes the main result of this section.

Lemma 4.13. *Given a weight-k ℓ-local Hamiltonian problem $H = \sum_{j=1}^{m} H_j$ of m terms on n qubits and energy bounds a and b with gap $b - a > 1/\mathrm{poly}(n)$. Suppose also that $\|H_j\| \leq 1$ for all $j = 1, 2, \ldots, m$. Then there is a weight-preserving circuit W_H of $\mathrm{poly}(n)$ size on $n + M + k + 2$ qubits that accepts with probability*

$$1 - \frac{m + \langle\psi| H |\psi\rangle}{2M}$$

where $|\psi\rangle$ is the input witness state and $M = 2^{\lceil \log_2 m \rceil}$, the smallest integer power of 2 larger than m.

Proof of Lemma 4.13. We use $P_m^{(k)}$ to denote the projector onto the subspace of weight-k basis states of length m. By convention, If $k > m$ then $P_m^{(k)}$ is the zero operator. We first show how we can implement a weight-preserving unitary circuit that accepts with probability $\langle\psi| (I - H_j) |\psi\rangle /2$. Assume for simplicity that the term H_j acts on the first ℓ qubits and let $O = (I - H_j)/2$ be a positive semi-definite operator. We are interested in the quantity $\langle\psi| O |\psi\rangle$ and we claim the following identity

$$\langle\psi| O \otimes I_{n-l} |\psi\rangle = \sum_{w=0}^{l'} \langle\psi| O^{(w)} \otimes P_{n-l}^{(k-w)} |\psi\rangle$$

for state $|\psi\rangle$ of weight k where $O^{(w)} = P_l^{(w)} O P_l^{(w)}, l' = \min(k, l)$. This follows by computing the matrix entries of $O \otimes I_{n-l}$ with indices i, i' of weight k. Alternatively, one can see that

$$\begin{aligned}
\langle\psi|O \otimes I|\psi\rangle &= \langle\psi| \left(\sum_{w=0}^{l'} P_l^{(w)} \otimes P_{n-l}^{(k-w)} \right) O \otimes I \left(\sum_{w'=0}^{l'} P_l^{(w')} \otimes P_{n-l}^{(k-w')} \right) |\psi\rangle \\
&= \langle\psi| \sum_{w=0}^{l'} P_l^{(w)} O P_l^{(w)} \otimes P_{n-l}^{(k-w)} |\psi\rangle \\
&= \langle\psi| \sum_{w=0}^{l'} O^{(w)} \otimes P_{n-l}^{(k-w)} |\psi\rangle
\end{aligned}$$

Now we introduce in the circuit we are building two ancilla qubits starting in state $|01\rangle$. Then the following matrix

$$U^{(w)} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & \sqrt{O^{(w)}} & \sqrt{I - O^{(w)}} & 0 \\ 0 & \sqrt{I - O^{(w)}} & -\sqrt{O^{(w)}} & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

is unitary and weight-preserving. It is unitary as $U^{(w)}\left(U^{(w)}\right)^{\dagger} = I$ follows by direct calculations. The weight-preserving property follows from the weight-preserving property of $O^{(w)}$, and therefore also $\sqrt{O^{(w)}}$ and $\sqrt{I - O^{(w)}}$. The ancilla qubits in the state $|01\rangle$ are chosen such that $U^{(w)}|\psi\rangle|01\rangle = \sqrt{O^{(w)}}|\psi\rangle|01\rangle + \sqrt{I - O^{(w)}}|\psi\rangle|10\rangle$. We want to act with $U^{(w)}$ conditioned on the remaining $n-l$ qubits having weight $k-w$. We can do this by adding $k+1$ ancillas in the state $|100\cdots0\rangle$ and then act on this ancilla registers with controlled gates that perform a cyclic shift of the registers controlled by the original $n-l$ qubits. We define S as the circular shift operator that act as $S|i_1 i_2 \ldots i_n\rangle = |i_n i_1 \ldots i_{n-1}\rangle$. Now, we define the circuit V_{weight} which will keep track of the weight of the $n-l$ qubits.

$$V_{\text{weight}} = \sum_{i=0}^{l'} P_{n-l}^{(k-i)} \otimes S^i.$$

We can easily construct the operator V_{weight} by using controlled versions of the operator S which are controlled by each of the $n-l$ qubits. Concretely, we consider $C^{(i)}S$ where S is controlled by qubit i. If the $n-l$ qubits are numbered $1, \cdots, n-l$ then just consider $\prod_{i=1}^{n-l} C^{(i)}S$. Once we have a register with the weight of the $n-l$ qubits, we can act with a controlled version of $U^{(w)}$ on the remaining qubits depending on the weight registered. The circuit is drawn in Fig. 4.3. If the remaining $n-l$ have weight $k-i$, then the $k+1$ ancillas gets rotated from $\left|10^k\right\rangle$ to $\left|0^i 10^{k-i}\right\rangle$. Let U be the circuit with V_{weight} and the controlled versions $U^{(w)}$ just described and consider the probability that we measure the first group of ancillary qubit in basis $|01\rangle$ after acting with U

$$\left\| (|01\rangle\langle 01| \otimes I)U|01\rangle|\psi\rangle\left|10^{l'}\right\rangle \right\|^2 = \left\| |01\rangle \otimes \left(\sum_{w=0}^{l'} \sqrt{O^{(w)}} \otimes P_{n-l}^{(k-w)} |\psi\rangle \otimes \left|0^w 10^{k-w}\right\rangle \right) \right\|^2$$

$$= \langle\psi| \sum_{w=0}^{l'} O^{(w)} \otimes P_{n-l}^{(k-w)} |\psi\rangle$$

$$= \langle\psi| O |\psi\rangle .$$

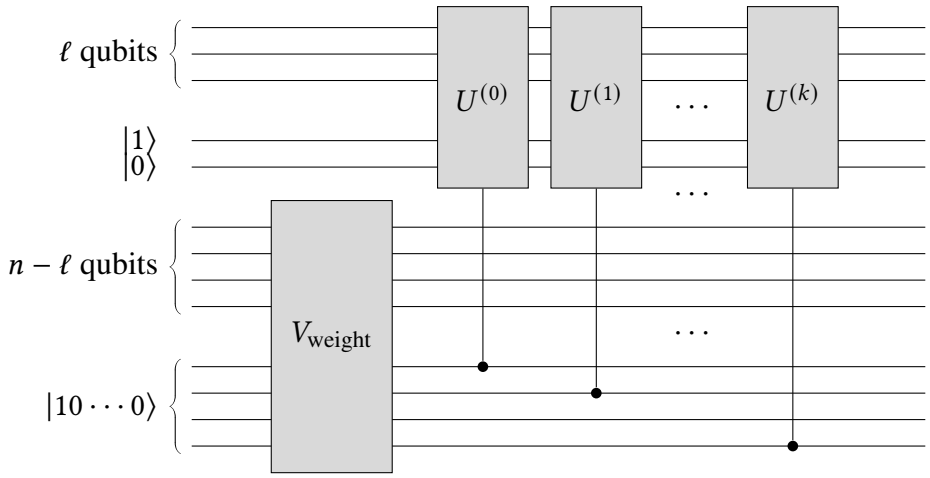

Figure 4.3: Circuit implementing the observable $O = (I - H_j)/2$ described in the text. The unitary V_{weight} writes the weight of the $n-l$ qubits on the counting registry $|10\cdots0\rangle$. The circuit acts on the ℓ qubits (and the pair of ancillas) depending on this weight.

We are now ready to construct the weight-preserving circuit for the local Hamiltonian H. We consider two registers of qubits, the first one selects a term H_i from the Hamiltonian. This register has $M = 2^{\lceil \log_2 m \rceil}$ qubits. The second register will contain n qubits for the witness state. The circuit is initiated by preparing the M-qubit state $|W\rangle$ in the first register for term selection. For all $j = 1, 2, \ldots, m$ and conditioned on the j-th qubit in the term selection register being in state $|1\rangle$, we perform the network of SWAP gates that moves the qubits that H_j acts on to the first ℓ qubits, apply the weight-preserving energy measurement circuit for $O = (I - H_j)/2$ as described above, note that the measurement performed depends on the chosen j as well. For all $j = m+1, \ldots, M$, the circuit accepts immediately.

It is easy to check that all gates used in the circuit are weight-preserving and the circuit accepts with probability

$$\Pr(\text{measuring } j \in \{m+1, \cdots, M\} \text{ in term register}) + \frac{1}{M} \sum_{j=1}^{m} \frac{1 - \langle\psi| H_j |\psi\rangle}{2}. \qquad (4.2)$$

This probability is equal to

$$\frac{M-m}{M}+\sum_{j=1}^{m}\frac{1-\langle\psi|H_j|\psi\rangle}{2M}=1-\frac{m+\langle\psi|H|\psi\rangle}{2M}.$$

■

4.3.3 Weight-Preserving Marriott-Watrous Amplification

In this section, we prove that it is possible to amplify the completeness and soundness gap for weight-preserving verification circuits with one copy of the witness state. This amplification will be seen to be necessary in the reduction. In the complexity class QMA, the verifier can always amplify the completeness-soundness gap as in the work of Marriott and Watrous [MW05], this procedure requires storing a polynomial number of measured bits. This last fact makes hard to implement the Marriott and Watrud procedure with weight-preserving circuit which work with Hilbert spaces of size n^k.

Since the procedure of Marriott and Watrous is not available and we can't simply repeat the circuit in parallel (repeating the circuit would increase the weight of the input state polynomially), then we use the fast QMA reduction in [NWZ09, Gil19]. The version of this amplification we use is based on the quantum singular value transform (QSVT).

Theorem 4.14. *Given a verifier circuit V for a language $L \in$ QMA with acceptance probability thresholds (a,b), we can construct a new verifier circuit V' with threshold $a' = \epsilon, b' = 1-\epsilon$ with one extra ancillary qubit, and $m = O\left(\frac{1}{\max\left[\sqrt{b}-\sqrt{a},\sqrt{1-a}-\sqrt{1-b}\right]}\log\left(\frac{1}{\epsilon}\right)\right)$ calls to V and $V^\dagger$ as in Fig. 4.4.*

To show how to implement this completeness-soundness gap amplification, we give a quick summary of the QSVT technique. Given a unitary V, define $\Phi = (\phi_1, \cdots, \phi_{2m}) \in \mathbb{R}^{2m}$ and the following circuit U_Φ:

$$U_\Phi = \prod_{j=1}^{n}\left(e^{i\phi_{2j-1}(2\Pi-I)}V^\dagger e^{i\phi_{2j}(2\tilde{\Pi}-I)}V\right), \tag{4.3}$$

where Π and $\tilde{\Pi}$ are both orthogonal projectors. By choosing the angles Φ appropriately one can implemement a a polynomial transformation on the singular values of V, this is shown in Theorem 2.3.7 of [Gil19]. The polynomial implemented is over the complex field. If we wish to implement a polynomial $P_{\mathbb{R}} \in \mathbb{R}$ over the real field then by Corollary 2.3.8 of [Gil19] we can use instead the circuit

$$P_{\mathbb{R}}(\tilde{\Pi}V\Pi) = (\langle +| \otimes \tilde{\Pi})(|0\rangle\langle 0| \otimes U_{\Phi} + |1\rangle\langle 1| \otimes U_{-\Phi})(|+\rangle \otimes \Pi)$$

To implement the controlled U_{Φ} in the previous formula, we only need to implement the gates $\sum_b |b\rangle\langle b| \otimes e^{i(-1)^b \phi(2\Pi - I)}$ as in Fig. 4.4.

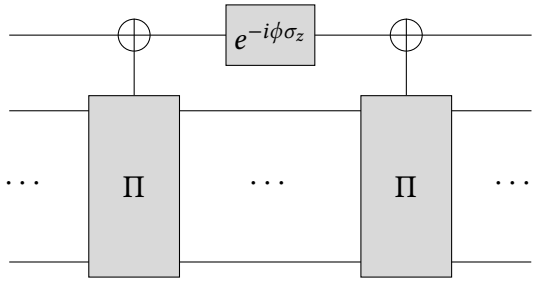

Figure 4.4: Implementing $\sum_b |b\rangle\langle b| \otimes e^{i(-1)^b \phi(2\Pi - I)}$.

The C_{Π}NOT gate is defined as $\Pi \otimes X + (I - \Pi) \otimes I$. In our weight preserving reduction, we replace the circuit V with our weight preserving instance C, and encode the ancillary qubit in the $\{|01\rangle, |10\rangle\}$ space as before, replacing all operations on the ancilla with their weight preserving counterpart. Now we explicitly construct Eq. (4.3). If we assume $V|\psi\rangle\left|1^{f(k)}0^p\right\rangle = \alpha|1\rangle|\varphi_1\rangle + \beta|0\rangle|\varphi_0\rangle$, let $\Phi \in \mathbb{R}^{2m}$ for Eq. (4.3), with $\Pi = I \otimes \left|1^{f(k)}0^p\right\rangle\!\left\langle 1^{f(k)}0^p\right|$ is the projector that checks the ancillary qubits are correctly initialized, and $\tilde{\Pi} = |1\rangle\langle 1| \otimes I$ is the accepting projector on the output qubit of V.

It is shown in [Gil19] that there exists some $\Phi \in \mathbb{R}^{2m}$, where m is set as in Theorem 4.14, such that

$$\left\|(\langle +| \otimes \Pi)(|0\rangle\langle 0| \otimes U_{\Phi} + |1\rangle\langle 1| \otimes U_{-\Phi})\left(|+\rangle \otimes |\psi\rangle \otimes \left|1^{f(k)}0^p\right\rangle\right)\right\|^2 \geq 1 - \epsilon,$$

$$\text{if } ||\tilde{\Pi}V|\psi\rangle||^2 \geq b;$$

$$\left\|(\langle +| \otimes \Pi)(|0\rangle\langle 0| \otimes U_{\Phi} + |1\rangle\langle 1| \otimes U_{-\Phi})\left(|+\rangle \otimes |\psi\rangle \otimes \left|1^{f(k)}0^p\right\rangle\right)\right\|^2 \leq \epsilon,$$

$$\text{if } ||\tilde{\Pi} V |\psi\rangle||^2 \leq a;$$

By careful examination of the new circuit constructed in [Gil19], we can show that the circuit could be implemented in a weight preserving manner, giving us the following corollary:

Corollary 4.15. *Given an instance circuit C of weight-k weight preserving quantum circuit with completeness and soundness c, s, $(c-s > 1/\text{poly}(n))$, we can construct a new weight preserving circuit C' with threshold $c' = 1-\epsilon, s' = \epsilon$, by making $\text{poly}(n)\log(1/\epsilon)$ calls to the circuit C.*

4.3.4 Spatially Sparse Weighted Local Hamiltonian

In this section we will consider an n qubit weight-preserving circuit W with R gates with a weight-k state and reduce it to an instance of the Local Hamiltonian problem with the property of being almost spatially sparse, which we define as follows

Definition 4.16 (Spatially Sparse Local Hamiltonian). A local Hamiltonian problem is *spatially sparse* if each qubit is only acted by $O(1)$ Hamiltonians.

Definition 4.17 (Almost Spatially Sparse Local Hamiltonian). A local Hamiltonian problem is *almost* spatially sparse with respect to a register of qubits if the Hamiltonian becomes spatially sparse if we remove all terms acting only on qubits in this register.

The spatially sparse local Hamiltonian is proven to be QMA complete in [OT08], their key lemma is stated as follows:

Lemma 4.18. *Given a verifier circuit V_x for a language $L \in$ QMA, there exists a spatially sparse local Hamiltonian $H = \sum_i H_i$ and $T = \text{poly}(n)$ that satisfies the following conditions:*

- *If V_x accepts some state $|\xi\rangle$ with probability $1-\epsilon$, there exists state $|\psi\rangle$ that $\langle\psi| H |\psi\rangle \leq \frac{\epsilon}{T+1}$.*

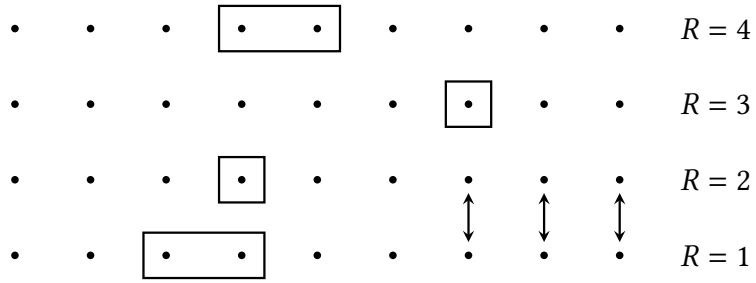

Figure 4.5: Reproduced Figure 1 of [OT08]. Each row of the qubits has the same number as the starting circuit. The number of rows is one more than the number of gates in the starting circuit. The R-th gate is performed on the R-th row and then all qubits are swappwd with those in the $(R+1)$-th row. This lazy simulation of the circuit will ensure that each qubit is acted on by a gate at most three times.

- *If V_x accepts any state $|\xi\rangle$ with probability no larger than ϵ, then all eigenvalues of H is larger than $\frac{c(1-\sqrt{\epsilon}-\epsilon)}{T^3}$, where c is some constant.*

This last lemma shows that the spatially sparse local Hamiltonian is QMA-hard, it can be easily seen that it is also in QMA. Our reduction from the weight preserving circuit to an almost spatially sparse local Hamiltonian will follow closely the construction in [OT08]. We give our reduction inspired in [OT08] in what follows. Consider the circuit $V_x = U_R \cdots U_2 U_1$ acting on n qubits and U_i are local gates from a universal gate set. We will map this circuit to a new circuit U_{sp} where the qubits will be arranged in a square grid and such that each qubit will be acted upon by a constant number of local gates. The grid will simulate the evolution of a qubit as time passes. The ith qubit in layer j in the grid of U_{sp} will simulate the state of the ith qubit in the jth step of running circuit V_x. In Fig. 4.5 we illustrate the simulation of circuit V_x with the grid. The first row $R = 1$ contains the initial qubits and the rectangle represents the two-qubit gate U_1. After this, all the qubits in the first row are swapped with those in the second row where the double arrows represent SWAP gates, after all the swaps, we act with gate U_2 and then repeat until the whole circuit has been simulated.

Next, we use Kitaev's clock construction to turn the circuit just described on the grid to a local Hamiltonian. We denote Q_{in} the set of qubits which correspond to the witness, Q_{out} the output qubit and $C = C_1, \cdots, C_T$ the clock registers. Let $U_{\text{sp}} = W_T \ldots W_2 W_1$

be the circuit acting over the grid qubits decomposed into local gates. The clock register will have $T+1$ qubits and valid clock basis states have the form $\left|0^{t-1}10^{T-t}\right\rangle_C$ for $t = 1, 2, \ldots, T$, i.e., we consider an indicator clock. Analogous to Kitaev's reduction, a valid history state for the circuit U_{sp} is

$$|\phi\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} \left|0^{t-1}10^{T-t}\right\rangle_C \otimes |\xi_t\rangle\,,$$

where $|\xi_t\rangle = W_t\,|\xi_{t-1}\rangle\,, |\xi_0\rangle = |\psi\rangle \otimes \left|1^{f(k)}0^m\right\rangle$. Note that each W_t is weight preserving and the initial state $|\xi_0\rangle$ for U_{sp} has weight $k + f(k) = k'$.

First, define t_q for the earliest time step when qubit q is actually used in the circuit U_{sp} and $\bar{i}_q$ is the inverse value in which ancillary qubit q should be initialized (so if qubit q should be initialized in state $|0\rangle$, then $\bar{i}_q = 1$). Consider then, the following Hamiltonian which sets the input state of the circuit.

$$H'_{\text{in}} = \sum_{q \notin Q_{\text{in}}} \left|\bar{i}_q\right\rangle\!\left\langle\bar{i}_q\right|_q \otimes |1\rangle\!\langle 1|_{C_{t_q-1}}\,. \tag{4.4}$$

Each term in this Hamiltonian increases the energy of those configurations where the qubits in the grid that are not part of the input are different from the value they should be initialized in. Note that the penalty is considered only when the clock qubit corresponding to time $t_q - 1$ is active. Next, we consider the Hamiltonian which introduces an energy penalty when the circuit does not accept the input.

$$H'_{\text{out}} = |0\rangle\!\langle 0|_{Q_{\text{out}}} \otimes |1\rangle\!\langle 1|_{C_T}\,. \tag{4.5}$$

Now we present the Hamiltonian terms which check that the proper circuit gates are applied on the input.

$$H'_{\text{prop}} = \sum_{t=1}^{T} H'_{\text{prop},t}, \tag{4.6}$$

where

$$H'_{\text{prop},t} = (|10\rangle\!\langle 10| + |01\rangle\!\langle 01|)_{C_{t,t+1}} - W_t \otimes |01\rangle\!\langle 10|_{C_{t,t+1}} - W_t^\dagger \otimes |10\rangle\!\langle 01|_{C_{t,t+1}}\,.$$

Each term $H'_{\text{prop},t}$ ensures that gate W_t is applied at time t. Before presenting the Hamiltonian terms checking the clock, we add an additional ingredient to our construction.

Consider the following isometry $\mathcal{U}$ on the state registers of our Hamiltonian: for each qubit q, we will duplicate it in the computational basis, i.e., $\mathcal{U}|0\rangle_q = |00\rangle_{I_q}$, $\mathcal{U}|1\rangle_q = |11\rangle_{I_q}$, where I_q are two qubits indicating the original qubit q, and $I_q \cap I_{q'} = \emptyset$ for $q \neq q'$. We remark that this isometry is applied over the state register or in other words, the qubits in the grid originally defined. The isometry does not affect the clock qubits. Thus we obtain new Hamiltonian terms constructed by conjugating $\mathcal{U}$ over the previous Hamiltonian terms; $H_{\text{in}} = \mathcal{U}H'_{\text{in}}\mathcal{U}^\dagger$, $H_{\text{out}} = \mathcal{U}H'_{\text{out}}\mathcal{U}^\dagger$ and $H_{\text{prop}} = \mathcal{U}H'_{\text{prop}}\mathcal{U}^\dagger$. Due to the isometry just described, the qubits in each I_q need to be either in the state $|00\rangle$ or $|11\rangle$ as they encode a single qubit. We need to include Hamiltonian terms that check that these are the only possibe states.

$$H_{\text{state}} = \sum_q |01\rangle\langle 01|_{I_q} + |10\rangle\langle 10|_{I_q}. \tag{4.7}$$

The Hamiltonian terms that check the correctness of the clock is given by

$$H_{\text{clock}} = \sum_{t<t'} |11\rangle\langle 11|_{C_{t,t'}}. \tag{4.8}$$

Our final local Hamiltonian will have the form $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{clock}} + H_{\text{state}}$ where

$$\begin{aligned}
H_{\text{in}} &= \sum_{q \notin Q_{\text{in}}} \left|\bar{i}_q \bar{i}_q\right\rangle\!\left\langle \bar{i}_q \bar{i}_q\right|_{I_q} \otimes |1\rangle\langle 1|_{C_{t_q-1}}, \\
H_{\text{out}} &= |00\rangle\langle 00|_{I_{Q_{\text{out}}}} \otimes |1\rangle\langle 1|_{C_T}, \\
H_{\text{clock}} &= \sum_{t<t'} |11\rangle\langle 11|_{C_{t,t'}} \\
H_{\text{state}} &= \sum_q |01\rangle\langle 01|_{I_q} + |10\rangle\langle 10|_{I_q}, \\
H_{\text{prop}} &= \sum_{t=1}^{T} H_{\text{prop},t},
\end{aligned}$$

and

$$H_{\text{prop},t} = (|10\rangle\langle 10| + |01\rangle\langle 01|)_{C_{t,t+1}} - W'_t \otimes |01\rangle\langle 10|_{C_{t,t+1}} - (W'_t)^\dagger \otimes |10\rangle\langle 01|_{C_{t,t+1}},$$

where $W'_t = \mathcal{U}|_{Q_{W_t}} W_t \mathcal{U}|^{\dagger}_{Q_{W_t}} \otimes I_{2n-2|Q_{W_t}|}$, Q_{W_t} for the qubits that W_t acts on. In our new construction, our history state could be defined as $|\phi'\rangle = (I \otimes \mathcal{U})|\phi\rangle$. We can observe that $\mathcal{U}$ doubles the weight on the state registers, the weight of our new witness state is $2k' + 1$. The fact that the weight on the new Hamiltonian has odd weight is important, as this makes the clock guarantee that it will have only a single qubit in state $|1\rangle$. If we didn't have this requirement, then it would be possible for the clock to be in the state $\left|0^T\right\rangle_C$ which would not be penalized with our Hamiltonian.

The difference between our construction and Oliveira-Terhal [OT08] is in the clock design and checking terms. We use the indicator clock and it is easy to see H_{clock} and H_{state} are 2-local Hamiltonians. H_{state} guarantees the two mapped qubits in I_q always have the same value, thus all valid witnesses should have even weight on the state registers. Since we require the weight of witness state to be odd, the clock registers must have non-zero weight, and H_{clock} guarantees the only valid clock states are the indicator states $\left|0^{t-1}10^{T-t}\right\rangle_C$.

For the completeness part, observe that if original V accepts $|\psi\rangle$ with probability $1-\epsilon$, the history state $|\phi'\rangle = (I \otimes \mathcal{U})|\phi\rangle$ would be projected to 0 for all Hamiltonian terms but H_{out}. Since U_{sp} simulates V faithfully, we obtain that $\langle\phi'| H_{\text{out}} |\phi'\rangle \leq \frac{\epsilon}{T+1}$. To see this, assume that $|\psi\rangle$ is an input state to the weight preserving circuit V such that $\|\Pi_1 V|\psi\rangle\|^2 \geq 1-\epsilon$, where Π_1 is the projector acting on the output qubit projecting onto the state $|1\rangle$. Let U_{sp} be the simulation of V using the method of [OT08] which we described above, since the simulation is faithful we have that $\left\|\Pi_1^{Q_{\text{out}}} U_{\text{sp}}|\psi\rangle \otimes \left|1^{f(k)}0^m\right\rangle\right\|^2 \geq 1-\epsilon$. Now, note that

$$
\begin{aligned}
\langle\phi'|H_{\text{out}}|\phi'\rangle &= \langle\phi|(I\otimes\mathcal{U}^{\dagger})(|00\rangle\langle 00|_{I_{Q_{\text{out}}}} \otimes |1\rangle\langle 1|_{C_T})(I\otimes\mathcal{U})|\phi\rangle \\
&= \langle\phi|(|0\rangle\langle 0|_{Q_{\text{out}}} \otimes |1\rangle\langle 1|_{C_T}) \\
&\quad \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} \left|0^{t-1}10^{T-t}\right\rangle_C \otimes W_t \cdots W_1 \left(|\psi\rangle \otimes \left|1^{f(k)}0^m\right\rangle\right) \\
&= \left\langle\phi\middle|0_{Q_{\text{out}}}1_{C_T}\right\rangle \frac{1}{\sqrt{T+1}} \left(|0\cdots 01\rangle_C \langle 0|_{Q_{\text{out}}} W_T \cdots W_1 \left(|\psi\rangle \otimes \left|1^{f(k)}0^m\right\rangle\right)\right)
\end{aligned}
$$

$$= \frac{\left\| \Pi_0^{Q_{\text{out}}} U_{\text{sp}} |\psi\rangle \otimes \left|1^{f(k)} 0^m\right\rangle \right\|^2}{T+1}$$
$$\leq \frac{\epsilon}{T+1}. \tag{4.9}$$

For soundness, consider the vector space of legal history states $\mathcal{S} = \{|\phi\rangle : H_{\text{clock}} |\phi\rangle = H_{\text{state}} |\phi\rangle = 0\}$. Since H preserves this space and the corresponding perpendicular space $\mathcal{S}^\perp$, we can discuss both of this spaces separately. Any eigenvector in $\mathcal{S}^\perp$ has eigenvalue at least 1 as it would violate the clock or the state constraint. The proof is similar to the one in [OT08] which in turn follows from the proof of [KSV02]. For the soundness in the subspace $\mathcal{S}$, consider the Hamiltonian restricted to this subspace $H|_{\mathcal{S}}$ (in this subspace, the clock and state constraint are satisfied). Define $H' = H'_{\text{in}} + H'_{\text{out}} + H'_{\text{prop}}$ and note that $\mathcal{U} H' \mathcal{U}^\dagger|_{\mathcal{S}} = H|_{\mathcal{S}}$ since in the subspace $\mathcal{S}$, the clock and the state Hamiltonian are 0. In [OT08], they performed analysis of eigenvalue on $H'|_{\mathcal{U}^\dagger \mathcal{S} \mathcal{U}}$, which is isometric to $\mathcal{U} H' \mathcal{U}^\dagger|_{\mathcal{S}}$, thus we obtain the same eigenvalue lower bound $\frac{c(1-\sqrt{\epsilon}-\epsilon)}{T^3}$. The resulting Hamiltonian in our reduction is not spatially sparse as in [OT08] because the clock checking Hamiltonian S_{clock} is not sparse. Excluding the clock checking terms, however, all other terms are spatially sparse. Therefore, this Hamiltonian is *almost* spatially sparse with respect to the clock register. Note that if we use Lemma 4.9 and a finite gate set, the types of resulting Hamiltonian terms will also be finite. We conclude with the following corollary:

Corollary 4.19. *Given a weight-k weight-preserving quantum circuit satisfiability instance C with parameter $(\epsilon, 1-\epsilon)$, we can construct a weight-$2k'+1$ almost spatially sparse local hamiltonian instance with energy thresholds $a = \frac{\epsilon}{T+1}, b = \frac{c(1-\sqrt{\epsilon}-\epsilon)}{T^3}$. Furthermore, if we assume C acts on n qubits, we have $T \leq 3n(|C|+1)$, the resulting Hamiltonian would act on $2n(|C|+1)+T+1$ qubits, and $k' = f(k)$ for some computable function f.*

4.3.5 QW[1] Verification for Almost Spatially Sparse Hamiltonian Problems

We are now ready to show that the almost spatially sparse Hamiltonian problem we end up with in the last subsection is in $\mathsf{QW}[1]$. To prove this, we design a constant depth circuit which verifies the almost sparse local Hamiltonian problem. There are two main techniques in the proof. The first technique will consist in taking advantage of the sparseness of the Hamiltonian terms (which do not include the clock) by coloring them such that no two terms acting on the same qubit are of the same color. This will allow us to measure in parallel the energy of Hamiltonian terms of the same color and thus perform this measurements in constant depth. Second, to check that the clock is in the correct state, we use the fact that the clock has the indicator format (a single 1 in the position corresponding to the time). We can measure the clock state an then perform a computation in $\mathsf{W}[1]$ (a constant-depth circuit with classical fan-outs) to check the result. Thanks to the simplification of the clock checking term using the weight constraint, it suffices to check that there are no two 1's in the measurement outcome of the clock register. This can be done in $\mathsf{W}[1]$, and therefore simulated by a constant depth quantum circuit with one big AND gate.

We need the following lemma to relate parallel measurements and Hamiltonian sum later on.

Lemma 4.20. *Let $M_1, M_2, \ldots, M_m$ be m commuting operators satisfying $0 \leq M_j \leq I$, then we have*

$$I - \sum_{j=1}^{m} M_j \leq \prod_{j=1}^{m} (I - M_j) \leq I - \frac{1}{m} \sum_{j=1}^{m} M_j.$$

Proof. By the commutativity of the m operators and the spectral decomposition theorem, this problem reduces to the scalar case. For real numbers $x_j \in [0, 1]$ where $j = 1, 2, \ldots, m$,

$$1 - \sum_{j=1}^{m} x_j \leq \prod_{j=1}^{m} (1 - x_j)$$

follows from a simple induction on m and

$$1 - \frac{1}{m}\sum_{j=1}^{m} x_j \geq \prod_{j=1}^{m}(1 - x_j)$$

follows from the geometric and arithmetic mean inequality

$$\frac{\sum_j (1 - x_j)}{m} \geq \left(\prod_j (1 - x_j)\right)^{1/m} \geq \prod_j (1 - x_j).$$

■

This last Lemma will help us bounding the energies when measuring in parallel. Now we show the following Lemma which shows that the almost spatially sparse local Hamiltonian can be verified by $\mathsf{QW}[1]$ circuits, provided that the energy threshold fulfill the constraint $b/n^2 - a \geq 1/\mathrm{poly}(n)$. In the proof of the Lemma we will see why this condition is necessary.

Lemma 4.21. *Let $H = \sum_j H_j$ be a local Hamiltonian problem that acts on n qubits. The energy thresholds a and b for the problem satisfies $b/n^2 - a \geq 1/\mathrm{poly}(n)$. Suppose that Hamiltonian H is almost spatially sparse with respect to a clock register of n_{clock} qubits and that each term H_j in the Hamiltonian is a projector. That is, except clock checking terms $|11\rangle\langle 11|_{C_{t,t'}}$ acting on qubits C_t and $C_{t'}$ in the clock register, all other Hamiltonian terms in H are spatially sparse. Then, there is a $\mathsf{QW}[1]$ verification circuit V and $c, s \in \mathbb{R}$ satisfying $c - s \geq 1/\mathrm{poly}(n)$ such that if the ground state energy of H is at most a, V accepts with probability c while if the ground state energy of H is at least b, V accepts with probability s. Furthermore, V can be chosen so that the big gate is a classical AND gate and it is the last gate in V.*

Proof. As the Hamiltonian is almost spatially sparse, it is possible to color the terms using $n_{\mathrm{color}} + 1$ colors where n_{color} is a constant. Note that this constant will depend on the locality of the Hamiltonian. We use $G^{(h)}$ to denote the set of terms of color h. For the first n_{color} sets $G^{(h)}$ where $h = 0, 1, \ldots, n_{\mathrm{color}} - 1$, the terms $H_j^{(h)}$ in the color group

$$G^{(h)} = \left\{H_j^{(h)} \mid j = 1, 2, \ldots, m_h\right\}$$

acts on different qubits for all j. Here, m_h is the number of terms in group $G^{(h)}$. For the last group $G^{(n_{\text{color}})}$, the terms are $H_j^{(n_{\text{color}})} = |11\rangle\langle 11|_{C_{t,t'}}$ acting on all pairs of qubits $C_t, C_{t'}$ in the clock register, i.e., the last color is assigned to the terms of the clock Hamiltonian. The number of terms in this group is $m_{n_{\text{color}}}$. Define

$$m_{\max} = \max\{m_i \mid i = 0, 1, \ldots, n_{\text{color}}\}.$$

For each $h = 0, 1, \ldots, n_{\text{color}} - 1$, the size m_h is at most n as the terms in $G^{(h)}$ all act on different qubits. For $h = n_{\text{color}}$, m_h is at most n^2 as $n_{\text{clock}} \leq n$ and the terms run over a pair of clock qubits. This implies that $m_{\max} \leq n^2$.

We now present the $\mathsf{QW}[1]$ verification circuit V as follows.

(1) First the circuit samples a random integer $h \in \{0, 1, \ldots, n_{\text{color}}\}$.

(2) Conditioned on h the circuit checks all the terms in the group $G^{(h)}$. In particular,

a) If $h < n_{\text{color}}$, the circuit performs measurements

$$\{M_{j,1}^{(h)} = I - H_j^{(h)}, M_{j,0}^{(h)} = H_j^{(h)}\},$$

for all $j = 1, 2, \ldots, m_h$. The circuit outputs the AND of all measurement outcomes.

b) If $h = n_{\text{color}}$, the circuit performs computational basis measurements on all the clock qubits. The circuit outputs the AND of all pairwise NAND of the measurement outcomes.

First, we note that the sampling of the integer h can be done using a constant size quantum circuit and computational basis measurement. We can fanout the measurement outcomes to control the later parts in the circuit. Second, as the Hamiltonian terms in each group $G^{(h)}$ act on different qubits for all $h = 0, 1, \ldots, n_{\text{color}} - 1$, the measurements $\{M_{j,0}, M_{j,1}\}$ can be implemented in parallel. These measurements output x_h, an m_h-bit vector of classical information. For $h = n_{\text{color}}$, the circuit first measures all the clock qubits and computes the pairwise NAND of the outcome. We denote this vector of

classical bits as $x_{n_{\text{color}}}$, its length is $m_{n_{\text{color}}}$. So far, all gates involved are constant size quantum circuits and the classical fanout gates. Finally, the output of the circuit V is the AND of x_h for the sampled integer h. It is easy to reuse the AND gate in all $n_{\text{color}} + 1$ cases as we can use fanout of input 1 to pad short x_h's so that they all have length $m_{\max}$. Then we use controlled SWAP gates to move the bits in x_h to the same register that can hold $m_{\max}$ qubits and output their AND.

Now we show that the soundness and completeness probabilities for the circuit are related to the gap condition on the almost spatially sparse local Hamiltonian. First, note that for the case $h = n_{\text{color}}$, the measurement associated to this choice of h has the probability of acceptance

$$\langle\psi| \left(\sum_{x:|x|\leq 1} |x\rangle \langle x| \right) |\psi\rangle = \langle\psi| \prod_{k,l} (I - |11\rangle \langle 11|)_{k,l} |\psi\rangle = \langle\psi| \prod_{j} \left(I - H_j^{(n_{\text{color}})}\right) |\psi\rangle .$$

where $|x|$ denotes the Hamming weight of bitstring x. This is the probability of acceptance because each projector in the last equality checks whether there are no two 1's in the clock.

The acceptance probability of the circuit is then given by

$$\Pr(V \text{ accepts}) = \frac{1}{n_{\text{color}} + 1} \sum_{h=0}^{n_{\text{color}}} \langle\psi| \bigotimes_{j=1}^{m_h} (I - H_j^{(h)}) |\psi\rangle . \tag{4.10}$$

Its important to remark here that each of this projectors can be implemented in constant depth since each local Hamiltonian only involves a finite number of qubits. In the yes case, the Hamiltonian has ground state energy at most a, which means that there is a witness state $|\psi\rangle$

$$\langle\psi| H |\psi\rangle = \langle\psi| \sum_{j=1}^{m} H_j |\psi\rangle \leq a.$$

Hence, continuing on Eq. (4.10), we have

$$\begin{aligned} \Pr(V \text{ accepts}) &\geq \frac{1}{n_{\text{color}} + 1} \sum_{h=0}^{n_{\text{color}}} \langle\psi| \left(I - \sum_{j=1}^{m_h} H_j^{(h)}\right) |\psi\rangle \\ &= 1 - \frac{\langle\psi| H |\psi\rangle}{n_{\text{color}} + 1} \geq 1 - \frac{a}{n_{\text{color}} + 1}, \end{aligned}$$

where the inequality follows from Lemma 4.20.

In the no case, we have for all state $|\psi\rangle$ of certain weight

$$\langle\psi| H |\psi\rangle = \langle\psi| \sum_{j=1}^{m} H_j |\psi\rangle \geq b.$$

From Eq. (4.10) we see that

$$\begin{aligned}\Pr(V \text{ accepts}) &\leq \frac{1}{n_{\text{color}}+1} \sum_{h=0}^{n_{\text{color}}} \langle\psi| \left(I - \frac{1}{m_h} \sum_{j=1}^{m_h} H_j^{(h)}\right) |\psi\rangle \\ &\leq \frac{1}{n_{\text{color}}+1} \sum_{h=0}^{n_{\text{color}}} \langle\psi| \left(I - \frac{1}{m_{\max}} \sum_{j=1}^{m_h} H_j^{(h)}\right) |\psi\rangle \\ &= 1 - \frac{\langle\psi| H |\psi\rangle}{m_{\max}(n_{\text{color}}+1)} \leq 1 - \frac{b}{n^2(n_{\text{color}}+1)},\end{aligned}$$

where the first inequality follows from Lemma 4.20. We have then completeness c and soundness s defined as

$$c = 1 - \frac{a}{n_{\text{color}}+1}, \quad s = 1 - \frac{b}{n^2(n_{\text{color}}+1)}.$$

$c - s = (b/n^2 - a)/(n_{\text{color}}+1) \geq 1/\text{poly}(n)$ follows from the strong gap condition on a, b for the Hamiltonian problem. ∎

From this proof we can also conclude that WEIGHT-k ℓ-LOCAL HAMILTONIAN and WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY can be reduced to each other.

Corollary 4.22. *Given a, b with $b - a > 1/\text{poly}(n)$,* WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b) *reduces to* WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) *under* FPT *reduction for some c, s such that $c - s > 1/\text{poly}(n)$. The same is true when reducing* WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) *to* WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b).

Proof. That WEIGHT-k ℓ-LOCAL HAMILTONIAN(a, b) reduces to WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) has been already shown. It has been shown also that WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT

SATISFIABILITY(c, s) reduces to almost spatially sparse weighted Local Hamiltonians.

■

Finally combining the above sections together, we could provide a proof for Theorem 4.4.

Proof. By Lemma 4.13, given a WEIGHT-k LOCAL HAMILTONIAN (a, b) instance $H = \sum_{j=1}^{m} H_j$ on n qubits with $b - a > 1/\text{poly}(n)$, we can obtain a WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY instance W with size $O(km\,\text{poly}(n)) = O(k\,\text{poly}(n))$, acting on $O(n+M+k) = \text{poly}(n)+k$ qubits, completeness $1 - \frac{m+a}{M}$ and soundness $1 - \frac{m+b}{M}$.

Now we can apply Corollary 4.15 to amplify the gap to $(2^{-n}, 1-2^{-n})$, and the new circuit has size $|C| = O\left(\frac{m}{b-a}|W|\log(2^n)\right) = O(k\,\text{poly}(n))$ acting on $n' = \text{poly}(n) + k$ qubits. Using the parameters in Corollary 4.19, we can construct a weight-$2k'+1$ almost spatially sparse local Hamiltonian instance H_{sp} with following parameters: $k' = k+O(1)$, $T \leq 3n'(|C|+1) = O(k^2\,\text{poly}(n))$, $a = \frac{1}{(T+1)2^n}$, $b = \frac{c(1-2^{-n/2}-2^{-n})}{T^3}$. The Hamiltonian H_{sp} acts on $n_f = O(k^2\,\text{poly}(n))$ qubits.

Finally we apply Lemma 4.21 to obtain our final QW[1] circuit. We can check that the energy thresholds a, b we obtained in the step beyond satisfies $b/n_f^2 - a \geq 1/\text{poly}(n)$. Thus our QW[1] circuit constructed in Lemma 4.21 has probability gap $c-s \geq 1/\text{poly}(n)$ since $k \leq n$. ■

4.4 QW-hierarchy and ETH

As mentioned in the introduction, one of the most important uses of parameterized complexity theory is in the fine-grained complexity analysis. In particular, there are important connections between W[1] and the exponential time hypothesis (ETH), some of which are presented in the book Fundamentals of Parameterized Complexity by Downey and Fellows (2013) [DF13]. We use the version of ETH that can be found in

Section 29.4 of [DF13]. In what follows we say that a circuit C has total description size D if the number of inputs and total number of gates are bounded by D.

Definition 4.23 (Exponential Time Hypothesis (ETH)). We define the Exponential Time Hypothesis as follows. There is no algorithm with running time $2^{o(n)}$ that decides for a weft 1 Boolean circuit C of total description size n, whether there is an input vector x such that $C(x) = 1$.

Note that this is a weaker definition than the typical one for ETH. The reason for the slight weakening of the hypothesis is done in order to make connections between fine-grained complexity and parameterized complexity (see Chapter 16, in [FG06]). In this section we shall consider a quantum version of ETH together with a quantum-classical version.

Definition 4.24 (Quantum Exponential Time Hypothesis (QETH)). We define the QETH as follows. For some c, s with $c - s > 1/\text{poly}(n)$, there is no *quantum* algorithm running in time $2^{o(n)}$ that decides for a weft-1 quantum circuit Q of total description size n whether (i) there is an input witness state $|\psi\rangle$ such that $\Pr(Q(|\psi\rangle) \text{ accepts}) \geq c$ or (ii) for all input witness states $|\psi\rangle$, $\Pr(Q(|\psi\rangle) \text{ accepts}) \leq s$, given the promise that one of the two holds.

Definition 4.25 (Quantum-Classical Exponential Time Hypothesis (QCETH)). We define the QCETH as follows. There is no *quantum* algorithm running in time $2^{o(n)}$ that decides for a weft-1 Boolean circuit C of total description size n, whether there is an input vector x such that $C(x) = 1$.

We have defined QETH as a hypothesis about *some* pair c, s with inverse polynomial gap rather than all such pairs c, s. The reason for this choice is that we want to show that if certain problems are tractable given any polynomial gap, then QETH is false. This will be evident later in this section. Nonetheless, we remark that by changing the definition of QETH, Proposition 4.26 would not be affected and Theorem 4.37 would

require some minimal modification. A natural question is the relationship between these two hypothesis just defined. We prove first QETH is a weaker statement than QCETH.

Proposition 4.26. QCETH *implies* QETH.

Proof. Assume that QETH is false, then there is a quantum algorithm $\mathcal{A}$ deciding the problem in Definition 4.24. We shall construct a quantum circuit Q and show that the satisfiability problem on C reduces to the satisfiability problem on Q. Let C be a weft-1 classical circuit of total description size n, we can assume that C has n gates of bounded fan-in f with gate basis {AND, OR, NOT}. First, we modify C into a reversible circuit by adding an ancilla bit initialized at 0 for each AND and OR gate, including the weft-1 gates. Note that this increases the number of input bits by n since there are at most n gates. For the fan-out gates in the classical circuit, these can be replaced by reversible CNOTs. Note that there are at most $f \cdot n$ possible inputs to the bounded fan-in gates, which implies we require at most $O(n)$ CNOT gates. After this procedure, we end up with a reversible circuit which can be transformed easily into a quantum circuit Q with $O(n)$ inputs and $O(n)$ gates and generalized Toffoli for weft-1 gates. We also include in Q a procedure to check that the ancilla qubits are all set to $|0\rangle$, which requires $O(n)$ measurements and gates.

Now we show the decision problem in Definition 4.25 with circuit C reduces to the promise problem with circuit Q in Definition 4.24 with completeness $b = 1$ and soundness $a = 0$. If there is $x \in \{0,1\}^n$ such that $C(x) = 1$, then consider the state $|x0^{cn}\rangle$ where $c > 0$ and $(c+1)n$ is the number of inputs to Q. We have then $Q|x0^{cn}\rangle = \sum_{y\in\{0,1\}^{(c+1)n-1}} \beta_y |1y\rangle$, where $\beta_y \in \mathbb{C}$ and $\sum_y |\beta_y|^2 = 1$. Letting $\Pi_1^{(0)} = |1\rangle\langle 1|$ be the projector onto the state $|1\rangle$ for the first qubit, we have that

$$\begin{aligned}\Pr\left(Q \text{ accepts } |x0^{cn}\rangle\right) &= \left\|\Pi_1^{(0)} Q|x0^{cn}\rangle\right\|^2 \\ &= \left\|\sum_{y\in\{0,1\}^{(c+1)n-1}} \beta_y |1y\rangle\right\|^2 \\ &= 1 \end{aligned} \tag{4.11}$$

Suppose now that for all $x \in \{0,1\}^n$, $C(x) = 0$. We have that $Q|x0^{cn}\rangle = \sum_{y\in\{0,1\}^{(c+1)n-1}} \beta_{y,x}|0y\rangle$ and thus $\Pi_1^{(0)}Q|x0^{cn}\rangle = 0$. Any state passing the initial verification of the ancillae qubits has the form $|\psi\rangle = \sum_{x\in\{0,1\}^n} \gamma_x |x0^{cn}\rangle$, with $\sum_x |\gamma_x|^2 = 1$. Then we have that

$$\begin{aligned}\Pr(Q \text{ accepts } |\psi\rangle) &= \left\|\Pi_1^{(0)}Q|\psi\rangle\right\|^2 \\ &= \left\|\Pi_1^{(0)}\sum_{x,y}\beta_{y,x}\gamma_x|0y\rangle\right\|^2 \\ &= 0. \end{aligned} \tag{4.12}$$

This shows the reduction and thus algorithm $\mathcal{A}$ can solve in time $2^{o(n)}$ the decision problem for circuit C and QCETH is false. ■

4.4.1 Miniaturized problems and ETH

In this subsection we shall introduce miniaturized problems which are a key ingredient in connecting results from parameterized complexity and ETH. First, we define the miniature version of the classical circuit satisfiability problem and then we will show how it connects to ETH and QCETH.

Definition 4.27 (MINI-CIRCSAT$_t$).

Instance: Positive integers k and n in unary, and a weft t Boolean circuit C of total description size at most $k \log n$.

Parameter: k in the problem instance.

Problem: Decide whether there is an input binary vector x such that $C(x) = 1$.

For simplicity, we will refer to MINI-CIRCSAT$_1$ as MINI-CIRCSAT. The following theorem illustrates the connection between the tractability of miniature problems and ETH.

Theorem 4.28 (Theorem 29.4.1 in [DF13]). MINI-CIRCSAT *is in* FPT *if and only if* ETH *is false.*

The MINI-CIRCSAT can be then reduced to WEIGHT-k INDEPENDENT SET which implies the following theorem.

Theorem 4.29 (Section 29.4 of [DF13]). *If* $\mathsf{W}[1] = \mathsf{FPT}$ *then* ETH *is false.*

Theorem 4.29 establishes a sufficient condition for ETH to be false. In classical parameterized complexity the complexity class $\mathsf{M}[1]$ is defined as the closure under FPT reductions of Mini-CircSAT, the claim that this class is tractable for FPT algorithms is equivalent to ETH being false.

Definition 4.30. Define $\mathsf{M}[t]$ as the set of problems FPT reducible to MINI-CIRCSAT$_t$

Theorem 4.31 (Restatement of Theorem 4.28). $\mathsf{M}[1] = \mathsf{FPT}$ *if and only if* ETH *is false.*

As an aside, it is straightforward to see that the weighted local Hamiltonian problem is $\mathsf{W}[1]$-hard, which makes unlikely any FPT algorithms for this problem as implied by the above theorem. To prove this we can simply reduce the weighted independent set problem to the weighted local Hamiltonian problem.

Proposition 4.32. *The* WEIGHT-k INDEPENDENT SET *problem reduces to the* WEIGHT-k LOCAL HAMILTONIAN PROBLEM(a, b) *under* FPT *reductions, for any* a, b *with* $b > a \geq 0$.

Proof. Let $G = (V, E)$ be a graph with vertex set $V = \{1, 2, \cdots, n\}$. For each $i \in V$ define a binary variable x_i and the formula $\varphi(x_1, \cdots, x_n) = \bigwedge_{(i,j)\in E}(\neg x_i \vee \neg x_j)$. G has an independent set of size k if and only if φ is satisfiable by a bitstring $x = x_1, \cdots x_n$ of Hamming weight k. We can map φ to a Hamiltonian $H = \sum_i H_i$ acting over n qubits, for this, consider the one qubit projector over qubit i, $\Pi_1^{(i)} = |1\rangle\langle 1|$. We map each term $(\neg x_i \vee \neg x_j)$ to $H_i = \Pi_1^{(i)}\Pi_1^{(j)}$. This Hamiltonian H is an instance of the WEIGHT-k

LOCAL HAMILTONIAN and has a ground state of energy 0 with weight-k if and only if graph G has an independent set of size k. Note this reduction works as long as the condition over the a, b in the proposition is as given. ■

It's known that WEIGHT-k INDEPENDENT SET is W[1]-complete [DF13], thus this implies that the weighted local Hamiltonian problem is W[1]-hard. An immediate consequence is that its unlikely that there are FPT algorithms for the weighted Local Hamiltonian as this would imply that ETH is false by Theorem 4.29. As we show in Theorem 4.40, if this problem can be solved by FPQT algorithms then this implies that QCETH is false.

We can trivially generalize Theorem 4.28 to the quantum case, in particular we will frame the results in terms of the weighted local Hamiltonian problem. We can give a trivial generalization of Theorem 4.28 as follows

Theorem 4.33. W[1] ⊆ FPQT *iff* QCETH *is false.*

Proof. The proof follows from a direct generalization from the proof of Theorem 4.28 in [DF13]. If QCETH is false, then we can solve MINI-CIRCSAT with a quantum algorithm in time $2^{o(k \log n)}$ which is an FPT function, implying that W[1] ⊆ FPQT.

Let C be a Boolean circuit of weft 1 and size N and assume there is an FPQT algorithm that solves MINI-CIRCSAT in time $f(k)n^c$ where we assume f to be a growing function in k. We now show that there is an algorithm deciding if C is satisfiable in time $2^{o(n)}$. Take $k = f^{-1}(N)$ and $n = 2^{(N/k)}$, thus, $N = k \log n$. In general, $f^{-1}(N)$ will be a growing function of N and thus $N/k = o(N)$. We can now consider the circuit C as an instance of MINI-CIRCSAT with k and n chosen as before, giving a runtime for the algorithm of $f(f^{-1}(N))(2^{N/k})^c = 2^{\frac{cN}{k} + \log N} = 2^{o(N)}$, thus QCETH is false. ■

As shown in Proposition 4.32, the weighted independent set problem can be reduced to the weighted local Hamiltonian problem. Moreover, as remarked before, MINI-CIRCSAT reduces to the weighted independent set.

This shows the following

Theorem 4.34. *If* WEIGHT-k ℓ-LOCAL HAMILTONIAN *is in* FPQT *then* QCETH *is false.*

Proof. The WEIGHT-k INDEPENDENT SET reduces to the WEIGHT-k ℓ-LOCAL HAMILTONIAN, by hypothesis we can solve instances of the Local Hamiltonian problem in FPQT and thus WEIGHT-k INDEPENDENT SET as well. By Theorem 4.33 the result follows. ∎

4.4.2 Miniaturized problems and QETH

Now we turn to a result pertaining to QETH as defined in Definition 4.24. Let us begin by defining a miniature version of the quantum circuit satisfiability problem.

We define the miniature version of the quantum circuit satisfiability MINI-QCSAT$_t(a, b)$ and the class QM$[t]$ as follows

Definition 4.35 (MINI-QCSAT$_t(c, s)$).

Instance: Integers k and n in unary, and weft-t quantum circuit C of description size $k \log n$.

Parameter: A natural number k.

Yes: There exists an input quantum state $|\psi\rangle$, such that $\Pr[C(|\psi\rangle) \text{ accepts}] \geq c$.

No: For every input quantum state $|\psi\rangle$, $\Pr[C(|\psi\rangle) \text{ accepts}] \leq s$.

Definition 4.36. Define QM$_{c,s}[t]$ as the set of problems FPQT-reducible to MINI-QCSAT$_t(c, s)$ and define QM$[t]$ as

$$\mathsf{QM}[t] := \bigcup_{\substack{c,s \\ c-s>1/poly(n)}} \mathsf{QM}_{c,s}[t].$$

We denote as Mini-QCSAT(c, s) the problem MINI-QCSAT$_1(c, s)$. Just as in the classical case, we give a theorem connecting the complexity of MINI-QCSAT and QETH from Definition 4.24.

Theorem 4.37. $\mathsf{QM}[1] \subseteq \mathsf{FPQT}$ *iff* QETH *is false.*

Proof. The argument from Theorem 4.33 can be repeated. First assume QETH is false, then for all c, s with polynomial gap there is an algorithm that solves the quantum circuit satisfiability problem with completeness c and soundness s with $c - s > 1/\mathrm{poly}(n)$. Then, given an instance C of Mini-QCSAT(c, s) we can use this algorithm to solve it in time $2^{o(k \log n)}$ which is an FPT function.

Now assume that for all c, s with polynomial gap, Mini-QCSAT(c, s) is solvable in time $f(k)n^{c_0}$ time for some constant $c_0 > 0$. Let C be a weft-1 circuit of size N. Set $k = f^{-1}(N)$ and $n = 2^{(N/k)}$, which implies $N = k \log n$. In general it will be true that $N/k = o(N)$. Using the FPQT algorithm on C, we have a running time $2^{o(N)}$ which solves the decision problem with completeness c and soundness s. Since this is true for all c, s such that $c - s > 1/\mathrm{poly}(n)$ then QETH is false. ∎

Now we show that the Mini-QCSAT reduces to the weight-preserving quantum circuit satisfiability problem from Definition 4.1.

Lemma 4.38. WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) *is* $\mathsf{QM}_{c,s}[1]$*-hard.*

Proof. Let C describe a Mini-QCSAT(c, s) circuit with at most $k \log n$ inputs and $k \log n$ gates. We can decompose these gates into one qubit gates and CNOTs, increasing the number of gates to poly$(k \log n)$. Note that a $k \log n$ qubit state $|\chi\rangle$ can be mapped to a weight-k n-qubit state $|\psi\rangle$ by considering the natural encoding of an n qubit state of weight-k with $k \log n$ qubits. If C has less than $k \log n$ input qubits then we can always add ancillas in the $|0\rangle$ state, and measure at the end of the circuit to check that they are all in the $|0\rangle$ state, we can thus assume that C has $k \log n$ input qubits.

We take the $k \log n$ input qubits and divide them into k groups of $\log n$ qubits and consider the encoding of the $\log n$ qubit state into an n qubit state of weight-1. For bitstring $x \in \{0, 1\}^{\log n}$ we denote as $E(x)$ the encoding into a bitstring of length n and Hamming weight 1 which preserves lexicographic order. For example, if $n = 4$

we consider the encoding $|E(00)\rangle = |0001\rangle$, $|E(01)\rangle = |0010\rangle$, $|E(10)\rangle = |0100\rangle$ and $|E(11)\rangle = |1000\rangle$. This mapping will result in a circuit with kn input qubits. We now explain how to map the gates in circuit C to the encoded version C' in such a way that the weight is preserved in circuit C'. A one-qubit gate V in C is mapped to $2^{\log n-1}$ $\hat{V}$ weight-preserving gates acting over two qubits as in Definition 4.7. The qubits over which these gates act can be computed efficiently. Suppose gate V acts over qubit $i \in \{1, 2, \ldots, \log n\}$, where the index i runs over the qubits inside some group of $\log n$ qubits. Denote as $\tilde{V}$ the encoded version in the new circuit of gate V. The action of $\tilde{V}$ over basis states is defined as follows. Let $x^{(1)}$,$x^{(2)} \in \{0,1\}^{\log n}$ be computational basis states which differ only on the ith bit, for example, suppose $x^{(1)} = 0$ and $x_i^{(2)} = 1$. Let p and q be the qubit indices in the new circuit where $E(x^{(1)})$ and $E(x^{(2)})$ have a 1. Then, $\tilde{V}$ will act as gate $\hat{V}$ on qubits p and q. For each such pair $x^{(1)}$ and $x^{(2)}$, $\tilde{V}$ acts on the prescribed pair of qubits as $\hat{V}$. Thus, in total $\tilde{V}$ requires $2^{\log n-1}$ $\hat{V}$ gates. An example is illustrated in Fig. 4.6 where $n = 8$ and $k = 1$, each group has 3 qubits and is encoded as a group of 8 qubits.

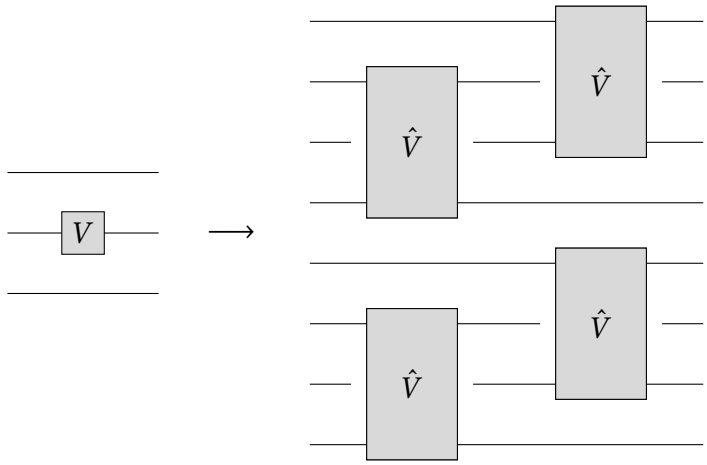

Figure 4.6: Example of mapping a one-qubit gate to gates acting on 8 qubits for $n = 8$ and $k = 1$. The discontinued lines are qubits that are not acted upon by the gates.

It is simple to check that this new circuit preserves the amplitudes of the original miniature circuit. Let $x = (x_1, x_2, \ldots, x_i, \ldots, x_{\log n}) \in \{0,1\}^{\log n}$ and $V^{(a)} = \sum_{r,s=0}^{1} v_{r,s} |r\rangle\langle s|$ the single-qubit unitary acting on qubit i. Then, the action of $V^{(i)}$

over a computational basis state is

$$V^{(i)} \left|x_1 \cdots x_i \cdots x_{\log n}\right\rangle = v_{0,x_i} \left|x_1 \cdots 0_i \cdots x_{\log n}\right\rangle + v_{1,x_i} \left|x_1 \cdots 1_i \cdots x_{\log n}\right\rangle .$$

where 0_i and 1_i denote a 0 or a 1 at the ith position respectively. The encoded version of V will act in a similar way by construction

$$\tilde{V}^{(i)} \left|E(x_1 \cdots x_i \cdots x_{\log n})\right\rangle = v_{0,x_i} \left|E(x_1 \cdots 0 \cdots x_{\log n})\right\rangle + v_{1,x_i} \left|E(x_1 \cdots 1 \cdots x_{\log n})\right\rangle .$$

For CNOT gates we need to consider two different cases, (i) the CNOT is acting between two qubits in the same group and (ii) the CNOT is acting between two qubits in different groups.

For case (i), suppose CNOT acts on control qubit i and target qubit j where i and j are in the same group of $\log n$ qubits. Let $x^{(1)}, x^{(2)} \in \{0,1\}^{\log n}$, if they differ in the jth qubit and the ith qubit is 1, then in the new circuit apply a SWAP between the qubits where $E(x_1)$ and $E(x_2)$ have 1s. We add as many SWAPS as pairs $x^{(1)}, x^{(2)}$ fulfilling this condition exist.

For case (ii), we consider control qubit i and target qubit j such that both qubits belong to different groups. To implement this gate in the weight-preserving circuit we will require two ancillae in the state $|01\rangle$. For every $x \in \{0,1\}^{\log n}$ such that qubit i is 1, then we apply a Fredkin gate with control qubit given by the position of 1 in $E(x)$ and with the ancilla qubits as targets. Such an example is given in Fig. 4.7. Before the SWAP network, the Fredkin gates are applied such that if any of the qubits are in state $|1\rangle$ then a SWAP network is applied. After this, the action of the Fredkin gates is undone. The SWAP network consists of SWAP operators acting over qubits as determined by the one-bit case mentioned earlier in our proof, these SWAP gates are controlled by the ancilla qubit.

Note that in the original circuit C the output is given by a single qubit. In the new weight-preserving circuit we can add two more extra ancillas in the state $|01\rangle$ which we assign as the output qubits. Then, after acting with the weight-preserving simulation of C, we can act with several controlled SWAP operators with the output qubits as target

and the control qubits corresponding to those that encode states of $\log n$ qubits with the output set to 1.

With the mapping in place, we have constructed a weight-preserving circuit and the last step is to implement measurements to check that each group of n qubits has only one qubit set to $|1\rangle$. In what follows, let $Q_i = \{q_1^{(i)}, \ldots, q_n^{(i)}\}$ be the set of qubits belonging to the ith group of qubits, where $i = \{1, \ldots, k\}$. To check that each Q_i is a weight-1 state, we can include $k(n+1)$ ancillas, which we also group into k sets of n qubits and denote as $A_j = \{a_1^{(j)}, \ldots, a_{n+1}^{(j)}\}$ the jth group of n qubits. First, initialize each A_j in the weight-1 state $|1000\cdots0\rangle$. Next, we will use the qubits in A_i to count the weight of the state in the Q_i register. We construct the following weight-preserving circuit acting between sets A_i and Q_i for each $i \in \{1, \ldots, k\}$. Act with a controlled SWAP on qubits $a_1^{(i)}, a_2^{(i)}$ as targets and qubit $q^{(i)_1}$ as control which we define as $\mathrm{CSWAP}_{q_1,a_1,a_2}$ (for simplicity, we surpress the index i from now on). Then, act with the gate $\mathrm{CSWAP}_{q_2,a_1,a_2} \cdot \mathrm{CSWAP}_{q_2,a_2,a_3}$. We act in the same way with succesive qubits in the set Q_i; for each qubit q_j we act with $\mathrm{CSWAP}_{q_i,a_1,a_2} \cdot \mathrm{CSWAP}_{q_i,a_2,a_3} \cdots \mathrm{CSWAP}_{q_j,a_j,a_{j+1}}$. Once applied this circuit, we need to only measure the qubit a_2 which tells us whether the weight of the state in Q_i is 1. Finally, to measure whether all $a_2^{(i)}$ are in the state 1, we add two ancillas in state $|01\rangle$ and act with CSWAP controlled by each $a_2^{(i)}$ and target the two new ancillas, such that we get the state $|10\rangle$ if all $a_2^{(i)}$ are in 1 are $|01\rangle$ otherwise. This construction requires the ancilla states to have in total weight-$(k+1)$ which together with the input state and two more ancillas for the CNOT gates and other two for the output qubits gives a total of weight-$2(k+2)$ with $kn + k(n+1) + 6$ qubits. We have then a new weight-preserving circuit C' which is satisfiable by a weight-$2(k+1)$ state if and only if the original circuit C is satisfiable. Moreover, C' simulates C faithfully (at each step the amplitudes are preserved). Let us now show that the reduction works as intended. For completeness, since the simulation is faithfull, then our new weight-preserving circuit preserves the completeness. For soundness, suppose for all states $|\psi\rangle$ we have that $\Pr(C \text{ accepts } |\psi\rangle) \leq s$. Let $|\phi\rangle$ be a $kn + k(n+1) + 4$ qubit state, where the witness has been supplied by the prover and the ancillas have been set as described above. Note that the only way for the prover

to cheat is by breaking the encoding we have delineated above, thus we introduce the decomposition $|\phi\rangle = \alpha\,|\xi_1\rangle + \beta\,|\xi_2\rangle$, where $|\xi_1\rangle$ is a state respecting the encoding and $|\xi_2\rangle$ is a state that doesn't respect the encoding above. Thus, defining Π_{10} as the projector on the output qubits onto the state $|10\rangle$, we have that

$$\begin{aligned}\Pr\left(C' \text{ accepts } |\phi\rangle\right) &= \|\Pi_{10}C'\,|\phi\rangle\|^2 \\ &= |\alpha|^2\|\Pi_{10}C'\,|\xi_1\rangle\|^2 \end{aligned} \tag{4.13}$$

Since $|\alpha|^2 < 1$ when the prover is cheating, then the accepting probability only diminishes when this is the case. Thus $\Pr\left(C' \text{ accepts } |\phi\rangle\right) \leq s$. This implies the $\mathsf{QM}_{c,s}[1]$-hardness of WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) ■

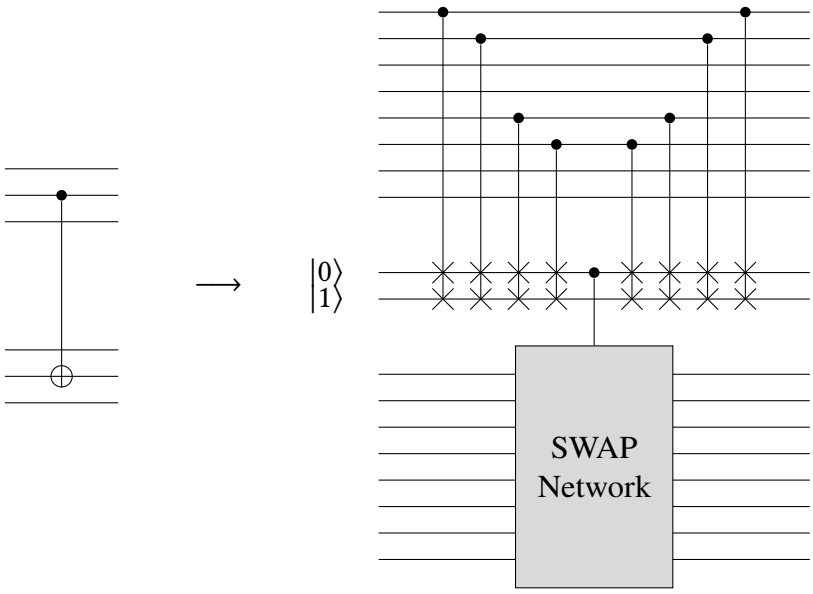

Figure 4.7: Example of mapping a CNOT gate acting between two different groups for $n = 8$ and $k = 2$. The Fredkin gates implement the control and the SWAP network implements the bit flip part.

We have thus shown the following corollary.

Corollary 4.39. MINI-QCSAT(c, s) *is in* FPQT *if* WEIGHT-k WEIGHT-PRESERVING QUANTUM CIRCUIT SATISFIABILITY(c, s) *is in* FPQT.

Now we can use the reduction from the proof of Theorem 4.4 (Corollary 4.22) to reduce the weigh-preserving circuit to an instance of the almost spatially sparse weight-k local Hamiltonian and thus also can be reduced to the weight-k ℓ-local Hamiltonian.

Theorem 4.40. *If for all a, b such that $b - a > 1/\mathrm{poly}(n)$,* WEIGHT-$k$ ℓ-LOCAL HAMILTONIAN(a, b) *is in* FPQT *then* QETH *is false.*

4.5 Conclusion

Throughout this chapter we have shown that the tractability of the parameterized local Hamiltonian problem is connected to the existence of quantum subexponential algorithms for circuit satisfiability problems. The big open question is whether this class is QW[1]-complete. Nonetheless, note that a reason to believe that QW[1] is an itnractable class is precisely this connection to the QETH and QCETH. Another important open question is related to success amplification in the class QW[1], the fact that we have not found a method to perform this amplification in constant depth is an interesting problem which could help proving intractability for other parameterized problems. Finally, it is important to note that in another work [CCZZ21] a different version of what we call the QCETH (the authors define it in their paper as QETH) has been given. The version in [CCZZ21] considers the 3-SAT, which is also considered in the ETH. In the classical case it is possible to prove the so-called Sparsification Lemma [IPZ01] which allows to show results in fine-grained complexity when reducing from 3-SAT to other problems. It would be interesting to prove a similar lemma in the quantum case (either Quantum Circuit Satisfiability or for the Local Hamiltonian problem) as this would allow for similar reductions in fine-grained complexity.

Part II

Fermion Sampling

Chapter 5

Fermion Sampling

This chapter is based on the published paper [ODMZ22] written in collaboration with Michal Oszmaniec, Ninnat Dangniam and Zoltan Zimboras. We introduce a new sampling scheme for quantum advantage based on Fermionic linear optics supplied with magic states which allow to prove hardness guarantees comparable to other schemes.

My contribution to this work consisted of the contribution of ideas and writing several parts in [ODMZ22]. In this chapter I have rewritten or expanded parts in [ODMZ22] which have been included here, moreover, several parts where I did not contribute have not been included but I cite the corresponding results when used. In the paper, I helped writing sections of Appendix E.3 which correspond to Appendix A.2 in this thesis, I also was involved in proofs of earlier versions of this part, but the final version includes many parts written by my coauthors and thus has been included as an appendix. The part corresponding to Sections 5.6 and 5.7 was written by my coauthors with some contributions by myself, but the version presented in this thesis is rewritten and expanded by me, in fact I obtain different error bounds than in our paper as I have followed a different path in certain calculations. Although these error bounds are not improved, these changes help make the calculations clearer. In the paper I contributed to an earlier version of the proof of Theorem 5.9 and in this thesis I have detailed some of the calculations that are present in the paper. The figures in Appendix A were done by

my coauthors but all the figures in this chapter (excluding the appendix) were done by me. The theorem, lemma and corollary statements that appear in this thesis are directly taken from [ODMZ22] unless otherwise specified, but as mentioned previously the explanations and discussions have been expanded by myself when appropriate. Finally, I contributed with some numerics which have not been included in this thesis but have been presented in [ODMZ22].

5.1 Introduction

As discussed in the introductory chapter of this thesis, one line of research to obtain quantum advantage over classical computers focuses on studying sampling problems. The main idea is to generate a random quantum circuit and sample from it such that the output probability distribution is hard to sample from for classical machines. Such separation results are stated under certain assumptions, including the non-collapse of the polynomial hierarchy. In this chapter we will show that such sampling schemes can be implemented with Fermionic particles and either random number-preserving or parity-preserving evolutions which we denote as Fermion Sampling.

In [AA11] Aaronson and Arkhipov introduce the so-called Boson sampling problem which is based on linear optics circuits with non-adaptive measurements and with n photons in m different optical modes. Other proposals based on sampling include IQP sampling [BJS11, BMS16], Random Circuit Sampling (RCS) [BIS+18, BFNV19, Mov19], quantum Fourier sampling [FU16] and many other schemes [Mor17, BVHS+18, BFK18, PBG20, YJS19]. Inspired by these previous works, quantum sampling schemes have been implemented experimentally. A version of RCS has been demonstrated in a quantum processor of 53 superconducting qubits with circuits of depth 20 [AAB+19]. The fact that these experiments can be implemented with near-term devices is an attractive feature and motivates the search for rigorous guarantees of hardness for classical computers.

To give complexity-theoretic support to such guarantees, all sampling schemes proceed in a similar way. Assuming the existence of an efficient classical algorithm S sampling from a distribution which approximates the output probabilities of a random quantum circuit in total variation distance, it is shown (based on certain assumptions which we detail later) that the Polynomial Hierarchy collapses to the third level. To prove this implication, it is shown using Stockmeyer's algorithm [Sto85] that given S, then it is possible to compute approximately the output probabilities of the random quantum circuit C for a large fraction of instances. This corresponds to an additive approximation up to an exponential factor, which can be reduced to an approximation in multiplicative error by showing the anticoncentration of the output probabilities. We remark that it is enough to compute a single output probability from a random circuit (for example the output probability $p_0(C)$ of the output 0^n) due to a property called the hiding property. The final step involves the conjecture that giving average-case multiplicative approximations is #P-hard. More detail for this argument in the context of our Fermionic sampling scheme is given in Section 5.5, but similar proofs are given for other sampling schemes

Recent work [BFNV19, Mov19, BFLL22, KMM22] has focused on showing the conjecture that average-case approximations to output probabilities of random circuits is in fact #P-hard. The conjecture requires to show that $p_0(C)$ is #P-hard to approximate up to an additive error of 2^{-n}. In [BFNV19] it was shown that if computing the output probability $p_0(C)$ is #P-hard in the worst case then it is #P-hard to compute it on the average-case up to additive error $2^{-\mathrm{poly}(n)}$. Proving this involved using polynomial interpolation techniques which were then improved with the so-called Cayley path [Mov19]. We will implement the Cayley path from [Mov19] for Fermion Sampling and show similar hardness guarantees. In [BFLL22] hardness has been shown for approximations up to $2^{-O(m \log m)}$, where m is the number of gates in the RCS circuit, analogous results have been shown for Boson Sampling. Recently in [Kro22] it is claimed that the error can be up to $2^{-O(m)}$. Although we have not applied this result to Fermion Sampling, we expect that similar results could be obtained.

In our work [ODMZ22] we have proposed a sampling scheme based on Fermionic linear optics (FLO), both in the passive (particle number conserving) and active (parity number conserving or matchgates) case. Previous work [Iva17, HJKS20] has studied the computational complexity of sampling from active FLO when supplied with additional resources. Concretely, the authors show that when certain magic inputs are supplied, then it becomes #P-hard to compute output probabilities. It remained open whether Fermionic sampling schemes could achieve similar hardness guarantees as in Boson sampling or RCS. In the case of RCS it has been shown the average-case hardness of approximating output probabilities and also anticoncentration. Note that such proof of anticoncentration is lacking in Boson sampling. In our work we find that Fermion sampling combines the advantages of these two sampling schemes, we show average-case hardness for approximating output probabilities, anticoncentration and moreover propose some efficient certification protocols for the circuits. In the case of average-case hardness, we show that this hardness is present up to errors of $2^{-\Theta(N^7)}$, where N is proportional to the number of Fermions. Although this is worse than RCS, we believe that more recent techniques mentioned in the previous paragraph could give better bounds.

We begin in Section 5.2 by giving some background on Fermions and also set the notation for the rest of the chapter. In Section 5.3 we define the Fermion sampling scheme. Then in Section 5.4 we show that the random circuits involved in this scheme have the property of anticoncentration. We will apply this property in the proof for hardness of classically sampling FLO circuits. Section 5.5 will show that an efficient classical sampler for the random FLO circuit is unlikely to exist using the anticoncentration property and assuming some plausible complexity-theoretical conjectures. In Section 5.6 and Section 5.7 we focus in one of this conjecture, namely, that it is #P-hard to approximate output probabilities in the average-case. Having shown this, in the next two chapters we focus on giving support to the conjecture regarding average-case hardness for computing output probabilities. Section 5.6 introduces the Cayley path

method and gives some technical results which then are applied in Section 5.7 to show the average-case hardness of approximations.

5.2 Some background and notation

Here we introduce some necessary background on Fermionic systems and some notation which will be necessary for this chapter. In this chapter we will denote a Hilbert space as $\mathcal{H}$. To denote density matrices on this Hilbert space we denote them as $\Psi = |\Psi\rangle\langle\Psi|, \Phi = |\Phi\rangle\langle\Phi|$ etc. We also define $\mathcal{D}(\mathcal{H})$ as the set of all (possibly mixed) quantum states on $\mathcal{H}$. The group of unitary operators over the Hilbert space $\mathcal{H}$ is denoted as $\mathrm{U}(\mathcal{H})$.

The single particle Hilbert space for a Fermion on d modes is given by $\mathbb{C}^d$. To represent quantum states with more than a single particle we will need the notion of the wedge product $\wedge$. Consider a basis $\{|e_i\rangle\}_{i=1}^d$ for the vector space $\mathbb{C}^d$, then define for $i_1, i_2 \in \{1, 2, \cdots, d\}$

$$\left|e_{i_1}\right\rangle \wedge \left|e_{i_2}\right\rangle = \left|e_{i_1}\right\rangle \otimes \left|e_{i_2}\right\rangle - \left|e_{i_2}\right\rangle \otimes \left|e_{i_1}\right\rangle, \tag{5.1}$$

which is an antisymmetrized version of the tensor product. We define the vector space $\mathbb{C}^d \wedge \mathbb{C}^d = \mathrm{span}\{\left|e_{i_1}\right\rangle \wedge \left|e_{i_2}\right\rangle\}$ and more generally $\bigwedge^n(\mathbb{C}^d) = \mathrm{span}\{\left|e_{i_1}\right\rangle \wedge \cdots \wedge \left|e_{i_n}\right\rangle\}$ where

$$\left|e_{i_1}\right\rangle \wedge \cdots \wedge \left|e_{i_n}\right\rangle = \frac{1}{\sqrt{n!}} \sum_{\sigma \text{ permutation}} sgn(\sigma) \left|e_{\sigma(i_1)}\right\rangle \otimes \cdots \otimes \left|e_{\sigma(i_n)}\right\rangle \tag{5.2}$$

with $sgn(\sigma)$ the sign of permutation σ. From the single particle space we can define the d-mode Fock space

$$\mathcal{H}_{\mathrm{Fock}}(\mathbb{C}^d) = \bigoplus_{n=0}^{d} \bigwedge^n (\mathbb{C}^d), \tag{5.3}$$

where $\bigwedge^n(\mathbb{C}^d)$, is the totally anti-symmetric subspace of $(\mathbb{C}^d)^{\otimes n}$ describing states consisting of exactly n fermions, and $\bigwedge^0(\mathbb{C}^d) = \mathrm{span}_{\mathbb{C}}(|0_F\rangle)$, where $|0_F\rangle$ is the Fock vacuum.

Over the Fock space $\mathcal{H}_{\text{Fock}}$ we define the creation and destruction operators $f_j^\dagger$ and f_j, respectively, where $j = 1, 2, \ldots, d$. These operators are defined by the anti-commutation relations $\{f_j, f_k^\dagger\} \equiv f_j f_k^\dagger + f_k^\dagger f_j = \delta_{j,k}$ and $\{f_j, f_k\} = \{f_j^\dagger, f_k^\dagger\} = 0$, with $\delta_{j,k}$ being the Kronecker delta symbol.

A basis over $\mathcal{H}_{\text{Fock}}(\mathbb{C}^d)$ is the Fock basis given by

$$|\mathbf{x}\rangle := (f_1^\dagger)^{x_1}(f_2^\dagger)^{x_2}\cdots(f_d^\dagger)^{x_d}|0_F\rangle \tag{5.4}$$

for any $\mathbf{x} \in \{0,1\}^d$, and where $(f_1^\dagger)^0 = \mathbb{1}$. We define also $[d] = \{1, 2, 3, \cdots, d\}$. Given a subset $\mathcal{X} \subset [d]$, we define the state $|\mathcal{X}\rangle$ as the Fock basis state $|\mathbf{x}\rangle$ such that $x_j = 1$ if $j \in \mathcal{X}$ and $x_j = 0$ otherwise. Moreover define $\binom{\mathcal{X}}{k}$ to be the collection of subsets of finite set $\mathcal{X}$ of size k (we shall assume the convention $\binom{\mathcal{X}}{k} = \emptyset$ if $|\mathcal{X}| < k$. For $\mathcal{X} = \{a_1, \ldots, a_n\}$ with $a_i < a_j$ if $i < j$, define the state

$$\begin{aligned} |\mathcal{X}\rangle &= |a_1\rangle \wedge |a_2\rangle \wedge \ldots \wedge |a_n\rangle \\ &= \frac{1}{\sqrt{n!}} \sum_{i_1,\ldots i_n=1}^{n} \epsilon_{i_1,i_2,\ldots,i_n} |a_{i_1}\rangle \otimes |a_{i_2}\rangle \otimes \cdots |a_{i_n}\rangle. \end{aligned} \tag{5.5}$$

where $\epsilon_{k_1,k_2,\ldots,k_n}$ is the Levi-Civita symbol. States such as $|\mathcal{X}\rangle$ are known as *Slater determinants*. These are states of the form $|\Psi\rangle = |\xi_1\rangle \wedge |\xi_2\rangle \wedge \ldots \wedge |\xi_n\rangle$, where $\{|\xi_i\rangle\}_{i=1}^n \subset \mathbb{C}^d$ is a set of orthonormal vectors of the one-particle Hilbert space $\mathbb{C}^d$. The overlap between any two Slater determinant states, $|\Psi\rangle = |\xi_1\rangle \wedge |\xi_2\rangle \wedge \ldots \wedge |\xi_n\rangle$ and $|\Phi\rangle = |\phi_1\rangle \wedge |\phi_2\rangle \wedge \ldots \wedge |\phi_n\rangle$, can be expressed by the simple determinant formula

$$\langle\Psi|\Phi\rangle = \det C, \quad C_{i,j} = \langle\xi_i|\phi_j\rangle. \tag{5.6}$$

The way that we have defined $|\mathcal{X}\rangle$ shows that to each Fock basis state there is a set $\mathcal{X} \subseteq [d]$, in physics when we work with states of the form as in Eq. (5.5) it is sometimes called "first quantization" whereas when we work with Fock states this is sometimes called "second quantization". As an example, consider the Fock state $|1100\rangle$ with 4 modes and 2 fermions in the first 2 modes. We have that $|1100\rangle = f_1^\dagger f_2^\dagger |0000\rangle$ in second quantization and $|1100\rangle = \frac{1}{\sqrt{2}}(|\phi_1\rangle|\phi_2\rangle - |\phi_2\rangle|\phi_1\rangle)$ for a basis $|\phi_i\rangle$ of $\mathbb{C}^4$ in first quantization.

Our work will be concerned with passive and active FLO operations. Both of these are known to be classically simulable when the input states are Gaussian [Val01, JM08, TD02, DT05], we will explain what this means later.

5.2.1 Passive FLO

Passive FLO operations correspond to particle number preserving operations. These correspond to unitary transformations of the form

$$U = e^{i \sum_{n,m} K_{jk} f_j^\dagger f_k} \tag{5.7}$$

with K_{nm} is a Hermitian matrix. The action of these unitaries over the creation and destruction operators can be easily computed. This is given by

$$U f_k U^\dagger = \sum_j V_{kj} f_j \tag{5.8}$$

where V is a unitary given by e^{-iK}.

At a more abstract level, passive FLO are the irreducible representation of the group $\mathrm{U}(d)$ in the Hilbert space $\bigwedge^n(\mathbb{C}^d)$.

$$\Pi_{\mathrm{pas}} : \mathrm{U}(d) \longrightarrow \mathrm{U}\left(\bigwedge^n(\mathbb{C}^d)\right), \tag{5.9}$$

$$U \longmapsto U^{\otimes n}\big|_{\bigwedge^n(\mathbb{C}^d)}. \tag{5.10}$$

where $U^{\otimes n}$ is n copies in a tensor product of the unitary over the single particle space.

5.2.2 Active FLO

Active FLO operations correspond to those that preserve the parity of states. We introduce the so called Majorana operators

$$m_{2j-1} = f_j + f_j^\dagger, \qquad m_{2j} = -i\,(f_j - f_j^\dagger), \tag{5.11}$$

with anti-commutation relations $\{m_j, m_k\} = 2\,\mathbb{1}\delta_{j,k}$. With this we can define the parity operator $Q = i^d \prod_{i=1}^{2d} m_i$ in $\mathcal{H}_{\mathrm{Fock}}(C^d)$. We denote by $\mathcal{H}^+_{\mathrm{Fock}}(\mathbb{C}^d)$ the eigenvalue +1

subspace of Q, which is the span of the Fock basis states $|\mathbf{n}\rangle$ having even number of particles. The active FLO operations (or Gaussian operations) are those that are generated by Hamiltonians quadratic on the Majorana operators, i.e., $U = \mathrm{e}^{iH}$, where

$$H = \frac{i}{4} \sum_{j,k=1}^{2d} A_{j,k}\, m_j\, m_k, \tag{5.12}$$

and $A = -A^T \in \mathbb{R}^{2d\times 2d}$. While the passive FLO transformations form a representation of the group $\mathrm{U}(d)$, the active FLO form a representation of the group $\mathrm{SO}(2d)$.

$$\Pi_{\mathrm{act}} : \mathrm{SO}(2d) \longrightarrow \mathrm{U}(\mathcal{H}_{\mathrm{Fock}}(\mathbb{C}^d)) \tag{5.13}$$

$$O \longmapsto \exp\left(\frac{1}{4} \sum_{i,j=1}^{2d} [\log(O)]_{ij}\, m_i m_j\right), \tag{5.14}$$

A single Majorana operator evolves under an active FLO transformation as follows

$$U^\dagger\, m_j\, U = \sum_{k=1}^{2d} O_{jk}\, m_k, \tag{5.15}$$

where $U = e^{-iHt}$ with $H = \frac{i}{4}\sum_{j,k=1}^{2d} A_{j,k} m_j m_k$ and and $O = e^{-A} \in \mathrm{SO}(2d)$. $\mathcal{G}_{\mathrm{pas}}$ and $\mathcal{G}_{\mathrm{act}}$ will refer to the passive and active fermionic linear optical gates.

5.2.3 Some facts about $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$

For the proof of some of our results we will require some basic facts about $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$, in particular the decompositions shown here will be applied in Section 5.6.3. It is a well-known fact from linear algebra that for skew-hermitian operators $A \in \mathfrak{u}(d)$ there is a unitary $U \in \mathrm{U}(d)$ such that

$$UAU^\dagger = \sum_{j=1}^{d} \phi_j X_j\,, \tag{5.16}$$

where $X_j = i\,|j\rangle\langle j|$. Similarly, for any $A \in \mathfrak{so}(2d)$ there exist $O \in \mathrm{SO}(2d)$ such that

$$OAO^T = \sum_{j=1}^{d} \phi_j \tilde{X}_j \tag{5.17}$$

where $\tilde{X}_j = |2j\rangle\langle 2j-1| - |2j-1\rangle\langle 2j|$ is the generator for the jth block. These statements have analogues on the level of elements of the group. Every unitary U can be transformed into a diagonal form

$$\mathrm{diag}(e^{i\phi_1}, e^{i\phi_2}, \ldots, e^{i\phi_d}) = \exp\left(\sum_{j=1}^{d} \phi_j X_j\right) . \tag{5.18}$$

For elements $\mathrm{SO}(2d)$, the block diagonalization amounts to the geometric fact that any $2d$-dimensional rotation can be decomposed into d independent planar rotations of the form $\exp\left(\sum_{j=1}^{d} \phi_j \tilde{X}_j\right) = \sum_{j=1}^{d} \exp\left(\phi_j \tilde{X}_j\right)$ as the exponential preserves the block diagonal form. Note that $\tilde{X}_j = -iY_j$ where Y_j is acting as the Pauli Y operator on the relevant subspace $\mathrm{span}(|2j-1\rangle, |2j\rangle)$. We have then that $\exp\left(\phi_j \tilde{X}_j\right) = \exp\left(-i\phi_j Y_j\right) = \cos\left(\phi_j\right)\mathbb{1} - i\sin\left(\phi_j\right) Y_j = \cos\left(\phi_j\right)\mathbb{1} + \sin\left(\phi_j\right)\tilde{X}_j$.

5.2.4 Haar measure

By virtue of the compactness of the groups $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$, we can define a unique normalized integration measure invariant under any group translation denoted as the Haar measure. We will denote this measure by μ_G where G is one of the symmetry groups above. Invariance of μ_G means that for any measurable subset $A \subset G$ and any $h \in G$, we have that

$$\mu(hA) = \mu(Ah) = \mu(A) . \tag{5.19}$$

The above condition to the level of expectation values (averages) reads

$$\int_G d\mu(g) f(gh) = \int_G d\mu(g) f(hg) = \int_G d\mu(g) f(g) . \tag{5.20}$$

where f is any integrable function on G and $h \in G$. We will denote by ν_{pas} the distribution of the unitaries $V = \Pi_{\mathrm{pas}}(U)$, where $\mathrm{U} \sim \mu_{\mathrm{U}(d)}$ and by ν_{act} distribution of the unitaries $\Pi_{\mathrm{act}}(O)$, where $O \sim \mu_{\mathrm{SO}(2d)}$.

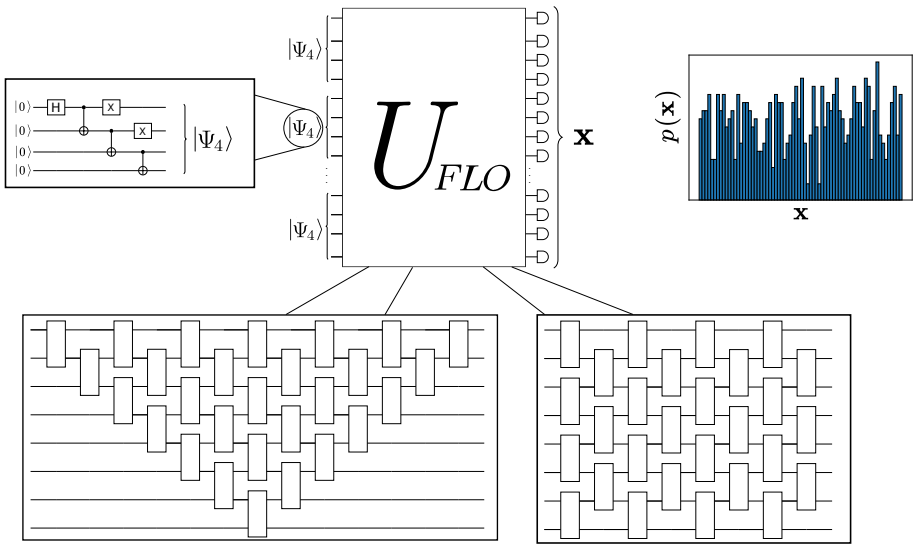

Figure 5.1: The setup considered in our work. We run an FLO circuit U_{FLO} (passive or active) with input state $|\Psi_{\text{in}}\rangle = |\Psi_4\rangle^{\otimes N}$ and sample bitstrings $\mathbf{x}$ with the probability distribution $p(\mathbf{x})$ induced by the circuit. Using Jordan-Wigner transformation that encodes fermions in qubits, the state $|\Psi_4\rangle$ can be easily prepared as shown in the inset to the left. The decomposition of the circuits into elementary gate set can be realized by the fermionic analogues of existing layouts for linear optical networks.

5.3 Fermion Sampling scheme

Having laid out the background and notation for this chapter, we proceed to define the Fermion Sampling scheme in this section. Our sampling scheme is based on magic inputs that will supply the necessary resources to make the system hard to simulate classically. Consider $d = 4N$ fermionic modes, the input to the circuit is given by the tensor product of N states $|\Psi_4\rangle = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle)$. The input is then given by $|\Psi_{\text{in}}\rangle = |\Psi_4\rangle^{\otimes N}$. Similar states have been used in previous works [Bra06, Iva17, HJK+19]. After the injection of the initial states, a random FLO (passive or active) circuit is applied. This random circuit is generated by decomposing a general FLO operation into a triangular or brickwork layout as is common for FLO (For details, check Appendix A of [ODMZ22]). The choice of the FLO operation V is done via the probability distributions ν_{pas} and ν_{act} induced from the Haar measures on $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$, respectively. After the circuit, particle number measurements give the output of the circuit.

The output probability of measuring $|\mathbf{x}\rangle$ for an input Ψ_{in} and circuit V is given by

$$p_{\mathbf{x}}(V, \Psi_{\text{in}}) = |\langle \mathbf{x}|V|\Psi_{\text{in}}\rangle|^2 \,, \tag{5.21}$$

where the output bitstring satisfies $|\mathbf{x}| = 2N$ and $|\mathbf{x}|$-even for passive FLO and active FLO respectively. In the next sections we will be concerned on showing that the existence of an efficient classical algorithm sampling from a distribution close to the one defined by $p_{\mathbf{x}}(V, \Psi_{\text{in}})$ is unlikely.

5.4 Anticoncentration of FLO circuits

As discussed in the introduction, we want to show that classically sampling from distributions close to those induced by the Fermion sampling scheme is unlikely based on plausible conjectures. An important ingredient in showing this is proving anticoncentration. Intuitively, we want the output probabilities of randomly chosen circuits to not be too small on average. This property has been shown in IQP circuits [BMS16], RCS [BFNV19] and has been conjectured for Boson sampling [AA11]. In this section we show that random FLO circuits have this anticoncentration property. This will be used in Section 5.5 to prove that approximate classical sampling of FLO circuits implies the capacity to approximately compute output probabilities in Theorem 5.9.

Theorem 5.1 (Anticoncentration for fermionic linear optical circuits initialized in product of magic states). *Let $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ and let $\mathcal{H}_{\text{act}} = \mathcal{H}^{+}_{\text{Fock}}(\mathbb{C}^{4N})$ be Hilbert spaces describing $2N$ Fermions in $4N$ modes and positive parity Fermions in $4N$ modes. Let $\mathcal{G}_{\text{pas}}$ and $\mathcal{G}_{\text{act}}$ be respectively passive and active FLO transformations acting on the respective Hilbert spaces and distributed according to the uniform measures ν_{pas} and ν_{act} (see Section 5.2). Let $|\Psi_{\text{in}}\rangle$ be the initial state to which both families of circuits are applied. Then, for every $\mathbf{x}$ of Hamming weight $|\mathbf{x}| = 2N$ we have*

$$\Pr_{V\sim\nu_{\text{pas}}}\left(p_{\mathbf{x}}(V, \Psi_{\text{in}}) > \frac{\alpha}{|\mathcal{H}_{\text{pas}}|}\right) > \frac{(1-\alpha)^2}{C_{\text{pas}}} \,, \tag{5.22}$$

where $C_{\text{pas}} = 5.7$ *and* $|\mathcal{H}_{\text{pas}}| = \binom{4N}{2N}$. *Moreover, for every* $\mathbf{x}$ *with even Hamming weight we have*

$$\Pr_{V\sim\nu_{\text{act}}}\left(p_{\mathbf{x}}(V,\Psi_{\text{in}}) > \frac{\alpha}{|\mathcal{H}_{\text{act}}|}\right) > \frac{(1-\alpha)^2}{C_{\text{act}}}\,, \tag{5.23}$$

where $C_{\text{act}} = 16.2$ *and* $|\mathcal{H}_{\text{act}}| = 2^{4N}/2$.

Proof. The starting point for proving anticoncentration in many of the sampling schemes is the Paley-Zygmund inequality.

Theorem 5.2 (Paley-Zygmund Inequality). *Given a nonnegative bounded random variable* X, *for* $0 < \alpha < 1$ *we have*

$$\Pr_X(X > \alpha\mathbb{E}X) \geq (1-\alpha)^2\frac{(\mathbb{E}X)^2}{\mathbb{E}X^2}\,. \tag{5.24}$$

By setting $X = p_{\mathbf{x}}(V,\Psi_{\text{in}}) = |\langle\mathbf{x}|V|\Psi_{\text{in}}\rangle|^2$, with $V\sim\nu_{\text{pas}}$ or $V\sim\nu_{\text{pas}}$ we can obtain expressions similar to Eq. (5.22) and Eq. (5.23), where now we need to compute the moments of X, i.e., we need to compute $\mathbb{E}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^k\right]$ for $k = 1, 2$. Any passive or active linear circuit V in $\mathcal{G}_{\text{pas}}$ and $\mathcal{G}_{\text{act}}$ represents group elements in $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$ and thus the Haar measure in this groups induces a uniform distribution over the corresponding linear circuits. We then write

$$\mathbb{E}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^k\right] = \int_G d\mu(g)\left[\mathrm{tr}\Big(|\mathbf{x}\rangle\langle\mathbf{x}|\,\Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\Big)\right]^k\,, \tag{5.25}$$

where μ is the Haar measure on a Lie group G, and Π is a unitary representation of G in a suitable Hilbert space $\mathcal{H}$. The case of passive FLO corresponds to $G = \mathrm{U}(4N)$ and $\Pi = \Pi_{\text{pas}}$ while for active FLO we have $G = \mathrm{SO}(8N)$ and $\Pi = \Pi_{\text{act}}$. In what follows we compute the first and second moment to prove the anticoncentration inequality.

We start by computing the first moment. Both groups are irreducibly represented in Hilbert spaces $\mathcal{H}_{\text{act}}$ and $\mathcal{H}_{\text{pas}}$ by virtue of Schur's lemma and unitaries $\Pi(g)$ forming a 1-design. To make this statement precise, let us first define a G-linear map and use this to state Schur's lemma.

Definition 5.3 (G-linear map [Ser12]). Let G be a group and let ρ_V and ρ_W be irreducible representations of G on V and W respectively. A G-linear map $f : V \to W$ is

such that

$$\forall g \in G, \rho_W(g) \circ f = f \circ \rho_V(g).$$

Lemma 5.4 (Schur's Lemma [Ser12]). *Let V and W be vector spaces and let ρ_V and ρ_W be irreducible representations of a group G on V and W respectively. Then*

(1) *If V and W are not isomorphic, then there are no non-trivial G-linear maps between them.*

(2) *If $V = W$ finite dimensional over $\mathbb{C}$ and if $\rho_V = \rho_W$ then the only nontrivial G-linear maps are the identity and scalar multiples of the identity.*

Then we can show that

$$\int_G d\mu(g) \left[\Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\right] = \frac{1}{|\mathcal{H}|} \tag{5.26}$$

To obtain this equality we start by identifying the map $f : \mathcal{H} \to \mathcal{H}$ with $h_0 = \int_G d\mu(g) \left[\Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\right]$ and noting that this map is a G-linear map since given $\tilde{g} \in G$, we have that

$$\Pi(\tilde{g})h_0\Pi(\tilde{g})^\dagger = \int_G d\mu(g) \left[\Pi(\tilde{g}g)\Psi_{\text{in}}\Pi(\tilde{g}g)^\dagger\right] \tag{5.27}$$

$$= \int_G d\mu(g) \left[\Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\right] \tag{5.28}$$

$$= h_0. \tag{5.29}$$

This implies by Schur's lemma that $h_0 = \lambda\mathbb{1}$ for some $\lambda \in \mathbb{C}$. Note that $\text{Tr}\{h_0\} = 1$ and $\text{Tr}\{\mathbb{1}\} = |\mathcal{H}|$ which implies $\lambda = \frac{1}{|\mathcal{H}|}$ and the equality we wanted to prove. Consequently

$$\begin{aligned} &\mathop{\mathbb{E}}_{V\sim\nu}\left[p_{\mathbf{x}}(V, \Psi_{\text{in}})\right] \\ &\quad= \int_G d\mu(g) \left[\text{tr}\left(|\mathbf{x}\rangle\langle\mathbf{x}|\, \Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\right)\right] \\ &\quad= \text{tr}\left(|\mathbf{x}\rangle\langle\mathbf{x}| \int_G d\mu(g) \left[\Pi(g)\Psi_{\text{in}}\Pi(g)^\dagger\right]\right) = \frac{1}{|\mathcal{H}|}, \end{aligned} \tag{5.30}$$

where in the last equality we used the 1-design property and the fact that $|\mathbf{x}\rangle \in \mathcal{H}$ is a normalized vector.

To complete the proof of anticoncentration we need to bound the second moments of the output probabilities. In Proposition 5.5 we give a bound in terms of combinatorial factors. The proof of this is given in Appendix E of [ODMZ22] which involves using group theoretic techniques to rewrite the second moment.

Proposition 5.5 (Passive FLO second moment [ODMZ22]). *The second moment of the output probabilities* $\mathop{\mathbb{E}}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^2\right]$ *in the passive FLO is bounded by the following expression.*

$$\begin{aligned}\mathop{\mathbb{E}}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^2\right] \leq \frac{(2N+1)}{\binom{4N}{2N}^2(4N+1)}\Bigg[2\sum_{k=0}^{N-1}\frac{1}{\binom{2N}{k}}\sum_{l=0}^{\lfloor k/2\rfloor}\frac{N!}{l!(k-2l)!(N-k+l)!} \\ + \frac{1}{\binom{2N}{N}}\sum_{l=0}^{\lfloor N/2\rfloor}\frac{N!}{l!(N-2l)!l!}\Bigg] \\ = \frac{(2N+1)}{\binom{4N}{2N}^2(4N+1)}K_{\text{pas}}.\end{aligned} \tag{5.31}$$

Where recall that $4N$ *is the number of modes in the sampling scheme.*

Proposition 5.6 (Active FLO second moment [ODMZ22]). *The second moment of the output probabilities* $\mathop{\mathbb{E}}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^2\right]$ *in the active FLO case is bounded by the following expression.*

$$\begin{aligned}\mathop{\mathbb{E}}_{V\sim\nu}\left[p_{\mathbf{x}}(V,\Psi_{\text{in}})^2\right] \leq \frac{2}{\binom{8N}{4N}2^{8N}}\Bigg[2\sum_{q}^{N-1}C_{2q}\sum_{l=0}^{\lfloor\frac{q}{2}\rfloor}\frac{N!}{l!(q-2l)!(N-q+l)!}14^{q-2l} \\ + C_{2N}\sum_{l=0}^{N}\frac{N!}{(l!)^2(4N-2l)!}14^{N-2l}\Bigg] \\ = \frac{2}{\binom{8N}{4N}2^{8N}}K_{\text{act}},\end{aligned} \tag{5.32}$$

where

$$C_{2q} = \frac{(4q)!(8N-4q)!}{((4N)!)^2}\binom{4N}{2q}. \tag{5.33}$$

Where recall that $4N$ *is the number of modes in the sampling scheme.*

From these bounds on the second moment and considering that $\left|\mathcal{H}_{\text{pas}}\right| = \binom{4N}{2N}$ and $|\mathcal{H}_{\text{act}}| = 2^{4N}/2$, we obtain that

$$\begin{aligned}\frac{\mathbb{E}(X)^2}{\mathbb{E}(X^2)} &\geq \frac{1}{\left|\mathcal{H}_{\text{pas}}\right|^2}\frac{\binom{4N}{2N}^2(4N+1)}{(2N+1)}\frac{1}{K_{\text{pas}}} \\ &\geq \frac{1}{N}\frac{2N+1}{K_{\text{pas}}},\end{aligned} \tag{5.34}$$

for the passive case and in the active case

$$\begin{aligned}\frac{\mathbb{E}(X)^2}{\mathbb{E}(X^2)} &\geq \frac{1}{|\mathcal{H}_{\text{act}}|^2}\frac{\binom{8N}{4N}2^{8N}}{2}\frac{1}{K_{\text{act}}} \\ &= \frac{4}{2^{8N}}\frac{\binom{8N}{4N}2^{8N}}{2}\frac{1}{K_{\text{act}}} \\ &\geq \frac{4}{\sqrt{\pi 4N}}\frac{2^{8N}}{2}\frac{1}{K_{\text{act}}} \\ &= \frac{1}{\sqrt{\pi N}}\frac{2^{8N}}{K_{\text{act}}}.\end{aligned} \tag{5.35}$$

Where we used the fact that $\binom{8N}{4N} \geq 2^{8N}/\sqrt{\pi 4N}$. We see then that in the passive case if we prove that $\frac{K_{\text{pas}}}{2N+1} \leq \frac{C_{\text{pas}}}{N}$ and in the active case $\frac{K_{\text{act}}}{2^{8N}} \leq \frac{C_{\text{act}}}{\sqrt{\pi N}}$ then the anticoncentration result follows. This inequalities are proven in Appendix A.2, which completes the proof.

■

5.5 Hardness of sampling

In this section we show that the scheme described in Section 5.3 is hard to sample from classically, provided that certain conjectures are true. To show this we will use the anticoncentration property of FLO circuits shown in Section 5.4. We begin by defining the computational task which the classical sample must solve in Definition 1. Then we show that the existence of a classical sampler for Fermion Sampling implies the capacity to approximately compute the output probabilities of the Fermion Sampling circuits in Theorem 5.9. Finally, we show that this implies the collapse of the Polynomial Hierarchy under some plausible assumptions in Theorem 5.10.

Definition 1 (Fermion Sampling task from [ODMZ22]). Let $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ and let $\mathcal{H}_{\text{act}} = \mathcal{H}^{+}_{\text{Fock}}(\mathbb{C}^{4N})$ be Hilbert spaces describing $2N$ Fermions in $4N$ modes and positive parity Fermions in $4N$ modes. Let $\mathcal{G}_{\text{pas}}$ and $\mathcal{G}_{\text{act}}$ be passive and active FLO transformation. Let V be an FLO circuit on the Hilbert space $\mathcal{H}_{\text{pas}}$ or $\mathcal{H}_{\text{act}}$ and let $p(V)$ denote probability distribution $p_{\mathbf{x}}(V, \Psi_{\text{in}})$. Given a description of V, sample from a probability distribution $q(V)$ that is ϵ-close to $p(V, \Psi_{\text{in}})$ in l_1-norm (twice the total variation distance)

$$\|p(V) - q(V)\|_1 = \sum_{\mathbf{x}} |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| \leq \epsilon, \tag{5.36}$$

in time poly N.

It was realized in [AA11, BMS16] that, by virtue of Stockmeyer's theorem, the hardness of classically sampling from $p_{\mathbf{x}}(V, \Psi)$ up to an additive error is connected to the hardness of computing $p_{\mathbf{x}}(V, \Psi)$ for most instances of $\mathbf{x}$ and U. We now state Stockmeyer's theorem.

Theorem 5.7 (Stockmeyer's Theorem [Sto85]). *There exists a* $\mathsf{BPP}^{\mathsf{NP}}$ *machine which given any Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *and* $y \in \{0,1\}^n$ *can output a number* $\tilde{p}$ *such that*

$$(1-\epsilon)p \leq \tilde{p} \leq (1+\epsilon)p \tag{5.37}$$

with $\epsilon = \Omega(1/\text{poly}(n))$ *and* $p = \Pr_{x\sim unif}[f(x) = y]$.

This theorem allows to approximate output probabilites of classical samplers with a multiplicative error shown in Eq. (5.37). Note that this error can be also written as $|p - \tilde{p}| \leq \epsilon p$. In particular, this implies that the existence of a classical machine that performs the sampling task implies average-case approximation in the polynomial hierarchy which we show in Theorem 5.9. To prove this, we start by defining the notion of approximation in the average-case.

Definition 2. An algorithm O is said to give an (η, δ)-multiplicative approximation of $q_{\mathbf{z}}$ on average over the probability distribution $\mathcal{P}$ of inputs $\mathbf{z}$ iff O outputs $O_{\mathbf{z}}$ such that

$$\Pr_{\mathbf{z}\sim\mathcal{P}}\left[|O_{\mathbf{z}} - q_{\mathbf{z}}| \leq \eta q_{\mathbf{z}}\right] \geq 1 - \delta \tag{5.38}$$

We now prove the hiding property [AA11, BMS16, BFNV19], of FLO circuits. This will would allow us to focus on hardness of particular outcome probability.

Lemma 5.8 (Hiding property for FLO). *Consider a fixed state $|\mathbf{x}_0\rangle \in \mathcal{H}_{\text{pas}}$ ($\mathcal{H}_{\text{act}}$ resp.) then for any V passive FLO (active FLO resp.) and $|\mathbf{x}\rangle \in \mathcal{H}_{\text{pas}}$ ($\mathcal{H}_{\text{act}}$ resp.) there is a passive (active) FLO $V_{\mathbf{x}}$ such that $|\langle\mathbf{x}| V |\Psi_{\text{in}}\rangle|^2 = |\langle\mathbf{x}_0| V_{\mathbf{x}} |\Psi_{\text{in}}\rangle|^2$*

Proof. It is enough to show that given $\mathbf{x}$ there is $V_{\mathbf{x}}$ passive (active) FLO s.t. $V_{\mathbf{x}} |\mathbf{x}_0\rangle = |\mathbf{x}\rangle$ up to a global phase. In the passive case this is achieved with gates implementing fermionic swaps $U^{[i,j]}$ such that $U^{[i,j]} f_i^\dagger U^{[i,j]\dagger} = f_j^\dagger$ and $U^{[i,j]} f_j^\dagger U^{[i,j]\dagger} = f_i^\dagger$, the order in which they are applied is defined by $|\mathbf{x}\rangle$. The same can be accomplished in active FLO case with operators $-\mathrm{i} m_{2i} m_{2i+1}$ changing the number of fermions (but not parity) and quasi-braiding operators $U^{(p,q)}$ to exchange the majorana operators to the corresponding places. The quasi-braidings act on majorana operators as $U^{(p,q)} m_p (U^{(p,q)})^\dagger = m_q$, $U^{(p,q)} m_q (U^{(p,q)})^\dagger = m_p$ and $U^{(p,q)} m_x (U^{(p,q)})^\dagger = m_x$ when $x \neq p, q$. ∎

An additional ingredient used in the hardness guarantees for a quantum sampling advantage is anti-concentration which states that most output probabilities of a random circuit are sufficiently big so that the approximation error to computing the probabilities is small relative to the probabilities being computed. Both average-case hardness and anti-concentration provide robustness in terms of the error of approximation.

Theorem 5.9 (From approximate sampling to approximately computing probabilities). *Let $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ and let $\mathcal{H}_{\text{act}} = \mathcal{H}^{+}_{\text{Fock}}(\mathbb{C}^{4N})$ be Hilbert spaces describing $2N$ Fermions in $4N$ modes and positive parity Fermions in $4N$ modes. Consider in parallel passive FLO circuits and active FLO circuits acting on the input state $|\Psi_{\text{in}}\rangle$. If there is a classical algorithm C that performs Fermion Sampling as described in Definition 1 with the l_1-error $1/(64C)$, where C is the constant $C_{\text{pas}} = 5.7$ (resp. $C_{\text{act}} = 16.2$)*

appearing in the anticoncentration condition for passive FLO circuits (resp. active FLO circuits) in Theorem 5.1.

Then there is an algorithm in $\mathsf{BPP}^{\mathsf{NP}}$ *that approximates the probability* $p_{\mathbf{x}_0}(V, \Psi_{\text{in}})$ *for an arbitrary but fixed outcome* $\mathbf{x}_0$ *up to multiplicative error* $1/4 + o(1)$ *on* $1/(8C)$ *fraction of FLO circuits drawn from the distribution* $\nu = \nu_{\text{pas}}$ *for passive FLO circuits (resp.* ν_{act} *for active FLO circuits.)*

Proof. We will consider in parallel active and passive FLO circuits. For pasive FLO we have $\mathcal{H} = \mathcal{H}_{\text{pas}}$ and $\nu = \nu_{\text{pas}}$ while for active FLO we have $\mathcal{H}_{\text{act}}$ and $\nu = \nu_{\text{act}}$. With the fixed input $|\Psi_{\text{in}}\rangle = |\Psi_4\rangle^{\otimes N}$, we write $p_{\mathbf{x}}(V) = |\langle \mathbf{x}|V|\Psi_{\text{in}}\rangle|^2$ for the probability of outcome $\mathbf{x}$ (we assume that $|x\rangle \in \mathcal{H}$), and $p(V)$ for the output probability distribution of a circuit V. Suppose that there exists a classical sampler C that performs Fermion Sampling for a fixed but arbitrary FLO circuit V, and denote by $q(V)$ the distribution from which C samples.

Then for a given $\mathbf{x}$, by Stockmeyer's approximate counting algorithm [Sto85], a BPP^{NP} machine with an oracle access to C can produce a multiplicative estimates $\tilde{q}_{\mathbf{x}}(V)$ of $q_{\mathbf{x}}(V)$ such that

$$|q_{\mathbf{x}}(V) - \tilde{q}_{\mathbf{x}}(V)| \leq \frac{q_{\mathbf{x}}}{\text{poly}\, N} \tag{5.39}$$

for every $\mathbf{x}$. We will show that $\tilde{q}_{\mathbf{x}}(V)$ is also close to $p_{\mathbf{x}}(V)$ for most $\mathbf{x}$ and V that anti-concentrate.

Judiciously applying the triangle inequality, we have that

$$\begin{aligned}
&|p_{\mathbf{x}}(V) - \tilde{q}_{\mathbf{x}}(V)| \\
&\quad \leq |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| + |q_{\mathbf{x}}(V) - \tilde{q}_{\mathbf{x}}(V)| && (5.40)\\
&\quad \leq |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| + \frac{q_{\mathbf{x}}(V)}{\text{poly}\, N} && (5.41)\\
&\quad \leq |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| + \frac{|p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| + p_{\mathbf{x}}(V)}{\text{poly}\, N} && (5.42)\\
&\quad = \frac{p_{\mathbf{x}}(V)}{\text{poly}\, N} + |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| \left(1 + \frac{1}{\text{poly}\, N}\right) && (5.43)
\end{aligned}$$

Given that the distributions $p(V)$ and $q(V)$ are ϵ-close in the l_1 norm, particular probabilities $p_{\mathbf{x}}(V)$ and $q_{\mathbf{x}}(V)$ must be exponentially close for most $\mathbf{x}$. This statement is made precise using Markov's inequality: for a nonnegative random variable X and $a > 0$,

$$\Pr(X \geq a) \leq \frac{\mathbb{E}X}{a}. \tag{5.44}$$

Setting $X = |p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)|$ and $a = \epsilon/(|\mathcal{H}|\delta)$,

$$\begin{aligned} &\Pr_{\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}\left(|p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)| \geq \frac{\epsilon}{|\mathcal{H}|\delta}\right) \\ &\qquad \leq \frac{\mathbb{E}_{\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}(|p_{\mathbf{x}}(V) - q_{\mathbf{x}}(V)|)|\mathcal{H}|\delta}{\epsilon} \leq \delta. \end{aligned} \tag{5.45}$$

Combining the probability bound with the inequality (5.43), we have that with probability at least $1 - \delta$ over random $\mathbf{x} \sim \mathrm{unif}(\mathcal{H})$ we have

$$|p_{\mathbf{x}}(V) - \tilde{q}_{\mathbf{x}}(V)| < \frac{p_{\mathbf{x}}(V)}{\mathrm{poly}\, N} + \frac{\epsilon}{|\mathcal{H}|\delta}\left(1 + \frac{1}{\mathrm{poly}\, N}\right). \tag{5.46}$$

To turn the above additive upper bound to a multiplicative one, we use the anticoncentration property (Theorem 5.1), which let us replace $1/|\mathcal{H}|$ by an upper bound $p_{\mathbf{x}}(V)/\alpha$ with probability $(1-\alpha)^2/C$. In order to do so, we must consider the joint probability of $(V, \mathbf{x})$. Let A be the event that $p_{\mathbf{x}}(V)$ and $q_{\mathbf{x}}(V)$ for a fixed V are exponential close due to Markov's inequality, and B be the event that the distribution $p(V)$ anticoncentrates. The probability of both "good events" happening is lower bounded by $\Pr(A \cap B) \geq \max\{0, \Pr(A) + \Pr(B) - 1\}$.

That is, if we denote by $\mathcal{A}(V, \mathbf{x})$ as the event that

$$\begin{aligned} &|p_{\mathbf{x}}(V) - \tilde{q}_{\mathbf{x}}(V)| \\ &\qquad < p_{\mathbf{x}}(V)\left[\frac{1}{\mathrm{poly}\, N} + \frac{\epsilon}{\alpha\delta}\left(1 + \frac{1}{\mathrm{poly}\, N}\right)\right], \end{aligned} \tag{5.47}$$

we have that

$$\Pr_{V\sim\nu,\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}[\mathcal{A}(V, \mathbf{x})] > \frac{(1-\alpha)^2}{C} - \delta, \tag{5.48}$$

which can be simplified by using the hiding property described in Lemma 5.8. The property implies that $p_x(V) = p_{\mathbf{x}_0}(V_{\mathbf{x}})$ and $\tilde{q}_{\mathbf{x}}(V) = \tilde{q}_{\mathbf{x}_0}(V_{\mathbf{x}})$ so that

$$\Pr_{V\sim\nu,\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}[\mathcal{A}(V, \mathbf{x})] = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}\left(\Pr_{V\sim\nu}[\mathcal{A}(V_{\mathbf{x}}, \mathbf{x}_0)]\right). \tag{5.49}$$

Moreover from the invariance of the Haar measure it follows that for every $|\mathbf{x}\rangle \in \mathcal{H}$, $V_{\mathbf{x}}$ is distributed in the same way as V. Consequently,

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}\left(\Pr_{V\sim\nu}\left[\mathcal{A}(V_{\mathbf{x}},\mathbf{x}_0)\right]\right) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathrm{unif}(\mathcal{H})}\left(\Pr_{V\sim\nu}\left[\mathcal{A}(V,\mathbf{x}_0)\right]\right) \\ &= \Pr_{V\sim\nu}\left[\mathcal{A}(V,\mathbf{x}_0)\right]. \end{aligned} \tag{5.50}$$

We finally obtain that for every $\mathbf{x}_0$,

$$\Pr_{V\sim\nu}\left\{|p_{\mathbf{x}_o}(V)-\tilde{q}_{\mathbf{x}_o}(V)| < p_{\mathbf{x}_o}(V)\left[\frac{1}{\mathrm{poly}\,N}+\frac{\epsilon}{\alpha\delta}\left(1+\frac{1}{\mathrm{poly}\,N}\right)\right]\right\} > \frac{(1-\alpha)^2}{C}-\delta. \tag{5.51}$$

Following [BJS11] and requiring a constant ϵ and relative error $\epsilon/(\alpha\delta)$ we may set, for instance,

$$\alpha=\frac{1}{2}, \qquad \delta=\frac{(1-\alpha)^2}{2C}=\frac{1}{8C}, \qquad \epsilon=\frac{\alpha\delta}{4}=\frac{1}{64C}. \tag{5.52}$$

Stockmeyer's algorithm is able to output $(1/4+o(1),1/(8C))$-multiplicative approximates of the output probabilities for $1/(8C)$ fraction of the (passive or active, with constant C_{pas} or C_{act} respectively) FLO circuits V if there is a classical machine that approximately sample from $p_{\mathbf{x}}(V)$ for any FLO circuit V within the l_1 distance $1/(64C)$. ■

Having proved Theorem 5.9 we go now to prove the hardness of sampling. We require two more conjectures for this.

Conjecture 1 (Average-case of approximating probabilities on FLO circuits initialized in $|\Psi_{\mathrm{in}}\rangle$). *Computing a $(1/4+o(1),1/(8C))$-multiplicative approximate to $p_{\mathbf{x}_0}(V,\Psi_{\mathrm{in}})$ for $1/(8C)$ fraction of V sampled from the Haar distribution ν is #P-hard. ($C=C_{\mathrm{pas}},\nu=\nu_{\mathrm{pas}}$ for passive FLO circuits and $C=C_{\mathrm{act}},\nu=\nu_{\mathrm{act}}$ for active FLO circuits)*

Conjecture 2. *The Polynomial Hierarchy does not collapse.*

Theorem 5.10 (Hardness of sampling from FLO circuits initialized in $|\Psi_{\mathrm{in}}\rangle$). *If Conjectures 1 and 2 are true, then there is no efficient classical algorithm that can*

approximately sample with l_1-error $1/(64C_{\text{pas}})$ (resp. $1/(64C_{\text{act}})$) from output probability distributions induced by passive (resp. active) FLO circuits with the input given by $|\Psi_{\text{in}}\rangle$.

Proof. By Theorem 5.9, if there were an approximate sampler with respect to passive (resp. active) FLO circuits with input $|\Psi_{\text{in}}\rangle$, then there would exist a algorithm $\mathsf{BPP}^{\mathsf{NP}}$ that $(1/4 + o(1), 1/(8C))$-multiplicative approximates $p_{\mathbf{x}_0}(V, \Psi_{\text{in}})$ in for $1/(8C)$ fraction of passive (resp. active) FLO circuits. Where $C = C_{\text{pas}}$ in the passive case and $C = C_{\text{act}}$ in the active. By Conjecture 1 this is a #P-hard problem. It is known [Lau83] that BPP is inside the third level of the Polynomial Hierarchy, i.e., $\mathsf{BPP}^{\mathsf{NP}} \subseteq \Sigma_3$. By a well known result of Toda [Tod91] $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$ and thus $\mathsf{PH} \subseteq \Sigma_3$. ∎

This established the hardness of sampling which crucially depends on Conjecture 1 and Conjecture 2. In the next chapters we seek to prove a weaker version of Conjecture 1 to lend support to it.

5.6 Cayley path and average-case hardness

In Section 5.5 we have shown the unlikely existence of a classical sampler for the Fermion Sampling scheme assuming certain plausible conjectures about average-case hardness of approximating output probabilities (Conjecture 1) and the non-collapse of the Polynomial Hierarchy (Conjecture 2). While Conjecture 2 is widely believed in computer science, it could be argued that the evidence for Conjecture 1 is weaker. To support this conjecture we follow [BFNV19, Mov19] in showing that it is #P-hard to give approximations to output probabilities up to error $\epsilon = \exp\left(-\Theta(N^7)\right)$ for a significant fraction of random FLO circuits, where N is the number of quadruples $|\Psi_4\rangle$ used in the scheme as defined in Section 5.3. We give a proof of this fact in Section 5.7 through a worst-to-average case reduction. This proof involves the use of the Cayley path technique, a high-level view of how this technique is applied in the proof is given at the beginning of Section 5.7.

In this section we provide the basic definitions to implement the Cayley path technique on $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$ which is used in the proof of the worst-to-average case reduction. First, in Section 5.6.1 we define the Cayley path for the unitary and orthogonal groups. Then, in Section 5.6.2 we show that by sampling from points far enough from an initial point of the Cayley path, the induced distribution is close to Haar random. This fact will be used in Section 5.7 for the average-case reduction. In Section 5.6.3 we show that the Cayley path allows us to define rational polynomials in terms of a deformation parameter. Finally, in Section 5.6.4 we compute the degree of this polynomials for passive FLO and active FLO. The degree of these polynomials has en effect in the error bounds obtained for the hardness result.

5.6.1 The Cayley path

To prove the reduction to the average-case, we use the technique introduced in [Mov19] which uses a rational interpolation of the output probabilities generated by a quantum circuit. We generalize this technique to be applicable to our case, which requires a rational interpolation between elements of the low-dimensional representations instead of the circuits themselves. To be more precise, the Cayley path will be applied on the groups $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$ for which passive FLO and active FLO are representations respectively. As mentioned in Section 5.2, we use G to denote either the Lie groups $\mathrm{U}(d)$ or $\mathrm{SO}(d)$. To perform the interpolation, we will define the Cayley map which maps elements from the Lie algebra $\mathfrak{g}$ to the corresponding Lie group G. The Lie algebras $\mathfrak{u}(d)$ and $\mathfrak{so}(2d)$ of $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$ are defined to be

$$\mathfrak{u}(d) = \{X \in \mathbb{C}^{d\times d} | X^\dagger = -X\}, \tag{5.53}$$

$$\mathfrak{so}(2d) = \{X \in \mathbb{R}^{2d\times 2d} | X^T = -X\}, \tag{5.54}$$

where X^T denotes the transpose of the matrix X.

Any element $X \in \mathfrak{g}$ defines a group element $\exp(\theta X)$ for $\theta \in \mathbb{R}$. As remarked in [BFNV19, Mov19], such an exponential map does not allow for polynomial interpolation

techniques, in [BFNV19] the authors approximate the exponential map by using the Taylor expansion and introducing a cutoff. In [Mov19], instead, the author uses the Cayley transform to map from Hamiltonians to circuits. We adapt the Cayley transform to apply it in our sampling scheme, we begin by introducing this mapping.

Definition 5.11. Let G be $\mathrm{U}(d)$ or $\mathrm{SO}(2d)$, and let $\mathfrak{g}$ denotes its Lie algebra. The Cayley transform is a mapping $f : \mathfrak{g} \to G$ defined via

$$f(X) = (\mathbb{1} - X)(\mathbb{1} + X)^{-1} . \tag{5.55}$$

Note that the image $f(\mathfrak{g})$ corresponds to the set of group elements $\tilde{G}$ that don't have -1 as an eigenvalue. This can be seen from the definition of the Cayley map in [Mov19] where $f(-\infty) = -1$ and the fact that any X has finite eigenvalues. We can define the inverse map on $\tilde{G}$ $f^{-1} : \tilde{G} \to \mathfrak{g}$ as

$$f^{-1}(g) = (\mathbb{1} - g)(\mathbb{1} + g)^{-1} , \tag{5.56}$$

where $g \in \tilde{G}$. That this is the inverse map can be verified from a simple calculation

$$\begin{aligned} f^{-1}(f(X)) &= f^{-1}((\mathbb{1} - X)(\mathbb{1} + X)^{-1}) \\ &= \left((\mathbb{1} - (\mathbb{1} - X)(\mathbb{1} + X)^{-1})\right) \left(\mathbb{1} + (\mathbb{1} - X)(\mathbb{1} + X)^{-1}\right)^{-1} \\ &= \left([(\mathbb{1} + X) - (\mathbb{1} - X)](\mathbb{1} + X)^{-1}\right) \left([(\mathbb{1} + X) + (\mathbb{1} - X)](\mathbb{1} + X)^{-1}\right)^{-1} \\ &= X. \end{aligned} \tag{5.57}$$

The Cayley map defines a continuous path between $g_0 \in G$ and $g_0 f(X)$ as we explain in the next paragraphs (see Fig. 5.2). To define a path on the group, define first the map $F_\theta : \tilde{G} \to G$, given by

$$F_\theta(g) = f(\theta f^{-1}(g)), \ \theta \in [0, 1] . \tag{5.58}$$

This map can be explicitly computed

$$\begin{aligned} F_\theta(g) &= f(\theta f^{-1}(g)) \\ &= f(\theta(\mathbb{1} - g)(\mathbb{1} + g)^{-1}) \end{aligned}$$

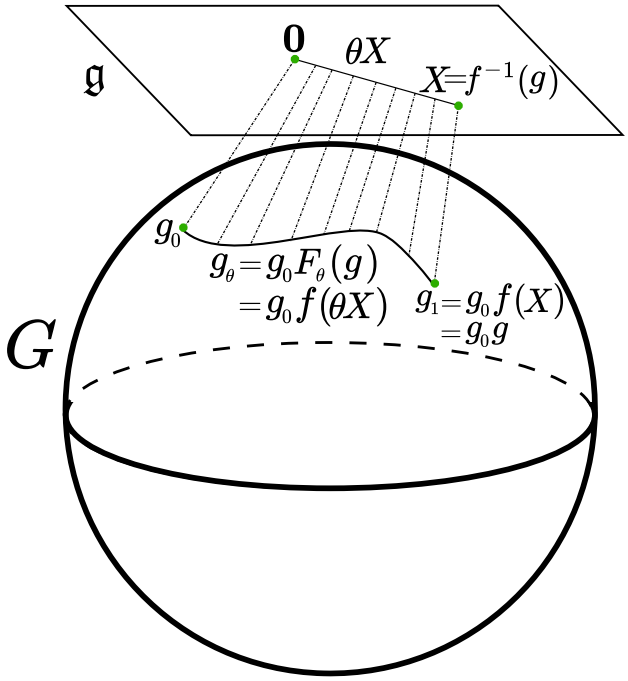

Figure 5.2: Path deformation defined by the Cayley map in Eq. (5.55). A path is induced between element $g_0 \in G$ and $g_0 g$ by taking $X = f^{-1}(g) \in \mathfrak{g}$ and considering the perturbation $g_\theta = g_0 f(\theta X)$.

$$
\begin{aligned}
&= (1-\theta(\mathbb{1}-g)(\mathbb{1}+g)^{-1})(\mathbb{1}+\theta(\mathbb{1}-g)(\mathbb{1}+g))^{-1} \\
&= \left[(\mathbb{1}+g)-\theta(\mathbb{1}-g)\right](\mathbb{1}+g)^{-1}\left[((\mathbb{1}+g)+\theta(\mathbb{1}-g))(\mathbb{1}+g)^{-1}\right]^{-1} \\
&= \left[(\mathbb{1}+g)-\theta(\mathbb{1}-g)\right]\left[(\mathbb{1}+g)+\theta(\mathbb{1}-g)\right]^{-1} \\
&= \left[(1-\theta)\mathbb{1}+(1+\theta)g\right]\left[(1+\theta)\mathbb{1}+(1-\theta)g\right]^{-1}. \qquad (5.59)
\end{aligned}
$$

Note that the elements in the Lie groups considered are represented as normal matrices and thus are diagonalizable, therefore functional calculus can be performed as if real functions were under consideration. We can then write the map F_θ in the following form

$$F_\theta(g) = \frac{(1-\theta)\mathbb{1}+(1+\theta)g}{(1+\theta)\mathbb{1}+(1-\theta)g}\,,\ \theta \in [0,1]\,. \qquad (5.60)$$

For both orthogonal and unitary operators we have $\|g\| = 1$. Therefore for $\theta \in (0,1]$ the denominator of (5.60) does not vanish, while for any $g \in G$ we get that $\lim_{\theta\to 0} F_\theta(g) = \mathbb{1}$ which can be easily seen from the original definition of F_θ. This implies that for $\theta \in [0,1]$ the denominator of (5.60) does not vanish and we can use (5.60) to define F_θ to a be a

function defined on whole G. With the definition of F_θ in place we can now define the following path

$$g_\theta = g_0 F_\theta(g),\ \theta \in [0,1]\ . \tag{5.61}$$

Note that the function F_θ is a rational function of g, which will allow us to use the interpolation techniques from [Mov19]. For the average-case hardness result, we pick g_0 as a group element which corresponds to a worst-case #P-hard FLO circuit while g will be a generic element of the group (i.e. Haar random).

5.6.2 Sampling from the path

In what follows we give a result which says that by sampling elements in the deformation path which are close enough to the final point $g_0 F_\theta(g)$ then the induced distribution is close to sampling from the Haar random distribution.

Lemma 5.12 (Total variation distance between the Haar measure in G and its θ-deformation). *Let G be equal to $\mathrm{U}(d)$ or $SO(2d)$. Let $g_0 \in G$ be a fixed element in G. Let $g \sim \mu_G$ an let $g_\theta = g_0 F_\theta(g)$, for $\theta \in [0,1]$ and $F_\theta : G \to G$ defined in (5.60). Let now μ_G^θ denotes the induced measure according to which g_θ is distributed. Assume furthermore that $\theta \in [1-\Delta, 1]$, for $\Delta > 0$. We then have*

$$\begin{aligned} \left\| \mu_{\mathrm{U}(d)} - \mu^\theta_{\mathrm{U}(d)} \right\|_{\mathrm{TVD}} &\leq d^2\Delta/2, \\ \left\| \mu_{\mathrm{SO}(2d)} - \mu^\theta_{\mathrm{SO}(2d)} \right\|_{\mathrm{TVD}} &\leq d^2\Delta/2\ , \end{aligned} \tag{5.62}$$

where $\|\cdot\|_{\mathrm{TVD}}$ is the total variation distance for probability distributions.

The proof of this lemma is given in appendix C of [ODMZ22]. To prove it, the total variation distance between the probability distributions is expressed in terms of the distance of the angles defining the unitaries and orthogonal rotation in the passive and active case. Using standard techniques from multivariable calculus the distance can be bounded.

5.6.3 Rational polynomials from Cayley path

In [Mov19], the degree of the interpolation is related to the robustness of the average-case hardness. This will become explicit in Section 5.7. In this section we write explicitly the rational functions which will give us the degree.

Given some element $g \in \mathrm{U}(d)$, as we mention in Section 5.2.3, we can always diagonalize with some operator $h \in \mathrm{U}(d)$ such that $hgh^{-1} = \sum_{j=1}^{d} e^{i\phi_j} |j\rangle\langle j|$. Then we have that,

$$g_\theta = g_0 F_\theta(g) \tag{5.63}$$

$$= g_0 \left[(1-\theta)\mathbb{1} + (1+\theta)g\right] \left[(1+\theta)\mathbb{1} + (1-\theta)g\right]^{-1} \tag{5.64}$$

$$= g_0 \left[\sum_{j=1}^{d} ((1-\theta) + (1+\theta)e^{i\phi_j})h^{-1} |j\rangle\langle j| h\right] \left[\sum_{j=1}^{d} ((1+\theta) + (1-\theta)e^{i\phi_j})h^{-1} |j\rangle\langle j| h\right]^{-1} \tag{5.65}$$

$$= \sum_{j=1}^{d} \frac{(1-\theta) + (1+\theta)e^{i\phi_j}}{(1+\theta) + (1-\theta)e^{i\phi_j}} g_0 h^{-1} |j\rangle\langle j| h \tag{5.66}$$

$$= \sum_{j=1}^{d} \frac{1 + i\theta \tan(\phi_j/2)}{1 - i\theta \tan(\phi_j/2)} g_0 h^{-1} |j\rangle\langle j| h \tag{5.67}$$

$$= \frac{1}{Q_g(\theta)} \sum_{j=1}^{d} P_j(\theta) g_0 h^{-1} |j\rangle\langle j| h =: \frac{\mathcal{P}_{g_0,g}(\theta)}{Q_g(\theta)}, \tag{5.68}$$

where

$$Q_g(\theta) = \prod_{j=1}^{d} (1 - i\theta \tan(\phi_j/2)), \tag{5.69}$$

$$P_j(\theta) = (1 + i\theta \tan(\phi_j/2)) \prod_{\substack{1 \le k \le d \\ k \neq j}} (1 - i\theta \tan(\phi_k/2)) \tag{5.70}$$

are both polynomials of degree d in θ, and $\mathcal{P}_{g_0,g}(\theta)$ is a formal polynomial that depends on the matrices g and g_0. In the $\mathrm{SO}(2d)$ case we use the fact that we can block diagonalize a group element as discussed in Section 5.2.3. We define $\mathbb{I}_j = |2j-1\rangle\langle 2j-1| + |2j\rangle\langle 2j|$ and $\tilde{X} = |2j\rangle\langle 2j-1| - |2j-1\rangle\langle 2j|$. Again, let $hgh^{-1} = \sum_{j=1}^{d} (\cos\phi_j \mathbb{1}_j + \sin\phi_j \tilde{X}_j)$, and one has

$$
\begin{aligned}
g_\theta &= g_0 \sum_{j=1}^{d} [1+\cos\phi_j + \theta^2(1-\cos\phi_j)]^{-1} \\
&\quad \times \left\{ \left[1+\cos\phi_j - \theta^2(1-\cos\phi_j)\right] \mathbb{1}_j + 2\theta \sin\phi_j h^{-1}\tilde{X}_j h \right\} && (5.71)\\
&= g_0 \sum_{j=1}^{d} \frac{[1-\theta^2\tan^2(\phi_j/2)]\mathbb{1}_j + 2\theta\tan(\phi/2)h^{-1}\tilde{X}_j h}{1+\theta^2\tan^2(\phi_j/2)} && (5.72)\\
&= \frac{1}{Q_g(\theta)} \sum_{j=1}^{d} \left(P_j^{\mathrm{diag}}(\theta) g_0 \mathbb{1}_j + P_j^{\mathrm{off}}(\theta) g_0 h^{-1}\tilde{X}_j h \right) && (5.73)\\
&=: \frac{\mathcal{P}_{g_0,g}(\theta)}{Q_g(\theta)}\,, && (5.74)
\end{aligned}
$$

where in (5.72) we divided both the numerator and the denominator by $1+\cos\phi_j$ and

$$
\begin{aligned}
Q_g(\theta) &= \prod_{j=1}^{d} (1+\theta^2\tan^2(\phi_j/2))\,, && (5.75)\\
P_j^{\mathrm{diag}}(\theta) &= (1+\theta^2\tan^2(\phi_j/2))^2) \\
&\quad \times \prod_{\substack{1\le k\le d \\ k\neq j}} (1+\theta^2\tan^2(\phi_j/2))^2)\,, && (5.76)\\
P_j^{\mathrm{off}}(\theta) &= 2\theta\tan\left(\phi_j/2\right) \prod_{\substack{1\le k\le d \\ k\neq j}} (1+\theta^2\tan^2(\phi_j/2))^2) && (5.77)
\end{aligned}
$$

are polynomials in θ of degree $2d$, $2d$, and $2d-1$ respectively, and $\mathcal{P}_{g_0,g}(\theta)$ is a formal polynomial that depends on the matrices g and g_0.

Following previous work [BFNV19, Mov19], we give a lower bound for $Q_g(\theta)$ to assure that the rational function does not blow up, and an upper bound for generic $g \in G$, which will be crucial for a robust reduction in Section 5.7. Note that the coefficients of the polynomial $Q_g(\theta)$ depend only on generalized eigenvalues of g ($e^{i\phi_j}$ in the unitary case and $\cos\phi_j$, $\sin\phi_j$ in the orthogonal case) and hence $Q(\theta)$ can be pre-computed in time polynomial in d by diagonalizing g, computing each $\tan\left(\phi_j/2\right)$ which is just an algebraic function of $e^{i\phi_j}$, and computing the final result.

Lemma 5.13. *Let $Q_g(\theta)$ be the polynomial in defined in (5.69) for $G = \mathrm{U}(d)$ and in (5.75) for $G = \mathrm{SO}(2d)$. Let now $\tilde{\Delta} > 0$. Then we have the following inequalities*

$$\Pr_{g\sim\mu_{\mathrm{U}(d)}}\left(\left|Q_g(\theta)\right|^2 \leq \left[1+\left(\frac{\theta\pi}{\tilde{\Delta}}\right)^2\right]^d\right) \geq 1 - d\frac{\tilde{\Delta}}{\pi}\,, \tag{5.78}$$

$$\Pr_{g\sim\mu_{\mathrm{SO}(2d)}}\left(\left|Q_g(\theta)\right|^2 \leq \left[1+\left(\frac{\theta\pi}{\tilde{\Delta}}\right)^2\right]^{2d}\right) \geq 1 - d\frac{\tilde{\Delta}}{\pi}\,. \tag{5.79}$$

In addition, for all g, $|Q_g(\theta)|^2 \geq 1$ for both $U(d)$ and $\mathrm{SO}(2d)$.

Proof. Since $g \in G$ is Haar distributed, every generalised eigenphase ϕ_j is distributed uniformly on the interval $[-\pi, \pi]$ [AS17]. Therefore, for every j we have

$$\Pr_{g\sim\mu_G}\left(\phi_j \in [\pi - \tilde{\Delta}, \pi] \cup [-\pi, -\pi + \tilde{\Delta}]\right) = \frac{2\tilde{\Delta}}{2\pi} = \frac{\tilde{\Delta}}{\pi}. \tag{5.80}$$

For $\phi_j \in [-\pi + \tilde{\Delta}, \pi - \tilde{\Delta}]$ we have $\left|\tan(\phi_j/2)\right| \leq \pi/\tilde{\Delta}$. Call A_j the event that $\phi_j \in [\pi - \tilde{\Delta}, \pi] \cup [-\pi, -\pi + \tilde{\Delta}]$. Then

$$\Pr_{g\sim\mu_G}\left(\bigcup_{i=1}^{d} A_j\right) \leq \sum_{i=1}^{d} \Pr_{g\sim\mu_G}(A_j) \tag{5.81}$$

$$= d\frac{\tilde{\Delta}}{\pi}. \tag{5.82}$$

From this bound we obtain

$$\Pr_{g\sim\mu_G}\left(\forall j \in [d],\ \left|\tan(\phi_j/2)\right| \leq \pi/\tilde{\Delta}\right) = 1 - \Pr_{g\sim\mu_G}\left(\bigcup_{i=1}^{d} A_j\right) \tag{5.83}$$

$$\geq 1 - d\frac{\tilde{\Delta}}{\pi}. \tag{5.84}$$

First, let us prove Eq. (5.78). We have that with probability $\geq 1 - d\frac{\tilde{\Delta}}{\pi}$

$$\left|Q_g(\theta)\right|^2 = \prod_{j=1}^{d}\left|1 - i\theta\tan(\phi_j/2)\right|^2 \tag{5.85}$$

$$= \prod_{j=1}^{d}(1 - i\theta\tan(\phi_j/2))(1 + i\theta\tan(\phi_j/2)) \tag{5.86}$$

$$= \prod_{j=1}^{d} (1 + \theta^2 \tan^2(\phi_j/2)) \tag{5.87}$$

$$\leq \prod_{j=1}^{d} (1 + (\frac{\theta\pi}{\tilde{\Delta}})^2) \tag{5.88}$$

$$= (1 + (\frac{\theta\pi}{\tilde{\Delta}})^2)^d \tag{5.89}$$

where the probability is introduced when using Eq. (5.83). The SO($2d$) case follows but the same reasoning but replacing d with $2d$. ■

5.6.4 Polynomials associated to probabilities in FLO circuits

Having obtained the rational polynomials from the Cayley path in Section 5.6.3, we show the explicit rational polynomials of the output probabilities for FLO in our scheme. In this section, we give the degrees of matrix polynomials associated to fermionic representations of $G = \mathrm{U}(d)$ and $G = \mathrm{SO}(2d)$. These polynomials, when evaluated on the Cayley path g_θ in the appropriate group (see Eq. (5.60)), give rise to polynomials and rational functions θ for the outcome probabilities $p_{\mathbf{x}}(\Pi(g_\theta), \Psi_{\mathrm{in}}) = |\langle \mathbf{x}| \Pi(g) |\Psi_{\mathrm{in}}\rangle|^2$ in our quantum advantage schemes. The explicit form of these polynomials will be used in Section 5.7 when discussing worst-to-average-case reductions.

We introduce some notation that will be helpful in what follows. Given a $d \times d$ matrix M and two subsets of indices $\mathcal{X}, \mathcal{Y} \subset [d]$ with cardinality n, where $\mathcal{X} = \{a_1, a_2, \ldots, a_n\}$ ($a_i < a_j$ if $i < j$) and $\mathcal{Y} = \{b_1, b_2, \ldots, b_n\}$ ($b_i < b_j$ if $i < j$), we define $M_{\mathcal{X},\mathcal{Y}}$ as the $n \times n$ matrix with entries

$$(M_{\mathcal{X},\mathcal{Y}})_{k,\ell} = M_{a_k,b_\ell}, \quad k, \ell = 1, \ldots, n. \tag{5.90}$$

In other words, the sets $\mathcal{X}$ and $\mathcal{Y}$ correspond to indexes of a submatrix of M. This will be relevant to FLO as we will only care about submatrices of the unitaries where the indexes are given by the input state. When working with passive FLO, it is well known that when the input is a Fock state and the output is measured in the particle number basis, then the output probabilities are described by determinants.

Lemma 5.14. *Given two Fock basis states* $|\mathcal{X}\rangle, |\mathcal{Y}\rangle \in \bigwedge^n(\mathbb{C}^d)$ *and a* $U \in \mathrm{U}(d)$*, the amplitude between* $|\mathcal{X}\rangle$ *and* $\Pi_{\mathrm{pas}}(U)\,|\mathcal{Y}\rangle$ *is provided by the expression*

$$\langle\mathcal{X}|\,\Pi_{\mathrm{pas}}(U)\,|\mathcal{Y}\rangle = \det\big(U_{\mathcal{X},\mathcal{Y}}\big). \tag{5.91}$$

Proof. Let $\mathcal{X} = \{a_1, a_2, \ldots, a_n\}$ ($a_i < a_j$ if $i < j$) and $\mathcal{Y} = \{b_1, b_2, \ldots, b_n\}$ ($b_i < b_j$ if $i < j$). By definition we have that

$$\begin{aligned}\Pi_{\mathrm{pas}}(U)\,|\mathcal{Y}\rangle &= U^{\otimes n}\,|b_1\rangle \wedge |b_2\rangle \wedge \cdots \wedge |b_n\rangle \\ &= |\xi_1\rangle \wedge |\xi_2\rangle \wedge \cdots \wedge |\xi_n\rangle\end{aligned} \tag{5.92}$$

where

$$|\xi_\ell\rangle = U\,|b_\ell\rangle = \sum_{j=1}^{d} U_{j,b_\ell}\,|j\rangle\ , \quad \ell = 1, \ldots, n. \tag{5.93}$$

Using the last two equations and Eq. (5.6), we can deduce that

$$\begin{aligned}\langle\mathcal{X}|\,\Pi_{\mathrm{pas}}(U)\,|\mathcal{Y}\rangle &= \det(C)\,, \\ C_{k,\ell} = \langle a_k|\xi_\ell\rangle = \langle a_k|\sum_{j=1}^{d} U_{j,b_\ell}\,|j\rangle &= U_{a_k,b_\ell} = (U_{\mathcal{X},\mathcal{Y}})_{k,\ell}\end{aligned} \tag{5.94}$$

which proves the statement. ∎

This result implies that just using Fock states as inputs in our scheme is not enough for obtaining output probabilities that are #P-hard. In the proof we have explicitly used the "first quantization" picture, but alternatively we could have used the second quantization. For completeness we sketch such procedure here.

Let $|x\rangle = f^\dagger_{a_1} f^\dagger_{a_2} \cdots f^\dagger_{a_n}\,|00\cdots0\rangle$ and $|\mathbf{y}\rangle = f^\dagger_{b_1} f^\dagger_{b_2} \cdots f^\dagger_{b_n}\,|00\cdots0\rangle$, such that $|x\rangle = |\mathcal{X}\rangle$ and $|y\rangle = |\mathcal{Y}\rangle$ as defined above. Now, consider the action of U on $|y\rangle$

$$U\,|y\rangle = U f^\dagger_{b_1} f^\dagger_{b_2} \cdots f^\dagger_{b_n}\,|00\cdots0\rangle \tag{5.95}$$

$$= (U f^\dagger_{b_1} U^\dagger) \cdots (U f^\dagger_{b_n} U^\dagger)\,|00\cdots0\rangle \tag{5.96}$$

$$= \sum_{p_1,\cdots,p_n=1}^{d} V_{b_1 p_1} \cdots V_{b_n p_n} f^\dagger_{p_1} \cdots f^\dagger_{p_n}\,|00\cdots0\rangle \tag{5.97}$$

We then have that

$$\langle x|U|y\rangle = \sum_{p_1,\cdots,p_n=1}^{d} V_{b_1 p_1}\cdots V_{b_n p_n} f_{a_n}\cdots f_{a_1} f_{p_1}^{\dagger}\cdots f_{p_n}^{\dagger}\,|0\rangle \tag{5.98}$$

$$= \sum_{\pi} sgn(\pi) V_{b_1\pi(a_1)}\cdots V_{b_n\pi(a_n)} \tag{5.99}$$

By considering $\tilde{V}$ as the submatrix of V with the selected rows $b_1,\cdots,b_n$ and columns $a_1,\cdots,a_n$ we have $\langle x|U|y\rangle = \det\left(\tilde{V}\right)$. By a direct translation from first quantization to second quantization we can see that both results are equivalent.

In the following result, we give the form of the polynomial for general inputs.

Proposition 5.15 (Degrees of polynomials describing probabilities associated to passive FLO circuits.). *Consider a state $|\Psi\rangle \in \bigwedge^n(\mathbb{C}^d)$. For an arbitrary $U \in \mathrm{U}(d)$ the outcome probability $p_{\mathbf{x}}(\Pi_{\mathrm{pas}}(U),\Psi) = |\langle \mathbf{x}|\,\Pi_{\mathrm{pas}}(U)\,|\Psi\rangle|^2$ is a degree $2n$ homogeneous polynomial in the entries of U and $U^{\dagger}$.*

Proof. One can expand the vector $|\Psi\rangle$ in terms of the Fock basis states belonging to $\bigwedge^n(\mathbb{C}^d)$ as

$$|\Psi\rangle = \sum_{\substack{\mathcal{Y}\subset[d]\\|\mathcal{Y}|=n}} c_{\mathcal{Y}}\,|\mathcal{Y}\rangle\,. \tag{5.100}$$

Let $\mathcal{X}\subset[d]$ denote the set of indices corresponding to $\mathbf{x}$ as an indicator function (i.e., $|\mathbf{x}\rangle = |\mathcal{X}\rangle$). Using Lemma 5.14, we can write the relevant amplitude as

$$\begin{aligned}\langle \mathbf{x}|\,\Pi_{\mathrm{pas}}(U)\,|\Psi\rangle &= \sum_{\substack{\mathcal{Y}\subset[d]\\|\mathcal{Y}|=n}} c_{\mathcal{Y}}\,\langle \mathcal{X}|\,\Pi_{\mathrm{pas}}(U)\,|\mathcal{Y}\rangle \\ &= \sum_{\substack{\mathcal{Y}\subset[d]\\|\mathcal{Y}|=n}} c_{\mathcal{Y}}\det\left(U_{\mathcal{X},\mathcal{Y}}\right).\end{aligned} \tag{5.101}$$

As each term in the sum is a determinant of a $n\times n$ submatrix of U, this expression gives a homogeneous polynomial of the entries of U of order n. This in turn directly implies that $p_{\mathbf{x}}(\Pi_{\mathrm{pas}}(U),\Psi) = |\langle \mathbf{x}|\,\Pi_{\mathrm{pas}}(U)\,|\Psi\rangle|^2$ is a degree $2n$ polynomial in the entries of U and $U^{\dagger}$. ∎

Lemma 5.16 (Polynomial for output amplitude of passive FLO [Iva17]). *Consider the input state* $|\Psi_{\text{in}}\rangle = |\Psi_4\rangle^{\otimes N} \in \bigwedge^{2N}(\mathbb{C}^{4N})$. *For an arbitrary* $U \in \mathrm{U}(4N)$ *the outcome amplitude is given by*

$$\begin{aligned}\langle \mathbf{x}|\Pi_{\text{pas}}(U)\,|\Psi\rangle = \frac{1}{\sqrt{2^N}} \sum_{(y_1,\cdots,y_N)\in\{0,1\}^N} \\ \times \det\left(U^T_{\{2y_1+1,2y_1+2,\cdots,2y_N+4N-3,2y_N+4N-2\},\mathcal{X}}\right),\end{aligned} \tag{5.102}$$

where $U^T_{\{2y_1+1,2y_1+2,\cdots,2y_N+4N-3,2y_N+4N-2\},\mathcal{X}}$ *indicates the transpose of* U *with the rows not indexed by* $\{2y_1+1, 2y_1+2, \cdots, 2y_N+4N-3, 2y_N+4N-2\}$ *and columns not indexed by* $\mathcal{X}$. *Note that this is a degree* N *polynomial in the entries of* U.

Proof. To derive this polynomial, we introduce the convenient notation for $|\Psi_{\text{in}}\rangle$:

$$|\Psi_{\text{in}}\rangle = \frac{1}{\sqrt{2^N}} \sum_{\mathcal{X}\in C_{\text{in}}} |\mathcal{Y}\rangle\ , \tag{5.103}$$

where C_{in} is a collection of subsets of $[4N]$ that appear in the decomposition of $|\Psi_{\text{in}}\rangle$. Note that from the definition of $|\Psi_{\text{in}}\rangle$ it follows that subsets are labelled by bitstrings $\mathbf{x} = (y_1, \ldots, y_N)$, where $y_i \in \{0, 1\}$ labels which pair of the neighbouring physical modes are occupied in a given quadropule of modes. For $N = 2$ we have four possible subsets belonging to C_{in}

$$\mathcal{Y}_{00} = \{1, 2, 5, 6\}\ ,\ \mathcal{Y}_{01} = \{1, 2, 7, 8\}\ ,\ \mathcal{Y}_{10} = \{3, 4, 5, 6\},\ \mathcal{Y}_{11} = \{3, 4, 7, 8\}\ . \tag{5.104}$$

For general N the collection C_{in} consists of the following subsets labelled by bitstrings $\mathbf{x}$

$$\mathcal{Y}_{\mathrm{y}} = \{1{+}2y_1, 2{+}2y_1, 5{+}2y_2, 6{+}2y_2, \ldots, 4i{-}3{+}2y_i, 4i{-}2{+}2y_i, \ldots, 4N{-}3{+}2y_N, 4N{-}2{+}2y_N\}\ . \tag{5.105}$$

We then write,

$$|\Psi_{\text{in}}\rangle = \frac{1}{\sqrt{2^N}} \sum_{\mathcal{Y}\in C_{\text{in}}} |\mathcal{Y}\rangle \tag{5.106}$$

$$= \frac{1}{\sqrt{2^N}} \sum_{\mathrm{y}\in\{0,1\}^N} \left|\mathcal{Y}_{\mathrm{y}}\right\rangle, \tag{5.107}$$

where C_{in} consists of subsets labelled by bitstrings. Using the expression for the output amplitude in Proposition 5.15 we write

$$\langle \mathbf{x}|\Pi_{\text{pas}}(U)\,|\Psi\rangle = \frac{1}{\sqrt{2^N}} \sum_{\mathbf{y}\in\{0,1\}^N} \det\Big(U_{\mathcal{X},\mathcal{Y}_{\mathbf{y}}}\Big) \tag{5.108}$$

$$= \frac{1}{\sqrt{2^N}} \sum_{(y_1,\cdots,y_N)\in\{0,1\}^N} \times \det\Big(U^T_{\{2y_1+1,2y_1+2,\cdots,2y_N+4N-3,2y_N+4N-2\},\mathcal{X}}\Big)\,, \tag{5.109}$$

where in the last line we have replaced the definition of $\mathcal{Y}_y$ and also used the fact that the determinant is invariant under the transpose. ■

For further clarity, we sketch the same result in second quantization. The input of the scheme is given by [Iva17]

$$|\Psi_{\text{in}}\rangle = \frac{1}{\sqrt{2}^N} \sum_{i_1,\cdots,i_k=0}^{1} f^\dagger_{2i_1+1} f^\dagger_{2i_1+2} f^\dagger_{2i_2+5} f^\dagger_{2i_2+6} \cdots f^\dagger_{2i_N+4N-3} f^\dagger_{2i_N+4N-2}\,|0\cdots 0\rangle\,. \tag{5.110}$$

Then, we have

$$\langle 0|f_{q_{2N}}\cdots f_{q_1} U|\Psi_{\text{in}}\rangle = \frac{1}{\sqrt{2}^N} \sum_{i_1,\cdots,i_N=0}^{1} \sum_{\pi\in S_{2N}} sgn(\pi) V_{2i_1+1,\pi(q_1)} V_{2i_1+2,\pi(q_2)} \cdots V_{2i_N+4N-2,\pi(q_2N)} \tag{5.111}$$

$$= \frac{1}{\sqrt{2}^N} \sum_{i_1,\cdots,i_N=0}^{1} \det\big(V_{\{2i_1+1,2_i1+2,\cdots,2i_N+4N-2\},\{q_1,\cdots,q_{2N}\}}\big) \tag{5.112}$$

where S_{2N} is the set of permutations over $2N$ elements. The expression in Eq. (5.102) can be rewritten as

$$D_{2,2}(v_1,\ldots,v_{4N}) = \frac{1}{\sqrt{2^N}} \sum_{\substack{i_k=0,1,\\ k=1,..,N}} \det \begin{bmatrix} v_{2i_1+1} \\ v_{2i_1+2} \\ v_{2i_2+5} \\ v_{2i_2+6} \\ \vdots \\ v_{2i_N+4N-3} \\ v_{2i_N+4N-2} \end{bmatrix}, \tag{5.113}$$

here v_k correspond to the rows of the matrix U^T in Eq. (5.109) with the columns not indexed by $\mathbf{x}$ removed. This polynomial over entries of matrices of size $2N \times N$ was found to be #P-hard to compute in the general case [Iva17]. This was proven by reducing the computation of the permanent of a weighted adjacency matrix to these polynomials of a transformed adjacency matrix with polynomial overhead.

Remark 5.17. For the hardness of sampling what is actually required is the #P-hardness of computing the square of the amplitude. In [Iva17] the permanents used only involved positive numbers and thus there is no issue in establishing #P-hardness for the probabilities.

Next we turn to studying the output probabilities after an active FLO evolution. It will be useful to introduce the following notation: given a set of (majorana) indices $\mathcal{A} = \{a_1, a_2, \dots a_k\} \subset [2d]$ (with $a_i < a_j$ if $i < j$), we define

$$m_{\mathcal{A}} = m_{a_1} m_{a_2} \cdots m_{a_k}. \tag{5.114}$$

These majorana monomials define an orthogonal (but not orthonormal) basis in the space of operators with respect to the Hilbert-Schmidt scalar product

$$\mathrm{tr}\left(m_{\mathcal{A}} m_{\mathcal{B}}^{\dagger}\right) = (-1)^{f(|\mathcal{B}|)} \mathrm{tr}(m_{\mathcal{A}} m_{\mathcal{B}}) = \delta_{\mathcal{A},\mathcal{B}} \frac{1}{2^d}, \tag{5.115}$$

where $f(n) = 1$ if $(n \mod 4) \in \{2, 3\}$ and $f(n) = 0$ otherwise.

Consider a subset $\mathcal{A} = a_1, a_2 \dots a_k \subset [2d]$ (with $a_i < a_j$ if $i < j$), then from Eq. (5.15) and the majorana anticommutation relations it follows for $O \in \mathrm{SO}(2d)$ that

$$\begin{aligned} &\Pi_{\mathrm{act}}(O) m_{\mathcal{A}} \Pi_{\mathrm{act}}(O)^{\dagger} \\ &= \sum_{b_1,\dots b_k=1}^{d} \epsilon_{b_1,b_2,\dots,b_k} O_{a_1,b_1} O_{a_2,b_2} \cdots O_{a_k,b_k} m_{\{b_1,\dots b_k\}}. \end{aligned} \tag{5.116}$$

Proposition 5.18 (Degrees of polynomials describing probabilities associated to active FLO circuits.). *Consider a state* $|\Psi\rangle \in$. *For an arbitrary* $O \in \mathrm{SO}(2d)$ *the outcome probability* $p_{\mathbf{x}}(\Pi_{\mathrm{act}}(O), \Psi) = |\langle \mathbf{x}| \Pi_{\mathrm{act}}(O) |\Psi\rangle|^2$ *is a degree* d *polynomial in the entries of* O.

Proof. Let us consider the expansion of $|\mathbf{x}\rangle\langle\mathbf{x}|$ and Ψ in terms of majorana monomials

$$\begin{aligned} |\mathbf{x}\rangle\langle\mathbf{x}| &= \sum_{\mathcal{A}\subset[d]} \left(a_{\mathcal{A}}\, m_{\mathcal{A}} + b_{\mathcal{A}}\, Q\, m_{\mathcal{A}}\right), \\ \Psi &= \sum_{\mathcal{B}\subset[d]} \left(c_{\mathcal{B}}\, m_{\mathcal{B}} + d_{\mathcal{B}}\, Q\, m_{\mathcal{B}}\right). \end{aligned} \tag{5.117}$$

Using this and Eqs. (5.115) and (5.116) can now write the outcome probability as

$$\begin{aligned} p_{\mathbf{x}}(\Pi_{\mathrm{act}}(O), \Psi) &= \mathrm{tr}\left(|\mathbf{x}\rangle\langle\mathbf{x}|\, \Pi_{\mathrm{act}}(O)\Psi\Pi_{\mathrm{act}}(O)^\dagger\right) \\ &= \sum_{\mathcal{A},\mathcal{B}\subset[d]} a_{\mathcal{A}}c_{\mathcal{B}}\, \mathrm{tr}\left(m_{\mathcal{A}}\Pi_{\mathrm{act}}(O)m_{\mathcal{B}}\Pi_{\mathrm{act}}(O)^\dagger\right) \\ &\quad + b_{\mathcal{A}}d_{\mathcal{B}}\, \mathrm{tr}\left(Qm_{\mathcal{A}}\Pi_{\mathrm{act}}(O)Qm_{\mathcal{B}}\Pi_{\mathrm{act}}(O)^\dagger\right) \\ &= \sum_{k=0}^{d} \sum_{\substack{\mathcal{A},\mathcal{B}\subset[d] \\ |\mathcal{A}|=|\mathcal{B}|=k}} w_{\mathcal{A},\mathcal{B}} \sum_{\ell_1,\ldots\ell_k=1}^{d} \epsilon_{\ell_1,\ell_2,\ldots,\ell_k}\delta_{\mathcal{A},\{\ell_1,\ldots,\ell_k\}}\, O_{b_1,\ell_1}O_{b_2,\ell_2}\cdots O_{b_k,\ell_k}, \end{aligned} \tag{5.118}$$

where $w_{\mathcal{A},\mathcal{B}} = \frac{(-1)^{f(|A|)}}{2^d}(a_{\mathcal{A}}c_{\mathcal{B}} + (-1)^k b_{\mathcal{A}}d_{\mathcal{B}})$. Since each term in the sum is a degree d or less polynomial in the entries of O the theorem is proved. ∎

Definition 5.19 (Degree of rational functions). Let $P(\theta), Q(\theta)$ be polynomials of degree d_1 and d_2 respectively. Let $R(\theta) = \frac{P(\theta)}{Q(\theta)}$ be the corresponding rational function. Assume that that P and Q do not have non-constant polynomial divisors. Then, we define rational degree of R as the pair $\deg(R) = (d_1, d_2)$

The following results states that FLO circuit representations of elements of the appropriate symmetry group G, when evaluated on Cayley paths, give rise to outcome probabilities that are rational functions of low degree (in number of modes d and number of particles n).

Lemma 5.20 (Degrees of rational functions describing probabilities associated to interpolation of FLO circuits). *Let G be equal to $\mathrm{U}(d)$ or $\mathrm{SO}(2d)$. Let $g_0, g \in G$*

be a fixed elements of the group G. Consider a rational path in the group defined by interpolation via Cayley path

$$g_\theta = g_0 F_\theta(g) \ , \ \theta \in [0,1] \ . \tag{5.119}$$

Let now $\Pi : G \to \mathrm{U}(\mathcal{H})$ *be the appropriate representation of G describing appropriate class of FLO circuits (*$G = \mathrm{U}(d)$, $\Pi = \Pi_{\mathrm{pas}}$, $\mathcal{H} = \bigwedge^n(\mathbb{C}^d)$ *for passive FLO and* $G = \mathrm{SO}(2d)$, $\Pi = \Pi_{\mathrm{act}}$, $\mathcal{H} = \mathcal{H}^+_{\mathrm{Fock}}(\mathbb{C}^d)$ *for active FLO). Let us fix* $|\Psi\rangle \in \mathcal{H}$ *and a Fock state* $|\mathbf{x}\rangle \in \mathcal{H}$*. Then the outcome probability*

$$R_{g_0,g}(\theta) = \mathrm{tr}\Big(|\mathbf{x}\rangle\langle\mathbf{x}|\, \Pi(g_\theta)\rho\Pi(g_\theta)^\dagger\Big) \tag{5.120}$$

viewed as a function of parameter θ *is a rational function of degrees*

$$\begin{aligned} \textit{Passive FLO:} \quad & \deg(R_{g_0,g}) = (2dn, 2dn) \ , \\ \textit{Active FLO:} \quad & \deg(R_{g_0,g}) = (2d^2, 2d^2) \end{aligned} \tag{5.121}$$

Moreover the denominator of the rational functions are given by

$$\begin{aligned} \textit{Passive FLO:} \quad & Q_g(\theta) = \prod_{j=1}^{d} (1 + \theta^2 \tan^2(\phi_j/2))^n \ , \\ \textit{Active FLO:} \quad & Q_g(\theta) = \prod_{j=1}^{d} (1 + \theta^2 \tan^2(\phi_j/2))^d \ , \end{aligned} \tag{5.122}$$

where ϕ_j , $j \in [d]$ *are phases of generalized eigenvalues of matrix g belonging to the suitable group G and thus* $Q_g(\theta)$ *can be efficiently computed (see Section 5.6).*

Proof. We begin by proving the passive FLO case. Recall from Eq. (5.68) that g_θ was expressed as a matrix with entries of degree (d, d) on θ. By virtue of Proposition 5.15, we know that $p_{\mathbf{x}}(\Pi_{\mathrm{pas}}(g_\theta), \Psi) = R_{g_0,g}(\theta)$ is a polynomial of degree $2n$ on the entries of g_θ which immediately implies the degree on θ is $\deg(R_{g_0,g}) = (2dn, 2dn)$. The denominator of the rational functions in g_θ is given by Eq. (5.69), from the expression for the amplitude in Proposition 5.15, we know that the denominator in $R_{g_0,g}$ must be of the form $\left|Q_g(\theta)^n\right|^2$ which gives the result form $\prod_{j=1}^{d}(1 + \theta^2 \tan^2(\phi_j/2))^n$.

For the active case, we obtain from Eq. (5.73) that g_θ is a matrix with entries that are polynomials of degree $(2d, 2d)$. Then by Proposition 5.18, $p_{\mathbf{x}}(\Pi_{\text{act}}(g_\theta), \Psi)$ is of degree d on the entries of g_θ implying $\deg(R_{g_0,g}) = (2d^2, 2d^2)$. The denominator $Q_g(\theta)$ is obtained by noting that the expression in Eq. (5.75) for $Q_g(\theta)$ appears as the denominator in each entry of g_θ and by Proposition 5.18 the degree on this denominator is d, thus proving the result. ■

5.7 Robust average-case hardness of output probabilities

In this part we give strong evidence for the Conjecture 1 used to show classical hardness of sampling from fermionic linear circuits initialized in $|\Psi_{\text{in}}\rangle$ (cf. Theorem 5.10). There we conjectured that it is #P-hard to approximate probabilities $p_{\mathbf{x}}(V, \Psi_{\text{in}}) = |\langle \mathbf{x}| V |\Psi_{\text{in}}\rangle|^2$ of *generic* FLO circuits initialized in $|\Psi_{\text{in}}\rangle$ to relative error. To support the conjecture we prove weaker theorems showing average-case #P hardness of exact computation of $p_{\mathbf{x}}(V, \Psi_{\text{in}})$ (Theorem 5.22) and extend it further to average-case #P-hardness of approximating $p(\mathbf{x}|V, \Psi_{\text{in}})$ up to error $\epsilon = \exp\left(-\Theta(N^7)\right)$ (Theorem 5.27), where N is the number of states $|\Psi_4\rangle$ used.

We will use the interpolation technique by [Mov19] on the polynomials obtained in Section 5.6 by using the Cayley transform. Since passive and active FLO circuits form representations of $\mathrm{U}(d)$ and $\mathrm{SO}(2d)$ the Cayley mapping will apply to both cases and we will obtain low-degree polynomials for the output probabilites. We will need the following result that guarantees that it is possible to recover an unknown rational function $F(\theta)$ from a set of its values at different points, even if some of the evaluation are erroneous.

We will consider an element $g_0 \in G$ represented by a circuit $V_0 = \Pi(g_0)$, such that the output probability $p_{\mathbf{x}_0}(V_0, \Psi_{\text{in}})$ is #P-hard to compute for $\mathbf{x}_0$ an specific output state and input $|\Psi_{\text{in}}\rangle = |\Psi_4\rangle^{\otimes N}$, for $|\Psi_4\rangle = (|0011\rangle + |1100\rangle)/\sqrt{2}$. We will use the Cayley path

interpolation between such g_0 and a generic element g.

$$g_\theta = g_0 F_\theta(g) \ , \ g \sim \mu_G \ . \tag{5.123}$$

Let μ_G^θ be the distribution of g_θ obtained by picking $g \sim \mu_G$ and then generating g_θ as in Eq. (5.123). In Lemma 5.12 we proved bounds for the total variation distances $\|\mu_G - \mu_G^\theta\|_{\text{TVD}}$, according to this bounds we need to chosse θ close to 1 in order to obtain a probability distribution close to Haar random. These bounds can be directly translated to bounds between FLO circuits. Let $V_\theta = \Pi(g_\theta)$ and let ν_G^θ denote the distribution of the corresponding quantum circuits obtained by appropriate representation Π of G. Since distribution of the Haar random FLO circuits $\nu_{\text{pas}}, \nu_{\text{pas}}$ are obtained in exactly the same way we get from the monotonicity of TV distance (cf. Section 5.2).

$$\begin{aligned} \left\| \nu_{\text{pas}} - \nu_{\text{pas}}^\theta \right\|_{\text{TVD}} &\leq 8N^2\Delta \ , \\ \left\| \nu_{\text{act}} - \nu_{\text{act}}^\theta \right\|_{\text{TVD}} &\leq 8N^2\Delta \ , \end{aligned} \tag{5.124}$$

where $\theta \in [1-\Delta, 1]$. From Lemma 5.20 we know that the output probabilities $R(\theta) = \text{tr}\left(|\mathbf{x}_0\rangle\langle\mathbf{x}_0| \, \Pi(g_\theta)\rho\Pi(g_\theta)^\dagger\right)$ can be written explicitly as rational functions of the parameter θ with degrees

$$\begin{aligned} \text{Passive FLO:} \quad \deg(R) &= (16N^2, 16N^2) \ , \\ \text{Active FLO:} \quad \deg(R) &= (32N^2, 32N^2) \ . \end{aligned} \tag{5.125}$$

Finally, the interpolation method we use for our rational functions is given in the next theorem.

Theorem 5.21 (Berlekamp-Welch for rational functions [Mov19]). *Let $R(\theta)$ be a rational function of degree $\deg(R) = (d_1, d_2)$, where d_1 is the degree of the numerator and d_2 the degree of the denominator. A set of points $\mathcal{S} = \{(\theta_1, r_1), (\theta_2, r_2), \ldots, (\theta_L, r_L)\}$ specifies $R(\theta)$ uniquely provided $L > d_1 + d_2 + 2t$, where*

$$|\{ i \in [L] \mid R(\theta_i) \neq r_i \}| \leq t \ . \tag{5.126}$$

Moreover, $R(\theta)$ can be recovered in polynomial time in L and $\deg(R)$, when $\mathcal{S}$ is given.

With all this in place, we can now turn to show the average to worst-case reduction.

Theorem 5.22 (Average-case #P-hardness of computation of outcome probabilities of FLO circuits). *Let V_0 be a FLO circuit such that computing $p_{\mathbf{x}_0}(V_0, \Psi_{\text{in}}) = |\langle \mathbf{x}_0| V_0 |\Psi_{\text{in}}\rangle|^2$ is #P-Hard, where V_0 is element of either passive or active FLO circuits and the output Fock state $|\mathbf{x}_0\rangle$ belongs to the suitable Hilbert space $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ for passive FLO and $\mathcal{H}_{\text{act}} = \mathcal{H}^{+}_{\text{Fock}}(\mathbb{C}^{4N})$, respectively.*

Then it is #P-Hard to compute $p_{\mathbf{x}_0}(V, \Psi_{\text{in}}) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2$ with probability $\alpha > \frac{3}{4}+\delta$, $\delta = \frac{1}{\text{poly}\, N}$, over the the uniform distribution of circuits: $V \sim \nu_{\text{pas}}$ for passive FLO and $V \sim \nu_{\text{act}}$ for active FLO.

Remark 5.23. Due to hiding property (see Lemma 5.8 both active and passive FLO gates can permute between possible output Fock states $|\mathbf{x}\rangle$ in $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ and $\mathcal{H}_{\text{act}} = \mathcal{H}^{+}_{\text{Fock}}(\mathbb{C}^{4N})$, respectively. Therefore, using the invariance of the Haar measure on G, we can transform $\mathbf{x}_0$ above into any other output $\mathbf{x}$ satisfying $|\mathbf{x}| = 2N$ (for passive FLO) and $|\mathbf{x}|$ even (for active FLO).

Proof. Throughout this proof we will work with the fixed group G which will either stand for $\mathrm{U}(d)$ or $\mathrm{SO}(2d)$, the steps will be the same in both cases. Consider an oracle O which, when given as input the description of a FLO circuit $\Pi(g)$ representing a group element $g \in G$, computes exactly the quantity $|\langle \mathbf{x}_0| \Pi(g) |\Psi_{\text{in}}\rangle|^2$ with high probability over the group elements g.

$$\Pr_{g\sim\mu_G}\left[O(\Pi(g)) = |\langle \mathbf{x}_0| \Pi(g) |\Psi_{\text{in}}\rangle|^2\right] > \alpha\,. \tag{5.127}$$

The uniform distribution of FLO circuits ν_G is obtained by setting $V = \Pi(g)$, where $g \sim \mu_G$ (recall that $\nu_G = \nu_{\text{pas}}$ for $G = \mathrm{U}(4N)$ and $\nu_G = \nu_{\text{act}}$ for $G = \mathrm{SO}(8N)$). Therefore Eq. (5.127) is equivalent to

$$\Pr_{V\sim\nu_G}\left[O(V) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2\right] > \alpha\,. \tag{5.128}$$

We will argue that oracle O is able to compute output probabilities even when they are #P-hard, in polynomial time. We will follow the worst-to-average-case reduction from [AA11] and modify it accordingly for the rational interpolation technique in [Mov19].

For the rational interpolation, consider $g_\theta = g_0 F_\theta(g)$ where g_0 will be a group element giving rise to a #P-hard worst-case circuit. That such circuits exist can be seen from the fact that the input states $|\Psi_{\text{in}}\rangle$ make the circuit universal, moreover, in Appendix A.1 we argue that we only require shallow depth to implement these worst-case circuits. We will call the oracle O on L times, each time with different randomly picked $\Pi(g_{\theta_1}), \Pi(g_{\theta_2}), \cdots, \Pi(g_{\theta_L})$, where $\theta_i \in [1-\Delta, 1]$. Then, we can interpolate the polynomial using the Berlekamp-Welch algorithm for $R(\theta) = |\langle \mathbf{x}_0| \Pi(g_\theta) |\Psi_{\text{in}}\rangle|^2$ (see Theorem 5.21). Once we have interpolated the polynomial we can then evaluate $R(0) = |\langle \mathbf{x}_0| V_0 |\Psi_{\text{in}}\rangle|^2$, where $V_0 = \Pi(g_0)$.

Since we are sampling from ν_G^θ instead of the Haar random distribution ν_G, we need to estimate the deviation from the ideal distribution. We have done this in the bounds from Eq. (5.124) which gives

$$\Pr_{V\sim\nu_G}\left[O(V) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2\right] - \Pr_{V\sim\nu_G^\theta}\left[O(V) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2\right] \leq CN^2\Delta\,. \tag{5.129}$$

Where $C = 8$. Combining the above with Eq. (5.128) we get

$$\Pr_{V\sim\nu_G^\theta}\left[O(V) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2\right] \geq \alpha - CN^2\Delta\,. \tag{5.130}$$

Or equivalently, by noting that $V = \Pi(g_\theta)$ due to the definition of ν_G^θ,

$$\Pr_{g\sim\mu_G}\left[O(\Pi(g_\theta)) = |\langle \mathbf{x}_0| \Pi(g_\theta) |\Psi_{\text{in}}\rangle|^2\right] \geq \alpha - CN^2\Delta\,. \tag{5.131}$$

According to the Berlekamp-Welch algorithms we must evaluate the rational polynomial $R(\theta)$ in $L > d_1 + d_2 + 2t$ points. Where t are points where the polynomial is evaluated incorrectly. We have already computed the degrees of the numerator and denominator for $R(\theta)$ in (5.125), which gives $d_1 + d_2 = \Theta(N^2)$. We shall now estimate the number of errors which would make the interpolation method fail. Define the set of θ_i that make the oracle give a wrong answer for a given $g \in G$, the number of such θ_i is

$$t(g) = \left|\left\{\theta_i \,|\, O(\Pi(g_{\theta_i})) \neq |\langle \mathbf{x}_0| \Pi(g_{\theta_i}) |\Psi_{\text{in}}\rangle|^2,\ i \in [L]\right\}\right|. \tag{5.132}$$

From the definition of t and the inequality (5.131) it follows that $\mathbb{E}_{g\sim\mu_G} t(g) \leq [1-\alpha+CN^2\Delta]L$, since $t(g)$ follows the binomial distribution. Using this estimate in Markov inequality (recall that by assumption $\alpha > \frac{3}{4}+\delta$, for $\delta = \frac{1}{\text{poly}\, N}$) we get

$$\begin{aligned}\Pr_{g\sim\mu_G}\left[t(g) > \frac{L-d_1-d_2}{2}\right] &\leq \frac{[1-\alpha+CN^2\Delta]L}{\frac{L-d_1-d_2}{2}} \\ &\leq \frac{\frac{1}{4}-\delta+CN^2\Delta}{\frac{1}{2}-\frac{d_1+d_2}{2L}}\,.\end{aligned} \tag{5.133}$$

By choosing Δ and L such that $CN^2\Delta \leq \frac{\delta}{2}$ and $\frac{d_1+d_2}{2L} \leq \frac{\delta}{4}$ (this can be done with $\Delta = \frac{1}{\text{poly}\, N}$ and $L = \text{poly}\, N$ because $d_1+d_2 = \Theta(N^2)$), we obtain

$$\Pr_{g\sim\mu_G}\left[t(g) > \frac{L-d_1-d_2}{2}\right] \leq \frac{\frac{1}{4}-\frac{\delta}{2}}{\frac{1}{2}-\frac{\delta}{4}} \leq \frac{1}{2}-\frac{\delta}{4}\,. \tag{5.134}$$

The leftmost part of the above inequality is the probability of failure of our protocol. Therefore, since $\delta = \frac{1}{\text{poly}\, N}$, we can repeat the procedure polynomially many number of times, for different choices of $\Pi(g)$, compute $R\Pi(g)(0)$ each time, and output the majority vote. The probability of successfully computing the right result (i.e., $|\langle \mathbf{x}_0|\, V_0\, |\Psi_{\text{in}}\rangle|^2$) can be made exponentially close to 1 in this way. ■

We have chonse the #P-hardness of computing exactly the output probability distributions over a fraction of the circuit instances. Now we move into showing that this problem stays hard when approximating the output probabilities. To show this robustness, we will some known results from the theory of polynomials.

Lemma 5.24 (Paturi lemma [Pat92]). *Let $P(\theta)$ be a polynomial of degree k and suppose that $|P(\theta)| \leq \epsilon$ for $\theta \in [1-\Delta, 1]$, $\Delta \in (0,1]$. Then*

$$P(0) \leq \epsilon\, \exp\left(4k(1+\Delta^{-1})\right)\,. \tag{5.135}$$

Theorem 5.25 (Values of polynomials bounded at equally spaced points [CR92]). *Let θ_i, $i = 1,\ldots,L$ be a collection of L equally spaced points in the interval $[1-\Delta, 1]$, $\Delta \in (0,1)$. Let $P(\theta)$ be a polynomial of degree k. Assume that for every i, $|P(\theta_i)| \leq \epsilon$.*

Then there exist absolute constants $a, b > 0$ such that

$$\max_{\theta\in[1-\Delta,1]} |P(\theta)| \leq \epsilon \, \exp\left(b\frac{k^2}{L} + a\right) . \tag{5.136}$$

These theorems will be key in the proof of robustness, as they will allow to bound the error at $\theta = 0$. We recall here that the output probability $R_{g_0,g}(\theta) = |\langle \mathbf{x}_0| \, \Pi(g_\theta) \, |\Psi_{\text{in}}\rangle|^2$ for passive or active FLO is

$$R_{g_0,g}(\theta) = \frac{D_{g_0,g}(\theta)}{Q_g(\theta)} , \tag{5.137}$$

where we have shown in Lemma 5.20 that the degrees of the numerator and denominator are $D_{g_0,g} = d_1 = \Theta(N^2)$, $Q_g = d_2 = \Theta(N^2)$, moreover it can be seen from the proof that these are computable in polynomial time. Finally, we will need the following Corollary

Corollary 5.26. *Let $g \in G$ and let $Q_g(\theta)$ be the polynomial in defined in (5.122) for $G = \mathrm{U}(d)$ in $G = \mathrm{SO}(2d)$. Assume that $n = 2N$, $d = 4N$. Let now $\tilde{\Delta} > 0$. We then have the following inequalities*

$$\Pr_{g\sim\mu_{\mathrm{U}(d)}}\left(Q_g(\theta) \leq \left[1 + \left(\frac{\theta\pi}{\tilde{\Delta}}\right)^2\right]^{16N^2}\right) \geq 1 - 4N\frac{\tilde{\Delta}}{\pi} , \tag{5.138}$$

$$\Pr_{g\sim\mu_{\mathrm{SO}(2d)}}\left(Q_g(\theta) \leq \left[1 + \left(\frac{\theta\pi}{\tilde{\Delta}}\right)^2\right]^{32N^2}\right) \geq 1 - 4N\frac{\tilde{\Delta}}{\pi} . \tag{5.139}$$

Proof. We will show the passive case as the active case is identical. In Lemma 5.20 we show that $Q_g(\theta) = \left|\mathcal{Q}_g(\theta)^n\right|^2$, since $\mathcal{Q}_g(\theta)$ appears as the denominator in the entries of g_θ. Then, by Lemma 5.13 the result follows. ∎

We mow proceed to prove the main theorem regarding the robustness of our sampling scheme.

Theorem 5.27 (Average-case #P-hardness of approximation outcome probabilities of FLO circuits). *Let V_0 be a FLO circuit such that computing $p_{\mathbf{x}_0}(V_0, \Psi_{\text{in}}) = |\langle \mathbf{x}_0| \, V_0 \, |\Psi_{\text{in}}\rangle|^2$ is #P-Hard, where V_0 is element of either passive or active FLO circuits and the output Fock state $|\mathbf{x}_0\rangle$ belongs to the suitable Hilbert space $\mathcal{H}_{\text{pas}} = \bigwedge^{2N}(\mathbb{C}^{4N})$ for passive FLO and $\mathcal{H}_{\text{act}} = \mathcal{H}^+_{\text{Fock}}(\mathbb{C}^{4N})$, respectively.*

Let $\epsilon = \exp\left(-\Theta(N^6)\right)$. *Then it is* #P*-Hard to compute* $p_{\mathbf{x}_0}(V, \Psi_{\text{in}}) = |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2$ *to accuracy* ϵ *with probability* $\alpha > 1 - \delta$, $\delta = o(N^{-2})$, *over the the uniform distribution of circuits:* $V \sim \nu_{\text{pas}}$ *for passive FLO and* $V \sim \nu_{\text{act}}$ *for active FLO.*

Proof. Once again we use G to denote either $\mathrm{U}(d)$ or $\mathrm{SO}(2d)$ with $d = 4N$. As in Theorem 5.22, we define assume there is an oracle O which when given a description of an FLO circuit $\Pi(g)$, it approximates the ourput probability $p_{\mathbf{x}_0}(V, \Psi_{\text{in}}) = |\langle \mathbf{x}_0| \Pi(g) |\Psi_{\text{in}}\rangle|^2$ with high probability, i.e.,

$$\Pr_{g\sim\mu_G}\left[\left|O(\Pi(g)) - |\langle \mathbf{x}_0| \Pi(g) |\Psi_{\text{in}}\rangle|^2\right| \leq \epsilon\right] > 1 - \delta\,. \tag{5.140}$$

When translated to circuits, we have

$$\Pr_{V\sim\nu_G}\left[\left|O(V) - |\langle \mathbf{x}_0| V |\Psi_{\text{in}}\rangle|^2\right| \leq \epsilon\right] > 1 - \delta\,. \tag{5.141}$$

Once again we consider a deformation path where $g_\theta = g_0 F_\theta(g)$ where g_0 is a #P-hard instance and $g \in G$ is Haar random. We pick θ_i for $i = 1, \cdots, L$ equally distributed points in $[1 - \Delta, 1]$. As in Theorem 5.22, we obtain that $\forall \theta_i \in [1 - \Delta, 1]$

$$\Pr_{g\sim\mu_G}\left[\left|O(\Pi(g_{\theta_i})) - |\langle \mathbf{x}_0| \Pi(g_{\theta_i}) |\Psi_{\text{in}}\rangle|^2\right| \leq \epsilon\right] > 1 - \delta - 8\Delta N^2\,, \tag{5.142}$$

In Eq. (5.137) we gave the expression for the rational polynomial $R_{g_0,g}$. The numerator $D_{g_0,g}(\theta)$ is of degree $\Theta(N^2)$. As explained previously, the denominator $Q_g(\theta)$ is efficiently computable, this allows to construct an oracle $\tilde{O}$ from O which computes approximations of the numerator $D_{g_0,g}(\theta)$ with high probability.

$$\Pr_{g\sim\mu_G}\left[|\tilde{O}(\Pi(g_{\theta_i})) - D_{g_0,g}(\theta_i)| \leq \epsilon Q_g(\theta_i)\right] > 1 - \delta - 8\Delta N^2\,, \tag{5.143}$$

Now we can apply the bound from Corollary 5.26

$$\Pr_{g\sim\mu_G}\left[Q_g(\theta) \leq \left[1 + \left(\frac{\theta\pi}{\tilde{\Delta}}\right)^2\right]^{16N^2}\right] \geq 1 - 4N\frac{\tilde{\Delta}}{\pi}\,, \tag{5.144}$$

and since $1 + x \leq e^x$ and $\theta \leq 1$ we have that

$$\Pr_{g\sim\mu_G}\left[Q_g(\theta) \leq \exp\left(\frac{A}{\tilde{\Delta}^2}N^2\right)\right] \geq 1 - 4N\frac{\tilde{\Delta}}{\pi}\,, \tag{5.145}$$

where $\tilde{\Delta} > 0$ and A is a numerical constant which depends on whether $G = \mathrm{U}(4N)$ or $G = \mathrm{SO}(8N)$. Now call $\mathcal{A}$ the event that $|\tilde{O}(\Pi(g_{\theta_i})) - D_{g_0,g}(\theta_i)| \leq \epsilon Q_g(\theta_i)$ and $\mathcal{B}$ the event that $Q_g(\theta) \leq \exp\left(\frac{A}{\tilde{\Delta}^2}N^2\right)$, then $\Pr(\mathcal{A} \cap \mathcal{B}) \geq \Pr(\mathcal{A}) + \Pr(\mathcal{B}) - 1$. Thus,

$$\Pr_{g\sim\mu_G}\left[|\tilde{O}(\Pi(g_{\theta_i}))-D_{g_0,g}(\theta_i)| \leq \epsilon \exp\left(\frac{A}{\tilde{\Delta}^2}N^2\right)\right] > 1 - \delta - 8\Delta N^2 - 4N\frac{\tilde{\Delta}}{\pi}, \tag{5.146}$$

We finally use union bound lower to bound the probability that $\tilde{O}$ is successful *for all* L equally spaced θ_i in $[1-\Delta, 1]$:

$$\Pr_{g\sim\mu_G}\left[\forall\theta_i\ |\tilde{O}(\Pi(g_{\theta_i})) - D_{g_0,g}(\theta_i)| \leq \epsilon \exp\left(\frac{A}{\tilde{\Delta}^2}N^2\right)\right] > 1 - L(\delta + 8\Delta N^2 + 4N\frac{\tilde{\Delta}}{\pi}), \tag{5.147}$$

As argued previously, we set $L = \Theta(N^2)$ for the interpolation of a poynomial $\tilde{P}_{g_0,g}$ passing through points $(\theta_i, \tilde{O}(\Pi(g_{\theta_i})))$ and having identical degree to $D_{g_0,g}$. By Theorem 5.25, if the event on the left side of Eq. (5.147) is obtained, then

$$\begin{aligned}\max_{\theta\in[1-\Delta,1]} &\left|\tilde{P}_{g_0,g}(\theta) - D_{g_0,g}(\theta)\right| \\ &\leq \epsilon \exp\left(\frac{A}{\tilde{\Delta}^2}N^2\right)\exp\left(\Theta(N^2)\right) \\ &= \epsilon \exp\left(\frac{\Theta(N^2)}{\tilde{\Delta}^2}\right).\end{aligned} \tag{5.148}$$

Where in using Theorem 5.25 we have set $\tilde{P}_{g_0,g}(\theta) - D_{g_0,g}(\theta)$ as the polynomial, $k = \Theta(N^2)$ and $L = \Theta(N^2)$. Finally, we apply Lemma 5.24 for the polynomial $\tilde{D}_{g_0,g}(\theta) - D_{g_0,g}(\theta)$ we finally obtain

$$\left|\tilde{P}_{g_0,g}(0) - D_{g_0,g}(0)\right| \leq \epsilon \exp\left(\frac{\Theta(N^2)}{\tilde{\Delta}^2} + \Theta(N^2)(1+\Delta^{-1})\right) \tag{5.149}$$

To sum up, the initially assumed oracle O allows us to construct an efficient algorithm $\mathcal{A}$ that approximately computes #P-hard quantity $D_{g_0,g}(0) = Q_g(\theta)\,|\langle \mathbf{x}_0|\,\Pi(g_0)\,|\Psi_{\mathrm{in}}\rangle|^2$:

$$\begin{aligned}\Pr_{g\sim\mu_G}&\left[\left|\mathcal{A}(\Pi(g)) - D_{g_0,g}(0)\right| \leq \tilde{\epsilon}\right] \\ &> 1 - BN^2\left(\delta + 8\Delta N^2 + 4N\frac{\tilde{\Delta}}{\pi}\right),\end{aligned} \tag{5.150}$$

where $\tilde{\epsilon} = \epsilon \exp\left(\frac{\Theta(N^2)}{\tilde{\Delta}^2} + \Theta(N^2)(1+\Delta^{-1})\right)$, and $B > 0$ is a numerical constant. Success probability of the protocol exceeds $\frac{1}{2}$ with the following scaling

$$\Delta = \Theta(N^{-4}) \ , \ \tilde{\Delta} = \Theta(N^{-3}) \ . \tag{5.151}$$

From the result of [DGGJ00] we have #P hardness guarantees up to constant multiplicative error. Since for #P-hard quantity this such error implies additive error of magnitude at most $2^{-\Theta(N)}$. Therefore by setting $\tilde{\epsilon} \leq 2^{-\Theta(N)}$ which, by the virtue of Eq.(5.151) corresponds to a scaling of the original error $\epsilon = \exp\left(-\Theta(N^7)\right)$. ■

Since the publication of this work, the exponent in the error of RCS has been improved [BFLL22, KMM22] with techniques that could apply. More recently, it has been claimed [Kro22] that the error could scale as $2^{-O(m)}$ where m is the number of gates, whether this technique would apply for our scheme remains open.

5.8 Conclusion

In this work we have shown that FLO circuits when provided suitable inputs are able to provide a sampling quantum advantage with similar hardness guarantees as in Random Circuit Sampling and in Boson Sampling. In contrast to Boson Sampling, we are able to show anticoncentration. Then, we show the likely classical intractability of our Fermionic sampling task based on two plausible conjectures. We provide support for one of these conjectures by proving a weaker result, namely, we show that computing approximations of output probabilities is #P-hard up to $2^{-\Theta(N)}$ additive error.

An interesting open question is if anticoncentration can be obtained in logarithmic depth with matchgate circuits. We have performed some initial numerical studies [ODMZ22] which give evidence that logarithmic depth should be enough for anticoncentration. Another important open question is whether the error can be improved to show hardness, this is in fact one of the central questions in this area for all sampling schemes.

Part III

Improved Product Formulae

Chapter 6

Improved Product Formulae for Quantum Simulation

This chapter is based on work in [MCB+22]. This is work done in collaboration with Pedro C.S. Costa, Daniel K. Burgarth, Yuval R. Sanders and Dominic W. Berry. We improve over previous methods to implement product formulae of 8th and 10th order. We compare numerically the performance of previous methods with ours and find that 8th order product formulae may be the best option in quantum chemistry.

In this work I contributed in the idea, writing and numerics. All the theorems, lemmas and corollaries included are stated as in [MCB+22] and also the figures. Not much is changed from the original publication as much of the writing was done by myself with corrections from my coauthors. Those parts written by coauthors have not been included in this chapter.

6.1 Introduction

Trotter formulae are expected to be relevant for both noisy intermediate-scale quantum computation and fault-tolerant computation. It is then of great importance to seek efficient implementations of product formulae (PF) as it can have a great impact on

the efficiency of Hamiltonian simulation algorithms in practice. In Section 2.4.1 we introduced the basic Lie-Trotter formula which was first applied for quantum simulation in [Llo96]. Later work considered the broader class of sparse Hamiltonians [ATS03] and further work has focused on improving this simulations in terms of queries to the simulated Hamiltonian [BACS07, BC12, BCC+14, BCK15, Low19].

Recent results have shown that despite its simplicity, the Lie-Trotter product formula can compete with other Hamiltonian simulation algorithms due to the low error that it achieves in practice [CST+21]. Methods based on linear combinations of unitaries [CW12, BCC+15] or quantum signal processing [LC17b] give complexity logarithmic in the inverse error, but the error is not required to be extremely small, meaning those methods do not provide as large an advantage as might be expected. Product formula error bounds can be further improved by considering particular physical systems [BMW+15, CMN+18, SHC21] or leveraging randomization [Zha12, Cam19, COS19]. Trotter formulae are expected to be relevant for both noisy intermediate-scale quantum computation and fault-tolerant computation, any improvements in their implementation will have great impact on the efficiency of Hamiltonian simulation algorithms..

The error in the implementation of product formulae can be improved by using higher-order product formulae, like those given by Suzuki's method [Suz90, Suz91]. This method has the advantage that it can give arbitrary high-order scaling $O(t^k)$. The downside of the Suzuki method to generate higher-order formulae is that the number of exponential operators to implement it grows very rapidly. Suzuki's product formulae are usually assumed in quantum computing, but they can be greatly improved upon. An alternative method by Yoshida [Yos90] can be used to obtain product formulae with a smaller number of exponentials. This method is based on an ansatz written as the product of symmetric product formulae of 2nd order, then by imposing certain conditions on this ansatz a set of polynomial equations is obtained in terms of some free parameters that define the product formula. Solving these polynomial equations in terms of the free parameters requires using numerical methods. Yoshida gives what appears to be all 6th order solutions but only some 8th order integrators, and did not consider any

orders beyond that. That work from 1990 was limited by computing power, because it requires numerical solution of a system of nonlinear equations. More recent work [KL97, SS05, BCM08, BCM06] has pushed the search to higher orders, new methods such as post-processing have been devised which allow high-order product formulae with fewer use of exponentials asymptotically.

The objective of this work is to perform a comparison of several high-order product formulae which could be useful for Hamiltonian simulation applications and propose new product formulae that improve over previous ones. We have also extended Yoshida's method to derive a set of polynomial equations that need to be satisfied for a 10th order product formula. Our method for generating higher-order product formulae is based on a Taylor expansion of the exponential operators in a PF and then describing it as a non-commutative polynomial representing it with a tree data structure. Our method allows to represent new product formulae as solutions of systems of polynomial equations with multiple variables. Although the number of equations and variables grows quickly with the order of the PF desired, the tree data structure allows us to solve this equations up to 10th order. We have solved this equations for 8th and 10th order, we have found over 800 examples in the case of 8th order and 600 examples in 10th order, we also manage to recover results from previous literature. We have an example of an 8th order product formula with greatly reduced error, more than 100 times better than any of Yoshida's 8th order product formulae. This formula also provides greatly improved performance beyond that for any of the product formulae found by Suzuki's method when accounting for the number of exponentials. As well as finding solutions with a minimal number of exponentials, we have found solutions with extra exponentials at both 8th and 10th order that provide a further order of magnitude reduction in the error (so a factor of 1000 improvement over Yoshida for 8th order). Our 8th order product formulae also turn out to improve over other PF (at least in the context of quantum simulation) such as those found in [KL97]. In the comparison we also include post-processing techniques from [BCM06] and also higher-order product formulae from [SS05]. A detailed list of the product formulae considered in this work can be found in Section 6.6.

When comparing product formulae of different orders, it is better to use higher-order formulae for larger values of T/ϵ, where T is the total evolution time and ϵ is the required precision. This is because at different order, the PF differ in the number of exponentials and also the errors achieved. We compare these ratios of T/ϵ for unformly random Hamiltonians and random Hamiltonians that have a similar form as those in chemistry. We find that the threshold where our best 8th order formula outperforms lower-order formulae is quite modest at about 1600, which is well below the typical values for quantum chemistry applications. Moreover, even the best 6th order formula has poor performance, so as T/ϵ is increased, one should go straight from 4th to 8th order, and not use 6th order at all. The threshold of T/ϵ for using 10th order instead of 8th order is large, over 10^{11}, which is beyond expected values for quantum computing. As a result we expect that our 8th order product formula will be the best to use for quantum algorithms for applications in quantum chemistry.

We begin with a background summary in Section 6.2.1. We review first the Lie-Trotter product formulae and its higher-order versions. Then we present the Baker-Campbell-Hausdorff formula and how it is applied in the Suzuki method to generate higher-order product formulae and compare it with Yoshida's method in terms of cost of implementation. Moreover, we give a brief review of other recent methods to generate higher-order formulae. In Section 6.3 we introduce a method for obtaining higher-order product formulae based on a Taylor expansion of operators and studying these decompositions as non-commutative polynomials. We will use these decompositions to solve for new higher order formulae, this procedure is described at the end of that section. Section 6.4 provides results based on our numerics for 8th order product formulae. We give a list of the best solutions found by us and some metrics of comparison. Section 6.5 gives analogous results for 10th order. In Section 6.6 we provide details on the comparison between our best performing product formulae and previous ones in the literature.

6.2 Background

In this section we give a summary of the background for our work. We begin by defining product formulae and the Baker-Campbell-Hausdorff formula, then we introduce the Suzuki method and Yoshida method to obtain higher-order formulae. We conclude the section comparing the advantages of Yoshida's method over Suzuki's in terms of number of exponentials require to implement it.

6.2.1 Product formulae

It is well known that, for any non-commuting operators X and Y,

$$\exp((X+Y)t) = \exp(Xt)\exp(Yt) + O\left(t^2\right). \tag{6.1}$$

where we have absorbed the $-i$ factor used in quantum simulation into X and Y. The above equation demonstrates that the exponential of a sum of two operators is, to first order, equal to the product of the exponential of those operators. The above equation is often referred as a 'first-order product formula'. Higher-order terms can be computed via the Baker-Campbell-Haussdorff (BCH) formula.

Theorem 6.1 (Baker-Campbell-Haussdorff formula [BC04]). *Let X and Y be any operators such that $\|X\| + \|Y\| < \ln 2$. We have $\exp(X)\exp(Y) = \exp(Z)$, with*

$$Z = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \sum_{\substack{r_1+s_1>0 \\ \vdots \\ r_n+s_n>0}} \frac{[X^{r_1}, Y^{s_1}, \cdots X^{r_n}, Y^{s_n}]}{\left(\sum_{j=1}^{n} r_i + s_i\right) \prod_{i=1}^{n} r_i! s_i!}, \tag{6.2}$$

where

$$[X^{r_1}, Y^{s_1}, \cdots X^{r_n}, Y^{s_n}] = [\underbrace{X, [X, \cdots [X}_{r_1}, [\underbrace{Y, [Y, \cdots [Y}_{s_1}, \cdots [\underbrace{X, [X, \cdots [X}_{r_n}, [\underbrace{Y, [Y, \cdots Y}_{s_n}]] \cdots]].$$

The standard second-order symmetric product formula is as given in the definition below.

Definition 6.2 (Symmetric product formula of order two). Let X and Y be non-commuting operators and let t be a real variable. Define

$$S_2(t) := \exp\left(\frac{1}{2}Xt\right)\exp(Yt)\exp\left(\frac{1}{2}Xt\right). \tag{6.3}$$

The operators X and Y used in the definition of S_2 should always be clear from context. More generally, when there is a sum of X_j, the product formula is

$$S_2(t) := \left[\prod_{j=1}^{J}\exp\left(\frac{1}{2}X_j t\right)\right]\left[\prod_{j=J}^{1}\exp\left(\frac{1}{2}X_j t\right)\right]. \tag{6.4}$$

We take the canonical convention that products are ordered with the starting index on the right and the final one on the left. When there are $J = 2$ terms, the expression in the definition is obtained. The corollary below describes the form of the error terms in the symmetric product formula, and also implies that it is second order.

Corollary 6.3 (Symmetric BCH formula [Yos90]). *Let X and Y be any operators such that $\|X\| + \|Y\| < \ln 2$ and let t be a real variable. Define Z such that $S_2(t) = \exp(Z)$. Then there is a sequence α_ℓ consisting of linear combinations of ℓ-term commutators in X and Y such that*

$$Z = \sum_{\ell=1}^{\infty}\alpha_\ell t^\ell. \tag{6.5}$$

Moreover, $\alpha_\ell \equiv 0$ whenever ℓ is even.

Reference [Yos90] also shows that even terms are zero for more general symmetric product formulae. The first three non-zero α_ℓ terms from above are

$$\alpha_1 = X + Y, \tag{6.6}$$

$$\alpha_3 = \frac{1}{12}[Y,[Y,X]] - \frac{1}{24}[X,[X,Y]], \tag{6.7}$$

$$\alpha_5 = \frac{7}{5760}[X,X,X,X,Y] - \frac{1}{720}[Y,Y,Y,Y,X] + \frac{1}{360}[X,Y,Y,Y,X] + \frac{1}{360}[Y,X,X,X,Y] - \frac{1}{480}[X,X,Y,Y,X] \quad + \frac{1}{120}[Y,Y,X,X,Y]. \tag{6.8}$$

Here the square brackets are used to indicate multicommutator expressions similar to the notation in Theorem 6.1, for example

$$[Y,Y,X,X,Y] \equiv [Y,[Y,[X,[X,Y]]]]. \tag{6.9}$$

Expressions for each α_ℓ, and hence the proof of the symmetric BCH formula, arise from two applications of the usual BCH formula.

Definition 6.4 (Product formula). Let X and Y be any non-commuting operators. Given a natural number n, a *product formula of order n* is a pair $(\vec{c}, \vec{d})$ with $\vec{c}, \vec{d} \in \mathbb{R}^\ell$ and ℓ a natural number such that for all $t \in \mathbb{R}$

$$\exp((X+Y)t) = \prod_{j=1}^{\ell} \exp\left(c_j X t\right) \exp\left(d_j Y t\right) + O\left(t^{n+1}\right). \tag{6.10}$$

We refer to the number of non-zero coefficients in $\left(\vec{c}, \vec{d}\right)$ as the length of the product formula.

The ordinary BCH formula is a length-2 product formula of order 1. The symmetric BCH formula is a length-3 product formula of order 2.

Problem 2 (Minimal-length product formulae). Given a natural number n, find a natural number ℓ_n such that there is an n^{th}-order product formula of length ℓ_n but no n^{th}-order product formula of length $\ell_n - 1$.

It is easy to prove using Taylor expansions (for lower bounds) and the BCH formulae (for equality) that $\ell_1 = 2$ and $\ell_2 = 3$. As we discuss below, Suzuki's 'fractal' method demonstrates that $\ell_n = O(\exp n)$. But it may be possible to do better.

Suzuki method for generating higher-order product formulae. Here we describe Suzuki's fractal methods from [Suz90, Suz91] to obtain higher-order product formulae. Starting from the symmetrised product formula in Eq. (6.3), the fractal method generalises this expression to obtain product formulae at all even orders. Suzuki's first fractal method to generate a product formula of order $k = 2\kappa$ is [Suz90]

$$S_{2\kappa}(t) = S_{2\kappa-2}(s_\kappa t) S_{2\kappa-2}((1-2s_\kappa)t) S_{2\kappa-2}(s_\kappa t), \tag{6.11}$$

where $s_\kappa = 1/(2 - 2^{1/(2\kappa-1)})$. This method can be used to generate even orders starting at S_2. A drawback to this method is that both s_k and $1 - 2s_\kappa$ are greater than 1, so the coefficients in the higher-order methods are large, causing greater error.

Alternatively to Eq. (6.11), an order 2κ product formula can be obtained via Suzuki's second fractal method [Suz91]

$$\tilde{S}_{2\kappa}(t) = \tilde{S}_{2\kappa-2}(u_\kappa t)^2 \tilde{S}_{2\kappa-2}((1-4u_\kappa)t)\tilde{S}_{2\kappa-2}(u_\kappa t)^2, \tag{6.12}$$

where $u_\kappa = 1/(4 - 4^{1/(2\kappa-1)})$. This method has the advantage that both u_κ and $1 - 4u_\kappa$ are less than 1, so the coefficients of higher-order formulae are small resulting in small error. The drawback is that far more exponentials are required. Each iteration uses 5 copies of the lower-order formula, whereas the previous one uses 3 copies. One can use either of these iterative methods from any product formula of order $2\kappa - 2$ to obtain a formula of order 2κ, rather than simply iterating from S_2.

One of the virtues of the fractal method is that it allows one to generate arbitrarily high-order product formulae easily, as there are formulae for the coefficients s_κ and u_κ. A further advantage is that these methods work with arbitrary numbers of terms in the sum X_j. The number of exponentials used in the product formula is a crucial measure of its efficiency, and in quantum simulation it is proportional to the required number of gates.

Exponential length scaling of the Suzuki method. We can easily compute the number of exponentials that the Suzuki method gives assuming we start from S_2 and generate higher-order integrators. The total number of exponentials for a given order $2\kappa = 4, 6, 8, ...$ in the product formula $S_{2\kappa}$ is given by

$$2(J-1)3^{\kappa-1} + 1, \tag{6.13}$$

when considering exponentials of sums of J terms, so for example $J = 2$ for $X + Y$ and $\kappa = 1$ for second order gives 3. For the product formula $\tilde{S}_{2\kappa}$ the number of exponentials is

$$2(J-1)5^{\kappa-1} + 1. \tag{6.14}$$

The number of exponential operators in both cases of the Suzuki method grows very rapidly. Thus one may be interested in alternative method to obtain product formulae with a lower count, such as the method of Yoshida in the next section.

6.2.2 Yoshida's method for deriving 6th order product formulae

In this section we introduce Yoshida's method [Yos90] which in contrast to Suzuki's method in the previous section, requires fewer exponentials in the implementation. As we will see in this section, a downside is that Yoshida's method can only generate new product formulae by solving systems of polynomial equation which are not trivial to solve as the order increases.

Approach. Rather than using Eqs. (6.11) and (6.12), Yoshida uses the general procedure

$$S^{(m)}(t) = \left(\prod_{j=1}^{m} S_2(w_{m-j+1}t)\right) S_2(w_0 t) \left(\prod_{j=1}^{m} S_2(w_j t)\right), \tag{6.15}$$

where $w_j \in \mathbb{R}$ for $j = 0, 1, \cdots, m$ are parameters to be determined. Note the number of exponentials in this product is given by $(4m+2)(J-1)+1$. Given this ansatz, the task becomes to find m and w_i such that $S^{(m)}$ is an order k product formula. We will illustrate Yoshida's method by deriving the result for 6th order.

Expand Yoshida product using Baker-Campbell-Haussdorf formula. The method is to expand Eq. (6.15) using the BCH formula from Theorem 6.1 recursively as follows. First, note that from Corollary 6.3, $S_2(t) = e^{\frac{t}{2}X} e^{tY} e^{\frac{t}{2}X} = e^{t\alpha_1 + t^3\alpha_3 \cdots}$. We are for the moment considering sums of two terms $X+Y$. Define $C = tw_1\alpha_1 + t^3 w_1^3 \alpha_3 + t^5 w_1^5 \alpha_5 + O(t^7)$ and $D = tw_0\alpha_1 + t^3 w_0^3 \alpha_3 + tw_0^5 \alpha_5 + O(t^7)$. Then,

$$\begin{aligned}
&S_2(w_1 t) S_2(w_0 t) S_2(w_1 t) \\
&= e^C e^D e^C \\
&= \exp\left\{tw_1\alpha_1 + t^3 w_1^3 \alpha_3 + t^5 w_1^5 \alpha_5 + O(t^7)\right\} \exp\left\{tw_0\alpha_1 + t^3 w_0^3 \alpha_3 + tw_0^5 \alpha_5 + O(t^7)\right\} \\
&\quad \times \exp\left\{tw_1\alpha_1 + t^3 w_1^3 \alpha_3 + t^5 w_1^5 \alpha_5 + O(t^7)\right\} \\
&= \exp\left\{\tau(2w_1 + w_0)\alpha_1 + \tau^3(2w_1^3 + w_0^3)\alpha_3 + \tau^5(2w_1^5 + w_0^5)\alpha_5 + \frac{1}{6}([D,D,C] - [C,C,D]) + O(\tau^7)\right\}.
\end{aligned} \tag{6.16}$$

A simple computation shows

$$[D,D,C]-[C,C,D]=\tau^5(w_0^2w_1^3-w_1^2w_0^3+w_1^4w_0-w_0^4w_1)[\alpha_1,\alpha_1,\alpha_3]. \tag{6.17}$$

Define $\beta_5=[\alpha_1,\alpha_1,\alpha_3]$ so

$$S_2(w_1t)S_2(w_0\tau)S_2(w_1t)=\exp\left\{t(2w_1+w_0)\alpha_1+t^3(2w_1^3+w_0^3)\alpha_3+t^5(2w_1^5+w_0^5)\alpha_5 + t^5\frac{1}{6}(w_0^2w_1^3-w_1^2w_0^3+w_0w_1^4-w_0^4w_1)\beta_5+O\left(t^7\right)\right\}. \tag{6.18}$$

By an induction argument Yoshida shows that

$$S^{(m)}(t)=\exp\left\{tA_{1,m}\alpha_1+t^3A_{3,m}\alpha_3+t^5(A_{5,m}\alpha_5+B_{5,m}\beta_5)+O\left(t^7\right)\right\}, \tag{6.19}$$

where $A_{j,m}$ and $B_{5,m}$ are polynomials on the variables $w_0,\ldots,w_m$.

The case $m=0$ is just the symmetric BCH formula, so it is clear that Eq. (6.19) holds with

$$\begin{aligned}A_{1,0}&=w_0,\\ A_{3,0}&=w_0^3,\\ A_{5,0}&=w_0^5,\\ B_{5,0}&=0.\end{aligned} \tag{6.20}$$

To prove Eq. (6.19) for $m>0$, one needs to show that the exponential is of the form with operator α_1 for first order in t, operator α_3 for third order in t, and operators α_5 and β_5 for fifth order in t. This result may be shown using

$$\begin{aligned}S^{(m+1)}(t)&=S_2(w_{m+1}t)S^{(m)}(t)S_2(w_{m+1}t)\\ &=\exp\left\{tw_{m+1}\alpha_1+t^3w_{m+1}^3\alpha_3+t^5w_{m+1}^5\alpha_5+O\left(t^7\right)\right\}\\ &\quad\times\exp\left\{tA_{1,m}\alpha_1+t^3A_{3,m}\alpha_3+t^5(A_{5,m}\alpha_3+B_{5,m}\beta_5)+O\left(t^7\right)\right\}\\ &\quad\times\exp\left\{tw_{m+1}\alpha_1+t^3w_{m+1}^3\alpha_3+t^5w_{m+1}^5\alpha_5+O\left(t^7\right)\right\}\\ &=\exp\left\{2tw_{m+1}\alpha_1+tA_{1,m}\alpha_1+2t^3w_{m+1}^3\alpha_3+t^3A_{3,m}\alpha_3+2t^5w_{m+1}^5\alpha_5+t^5A_{5,m}\alpha_5+t^5B_{5,m}\beta_5\right.\end{aligned}$$

$$+\frac{1}{6}t^5(A_{1,m}^2 w_{m+1}^3 - A_{1,m}A_{3,m}w_{m+1} - w_{m+1}^2 A_{3,m} + w_{m+1}^4 A_{1,m})\beta_5 + O\left(t^7\right)\Big\}. \tag{6.21}$$

Hence, if the product formula can be expressed as in the form (6.19) for $S^{(m)}(t)$, it can again be expressed in this form for $S^{(m+1)}(t)$, establishing it for all $m \geq 0$ by induction. This expression also shows that the scalar coefficients can be determined from the formulae

$$\begin{aligned}
A_{1,m+1} &= 2w_{m+1} + A_{1,m},\\
A_{3,m+1} &= 2w_{m+1}^3 + A_{3,m},\\
A_{5,m+1} &= 2w_{m+1}^5 + A_{5,m},\\
B_{5,m+1} &= B_{5,m} + \frac{1}{6}(A_{1,m}^2 w_{m+1}^3 - A_{1,m}A_{3,m}w_{m+1} - w_{m+1}^2 A_{3,m} + w_{m+1}^4 A_{1,m}).
\end{aligned} \tag{6.22}$$

Constraints to satisfy in order to derive 6th order formula. To derive a 6th order formula, the lower-order terms in the exponential in Eq. (6.19) must be zero (see also Eq. (5.16) of [Yos90]), which gives the four conditions

$$A_{1,m} = 1, \quad A_{3,m} = 0, \quad A_{5,m} = 0, \quad B_{5,m} = 0. \tag{6.23}$$

For $m = 3$ there are four unknowns w_0 to w_3, and it can be expected there are solutions because there are the same number of equations as unknowns. In practice $A_{1,m} = 1$ is satisfied by taking $w_0 = 1 - 2\sum_j w_j$, so there are then three equations for three unknowns w_1, w_2, w_3. Whereas it is possible to solve the equations using the Newton-Raphson method, the expression for the appropriate Jacobian matrix is complicated, so Yoshida instead uses the Brent method. Yoshida produced three $m = 3$ solutions in this way, and states "It seems that there is no other solution." We have performed an extensive search and also find no more solutions.

The product formulae obtained through the Yoshida method also work for exponentials of sums of more operators. The S_2 product formula can again be used as the

building block for the product formula, and we can write

$$S_2(t) = \exp\left(\sum_{\ell=0}^{\infty} \tilde{\alpha}_\ell t^\ell\right), \tag{6.24}$$

where $\tilde{\alpha}_\ell$ are now order-ℓ multicommutator expressions on the J terms. The reasoning for finding the product formula is entirely based on the construction with α_ℓ, but without taking advantage of its particular form, so exactly the same reasoning holds for $\tilde{\alpha}_\ell$. This immediately implies that the higher-order product formulae work in general. This is an advantage of deriving product formulae as products of S_2, because deriving coefficients separately for exponentials of X and Y would not generalise.

For 8th order the number of equations must be at least $m = 7$, this makes even harder finding an optimal solution for the 8th order. Yoshida was able to find 5 solutions at this order but as we will see, we can greatly improve over the solutions that Yoshida has found.

6.2.3 Other high-order product formulae

We also include other product formulae found in the literature. Many results on product formulae were motivated by the search of integrators for differential equations, in this subsection we will point out to some of these results. In [BCM08] a summary of several integrators found is given (see Table 4 in the reference). Results from order 4 to 10 are reported in terms of the stages required to implement the product formulae, the number of stages corresponds to the number of S_2 product formulae required to implement the integrator.

Some integrators reported which we include in our comparison of profuct formulae for Hamiltonian simulations are of order 8 and 10 as we find that these are the most useful when considering error and number of exponential required to implement them. In [KL97] the authors provide 8th order product formulae of 15 and 17 stages, which correspond in Yoshida's ansatz to $m = 7$ and $m = 8$. These improve over previous reported 8th order formulae and we confirm this in our numerics.

Other product formulae based on the ansatz of Yoshida are also presented in [SS05]. For 8th order, the authors replicate the findings of [KL97] for product formulae with 15 and 17 stages. In addition, they provide solutions with 19 and 21 stages. For 10th order, the authors provide solutions with 31 stages (corresponding to $m = 15$ in Yoshida's case) and 33 states ($m = 16$ for Yoshida's).

Another technique to obtain higher-order product formulae is that of post-processing [BCM06]. In this technique a product formula S_κ of order κ is generated by the composition of a so-called kernel Σ conjugated by a post-processor operator P. More precisely,

$$S_k = P\Sigma P^{-1}. \tag{6.25}$$

The advantage of this method is that usually Σ uses less stages than other methods and due to the construction we have that $S_k^n = P\Sigma^n P^{-1}$, so the cost of using the post-processor is effectively constant when having to repeat the product formula many times, as it usually is when the total time of evolution is partitioned into intervals.

6.3 Solution using Taylor expansion

In this section we begin the exposition of the method used to find our solutions. Our solution method is based on computing the Taylor expansion of both the exact exponential and its product formula approximation with given w_j. This Taylor expansion is performed on both sides up to the desired order of approximation for the integrator. By imposing equality on terms of the same order we obtain a series of equations for w_j which can be solved. For higher orders, a large number of simultaneous equations are obtained, so we need an automated way of generating them.

Definitions used to specify problem. To make precise the problem we are solving, denote as $\mathcal{T}_k[\cdot]$ the map giving the Taylor expansion in t around 0, truncated at order k, so

$$\mathcal{T}_k[e^{t(X+Y)}] = \sum_{p=0}^{k} \frac{t^p}{p!}(X+Y)^p$$

$$= \sum_{p=0}^{k} \frac{t^p}{p!} \sum_{r_1,\cdots,r_p=0}^{1} X^{r_1} Y^{1-r_1} \cdots X^{r_p} Y^{1-r_p}. \tag{6.26}$$

We consider a sum of two operators $X + Y$, but this approach for solving for w_j will also be sufficient to provide product formulae for sums of arbitrary numbers of terms. That is because the solutions must also be solutions of the equations derived using Yoshida's method, and as explained above those equations will be the same when considering sums of arbitrary numbers of terms. Now consider the ansatz operator of Yoshida from Eq. (6.15)

$$\begin{aligned} S^{(m)}(t, w_1, \cdots, w_m) &= e^{tw_m X/2} e^{tw_m Y} e^{t(w_m+w_{m-1})X/2} e^{tw_{m-1}Y} e^{t(w_{m-1}+w_{m-2})X/2} \\ &\quad \cdots e^{tw_0 Y} e^{t(w_1+w_0)X/2} e^{tw_1 Y} \cdots e^{tw_m X/2} \\ &= e^{tc_1 X} e^{tc_2 Y} \cdots e^{tc_{4m+3}X}, \end{aligned} \tag{6.27}$$

where in the last line we have defined the constants $c_1 = w_m/2$, $c_2 = w_m$, $c_3 = (w_m + w_{m-1})/2, \cdots c_{4m+3} = w_m/2$. We Taylor expand this ansatz up order k, noting that the total number of exponentials in Yoshida's ansatz is $4m + 3$

$$\mathcal{T}_k[S^{(m)}(t, w_1, \ldots, w_m)] = \sum_{\substack{r_1,\cdots,r_{4m+3}=0 \\ r_1+\cdots+r_{4m+3}\leq k}}^{k} \frac{t^{r_1+\cdots+r_{4m+3}}}{r_1! \cdots r_{4m+3}!} c_1^{r_1} \ldots c_{4m+3}^{r_{4m+3}} X^{r_1} Y^{r_2} \ldots X^{r_{k-1}} Y^{r_{4m+3}}. \tag{6.28}$$

Data structure for coefficients. We require that the product formula obtained from our solution procedure be independent of the choice of X and Y, so need to match the coefficients for each sequence of products of X and Y. In order to do this in an automated way we construct a data structure to store the coefficients.

Given operators of the form $e^{cX} = I + cX + \frac{c^2}{2!}X^2 + \frac{c^3}{3!}X^3 + \cdots$ and $e^{dY} = I + dY + \frac{d^2}{2!}Y^2 + \frac{d^3}{3!}Y^3 + \cdots$ with c, d scalar coefficients and X, Y general operators, we can describe this Taylor expansion up to an order k using an array encoding. First, we write monomials composed of X and Y operators in lexicographical order and note that these operators

can be mapped to binary numbers:

$$\begin{array}{cccccccccc} I & X & Y & XX & XY & YX & YY & XXX & XXY & \cdots \\ 1 & 10 & 11 & 100 & 101 & 110 & 111 & 1000 & 1001 & \cdots \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \cdots \end{array} \tag{6.29}$$

To construct a bit string, we map each X with 0 and each Y with 1, then place a 1 on the left to flag the length of the string, as shown in the second line of Eq. (6.29). Then, to obtain the operator products, simply remove the leading 1 and then map 0 to X and 1 to Y. The empty string corresponds to the identity I. Now, to store the coefficients in a sum of products of X and Y, convert each product to a binary integer as above, then store the coefficient in the corresponding location in a vector.

By way of illustration, consider the polynomial $10I+3X+2Y+2X^2+YX$. This operator would be stored in an array as $[10, 3, 2, 2, 0, 1, 0, \cdots]$. In this way, any polynomial of X and Y can be efficiently stored in a vector. We denote this vector storing the coefficients of operators of order up to k as $\mathrm{vec}_k[p(X, Y)]$, where k denotes that the vector will only store the coefficients of the corresponding operators up to order k (so a vector of length $2^{k+1} - 1$) and $p(X, Y)$ is the polynomial in terms of X and Y.

Tree product between coefficient arrays. We can define a product between these arrays that corresponds to the (noncommutative) product of polynomials in X and Y. Let p and q be such polynomials, and define the tree product $*$ as $\mathrm{vec}_k[p(X, Y)] * \mathrm{vec}_k[q(X, Y)] = \mathrm{vec}_k[p(X, Y) \cdot q(X, Y)]$, where "$\cdot$" is the usual operator product.

In order to obtain the coefficients w_j, we encode $\mathcal{T}_k[e^{t(X+Y)}]$ and $\mathcal{T}_k[S^{(m)}(t, w_1, \cdots, w_m)]$ from Eq. (6.26) and Eq. (6.28) using the encoding just described, disregarding the time t. Finally, the problem to be solved is as follows.

Problem 3 (Cost function error minimisation of product formula coefficients). Let X and Y be arbitrary non-commuting operators and let k be a natural number. Solve

$$\mathrm{argmin}_{w_1, w_2, \cdots, w_m} \left\| \mathrm{vec}_k \left[\mathcal{T}_k[e^{t(X+Y)}] \right] - \mathrm{vec}_k \left[\mathcal{T}_k[S^{(m)}(t, w_1, \cdots, w_m)] \right] \right\|^2. \tag{6.30}$$

To find a solution for a product formula of order k, the minimum should be zero, and the result for $w_1, w_2, \cdots, w_m$ will give the desired product formula. We choose to use the Yoshida ansatz as it requires a lower number of parameters in comparison to the most general ansatz of length m, and also naturally generalises to sums of more terms. Nevertheless, it is also possible to use our technique for more general non-symmetric product formulae.

Solution strategy. We first need to choose m to be large enough such that the minimum (not argmin) is equal to zero; i.e. it is possible to find an order k product formula of length m. To choose a value of m that successfully yields an 8th order product formula, we can follow Yoshida's ansatz: we know $m = 7$ works. To choose an m that yields a 10th order product formula, we extend the work of Yoshida to determine that $m = 15$ works. See Appendix B.1 for details.

Having chosen an appropriate value of m, we then can use a numerical nonlinear function solver. We tried various approaches and found that Matlab's `fsolve` was able to succeed if used as follows. Choose a random starting vector $\vec{w}$ and evaluate `fsolve` with the vector of errors. We find that generating the components of $\vec{w}$ according to the standard normal distribution works. We found that the best solutions were those with smaller values for the coefficients, so reduced the standard deviation for the initial $\vec{w}$ a little below 1. We tried standard deviations of 0.6 initially for 8th order, 0.9 for 10th order, and later tried a standard deviation of 1 for 8th order.

Matlab reports that `fsolve` uses the Levenberg-Marquardt algorithm, which interpolates between the Gauss-Newton algorithm and gradient descent. It uses information in the full vector of errors, so provides better performance than `fminsearch` with the sum of squares of errors. In comparison, Yoshida used the Brent method to solve the polynomial equations he derived [Yos90]. In more recent work, other authors have also used this approach of minimising the errors [BCCM13, AHKK17]. Although the coefficients obtained are not guaranteed to be formal solutions of the equations, the error minimisation is performed to very high accuracy.

That is, it is in principle possible that the numerical solution is giving a local minimum, and there is no true solution close to that vector. (The true solution is never given exactly because it would require an infinite number of digits.) In most cases the sum of the squares of errors was on the order of 10^{-27}, though some were larger, on the order of 10^{-22}. Solving further using extended precision (instead of double precision) resulted in the sum of squared errors being reduced by many more orders of magnitude. That is a strong indication that these are good approximations of true solutions, not local minima. Even if they were not, the error is so small that the resulting product formulae would give accurate approximations of exponentials useful for quantum simulation.

In real analysis, the Poincaré-Miranda theorem [Kul97] provides necessary and sufficient conditions to check whether there is a root of a set of non-linear equations in a hypercube. This theorem is a generalisation of the intermediate value theorem to multiple functions and variables. Unfortunately, this theorem requires the evaluation of the function at infinite points and does not provide a way to check that we have indeed reached a root of the system of equations. Nonetheless, evaluating the polynomials found through Yoshida's method on the vertices of the hypercube near the solutions found can help to discard those that are not near roots. We evaluated the polynomials and check that the conditions of the Poincaré-Miranda theorem are fulfilled for this vertices. We performed this evaluation on many of the solutions we found, and they passed the test for points at a distance of 10^{-10} from our solutions.

6.4 Improved 8th order product formulae

In this section we present the result of our numerics for 8th order product formulae. We have solved for product formulae both using our Taylor series procedure and the polynomial equations of Yoshida, and found nearly 600 product formulae of 8th order. In what follows we number our solutions according to the order in which we found them. To clarify the terminology, there is a distinction between the "cost function error" and the "product formula error". The cost function error refers to the minimised error

in Problem 3, on the other hand the product formula error refers to the error of the product formula when compared to the total evolution, for example in Eq. (6.1) the product formula error is given by the expression $O(t^2)$. Not only do we find many more solutions than Yoshida reports, but we also find product formulae with reduced product formula error. To show this we have compared the different product formula errors attained by the integrators under a range of Hamiltonians chosen randomly. The detail of how we compare product formulae is given in Section 6.6.

To be more precise, we have performed the search of product formulae by solving the optimisation in Problem 3 with $k = 8$ and $m = 7$. We have also numerically solved the polynomial equations of Yoshida for the search. Whenever we found certain parameters $w_1, \cdots, w_7$ giving a sum of squared errors below 10^{-20}, we considered these parameters as a potential product formula to be tested. The search now finds almost only repeated solutions and very few new solutions. This indicates that we have found nearly all solutions, but it is also possible that there are many more solutions with large values of $\vec{w}$. Indeed, the most recent new solutions we found have significantly larger values of $\vec{w}$. We find that large values of $\vec{w}$ correspond to worse product formulae with larger error. Therefore, even if there are many more solutions with large $\vec{w}$, they likely will not give improved performance over those we have already found. Once we have obtained the potential solutions, we generate random Hamiltonians and compute the product formula errors as a function of time. We show these errors in Fig. 6.1 for an example Hamiltonian of dimension $d = 4$, and 5 examples of product formulae. For 8th order product formulae we know that the product formula error is $O(t^9)$. We check the error scaling by picking two times t_1 and t_2 and computing errors δ_1 and δ_2 at these times, then we compute $\log(\delta_1/\delta_2)/\log(t_1/t_2)$ and check that it is close to 9. As the error for our product formulae is given by $\delta(t) = \chi t^9$ where χ is a constant factor, we can compute χ for each of them by considering $\chi = \delta/t^9$. For each product formula, we compute an average constant factor error; this average corresponds to the geometric mean of the constant factors computed for each random Hamiltonian. This method of comparing the performance of product formulae through the estimation of the constant factor in

the error has been used before (see for example [BCCM13]) and is considered a good approximation to the performance of the product formula in practice.

The two best performing 8th order product formulae as measured by the constant factor χ obtained are shown in Table 6.1. These have average constant factors $\chi = 5.8\times10^{-6}$ and $\chi = 9.4\times10^{-6}$. They were the 42nd and 100th solutions found, respectively, so were found relatively early out of the more than 600 solutions. These solutions are reported in extended precision; by using extended precision arithmetic we reduced the cost function error of solution 42 to 10^{-600}. For comparison, the worst performing product formula we found had a constant of $\chi = 2.7\times10^{5}$. We evaluated all the 8th order solutions of Yoshida, and found Solution D was best, with a constant $\chi = 9.7\times10^{-4}$.

	Best 8th order with $m = 7$	Second-best 8th order with $m = 7$
w_1	0.31529309239676659663205666381 1	0.37122062648117505118097053722986
w_2	0.33462491824529818378495797988218	0.40544709650967949690890447887218
w_3	0.29906418130365592384446354068 86	0.16633724441837318387261356221838
w_4	−0.57386247111608226665638772663554	−0.62219910114766848553693391042818
w_5	0.19075471029623837995387625645037	0.26406879487125261601060713402535
w_6	−0.40910082580003159399730009589356	−0.45453364433377659463237935329715
w_7	0.74167036435061295344822780178381	0.79748609972350707868528219873049

Table 6.1: Our two best-performing 8th order solutions with $m = 7$.

	Best 8th order with $m = 8$
w_1	0.29137384767986663096528500968049
w_2	0.26020394234904150277316667709864
w_3	0.18669648149540687549831902999911
w_4	−0.40049110428180105319963667975074
w_5	0.15982762208609923217390166127256
w_6	−0.38400573301491401473462588779099
w_7	0.56148845266356446893590729572808
w_8	0.12783360986284110837857554950443

Table 6.2: Our best-performing 8th order solution when setting $m = 8$.

We have also conducted a search for 8th order solutions with $m = 8$. Using $m = 8$ results in an underdetermined system of equations with continuous sets of solutions, and

	Best 8th order with $m = 10$
w_1	0.5935806040085031
w_2	−0.4691601234700394
w_3	0.2743566425898439
w_4	0.1719387948465702
w_5	0.2343987448254160
w_6	−0.4861642448032533
w_7	0.4961736738811380
w_8	−0.3266021894843879
w_9	0.2327167934936900
w_{10}	0.0982495574147107 5

Table 6.3: Our best-performing 8th order solution when setting $m = 10$.

gives the flexibility to adjust the solution to reduce the error. The best solution found is given in Table 6.2, with an average constant factor of $\chi = 5.7 \times 10^{-7}$, which is an order of magnitude improvement with only a slight increase in the number of exponentials. Moreover, it is more than a factor of 1000 times better than the best solution of Yoshida. After a literature search we find that the best solutions we have found for $m = 7$ and $m = 8$ correspond to solutions in [KL97]. When extending our search to $m = 10$ we find that the best solution is the one given in Table 6.3 with a constant of $\chi = 2.1 \times 10^{-8}$. In Section 6.6 we compare the solutions we have found and find that our solution with $m = 10$ is the best performing one.

There is a second way to compare the product formulae. Note that $\vec{v} = \text{vec}_k[\mathcal{T}_k[e^{t(A+B)}] - \mathcal{T}_k[S^{(m)}(t, w_1, \cdots, w_m)]]$ will be a vector with entries very close to zero for a product formula solution. We could also pick a larger k than the solution was derived for, for example $k = 9$ for an 8th order solution (so the k here is no longer the order). Then we consider the entries of the vector $\vec{v}$ that includes the 9th order operators. We have computed 9th, 10th, 11th and 12th order cost function errors. For each order we take the absolute value of this errors and sum them. We find that the lowest sum of absolute errors at each order is achieved by the best solution determined by the previous method (solution 42), and in fact the lowest sum of errors is strongly correlated with the average constant factor χ mentioned above.

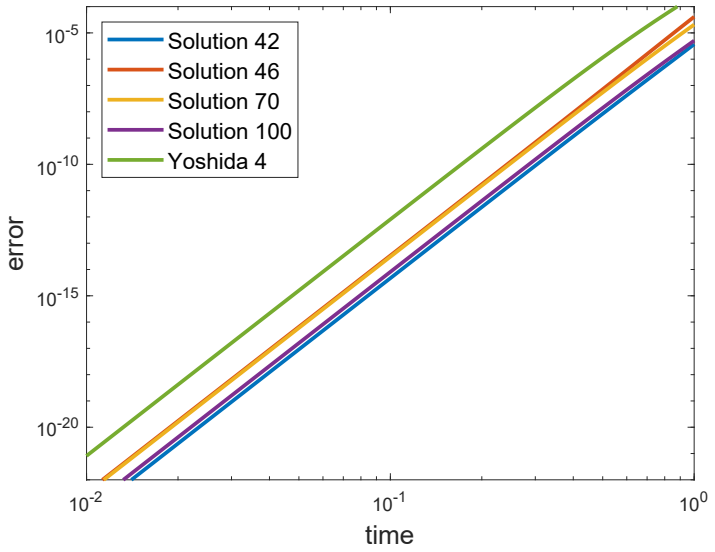

Figure 6.1: Error in product formula as determined by the spectral norm of the difference of operators as a function of t. We have shown our four best-performing product formulae for 8th order; these correspond to solutions 42, 46, 70, 100. For comparison we also show the best-performing solution of Yoshida, with errors an order of magnitude higher than the solutions we have obtained.

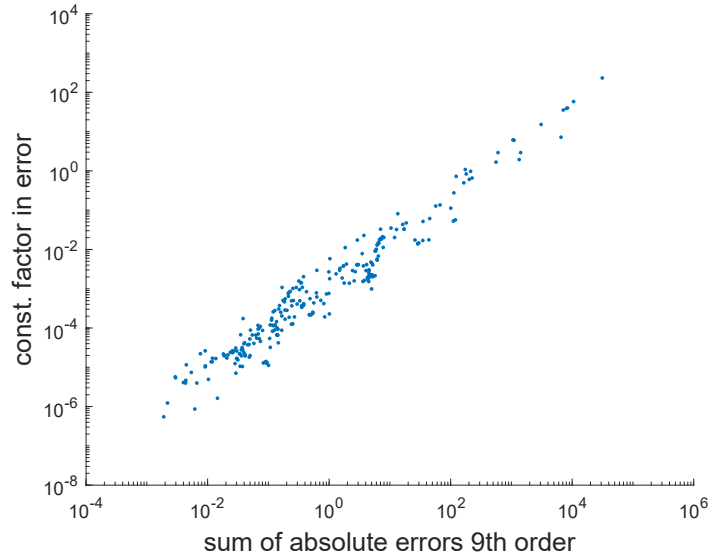

Figure 6.2: Plot of average constant factor in the error χt^9 for 100 random Hamiltonians and the sum of absolute 9th order errors defined in the main text. Each of the points represents one product formula obtained with our optimisation procedure.

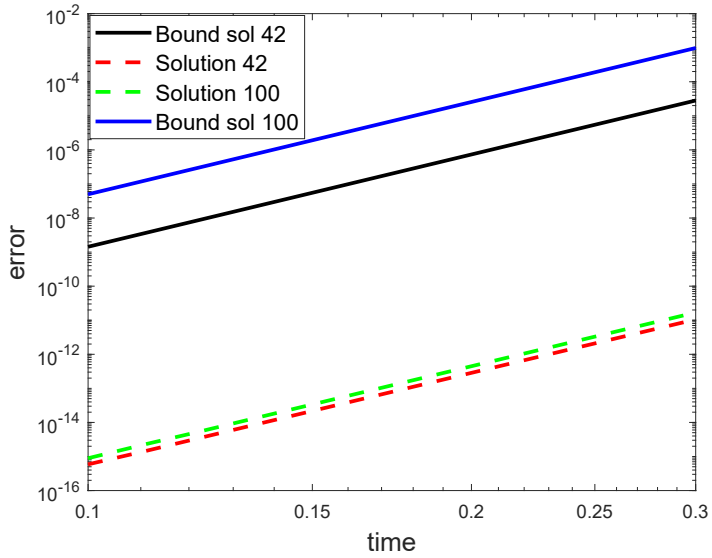

Figure 6.3: Error for two of our best solutions for an example Hamiltonian, compared with the error bound computed using a method based on that in [CST+21].

In Fig. 6.2 we plot the constant factor in the error versus the sum of absolute errors at 9th order. It can be seen that there is strong correlation, though the sum of absolute errors overestimates the error by a factor of 100 to 1000. The correlation coefficient between the sum of errors at order 9 and the constant factor is 0.977, at order 10 the correlation coefficient is 0.973, at order 11 it is 0.948, and at 12 it is 0.938.

We also have computed a bound on the error for 8th order integrators based on the work of [CST+21]. In Appendix M of that work, the authors derive a bound on the error for the Suzuki product formula of 4th order. We have extended the bounds to 8th order by generalisation their method to this case by implementing it in Mathematica. We have computed this bound for two of our best 8th order product formulae with $m = 7$ and show the results in Fig. 6.3. The bounds are very loose, being about a factor of 10^6 times larger than the actual error, though they follow the same trend.

We also have computed a bound on the error for 8th order integrators based on the work of [CST+21]. In Appendix M of that work, the authors derive a bound on the error for the Suzuki product formula of 4th order. We have extended the bounds to 8th order

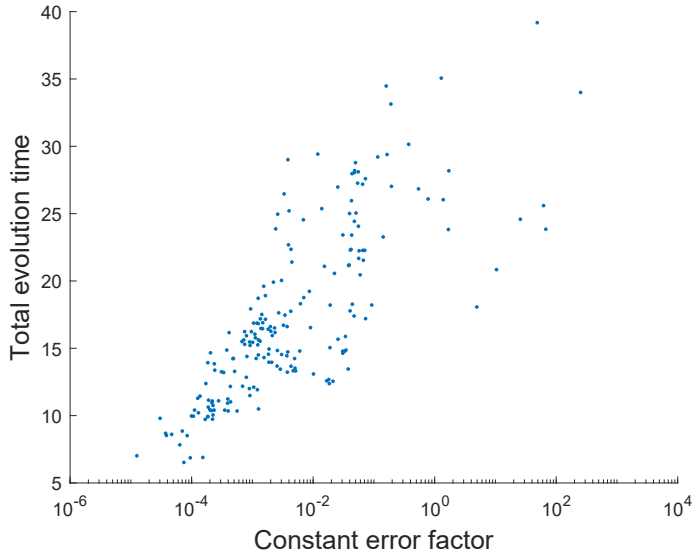

Figure 6.4: Comparison of the total evolution time $|w_0| + 2\sum_{j=1}^{m} |w_j|$ and the constant factor χ in the product formula error for the 8th order product formulae we have found with $m = 7$. Each point corresponds to a different product formula for which the constant error factor and total evolution time is computed.

by generalisation their method to this case by implementing it in Mathematica. We have computed this bound for two of our best 8th order product formulae with $m = 7$ and show the results in Fig. 6.3. The bounds are very loose, being about a factor of 10^6 times larger than the actual error, though they follow the same trend.

It can be shown that the total evolution time for a particular product formula can be used to bound the error in approximating the total evolution. For Yoshida's ansatz with parameters $(w_0, \cdots, w_m)$ and simulation evolution time t, the total evolution time is given by $t\left(|w_0| + 2\sum_{j=1}^{m} |w_j|\right)$. We give a derivation of the bound given by the total evolution time in Appendix B.2. We also compare the estimated constant factor of the product formula error with the expression $|w_0| + 2\sum_{j=1}^{m} |w_j|$ for the best 8th order product formulae we have found. We find a strong correlation between these two quantities, characterised by a correlation coefficient of 0.78. A plot showing this correlation is given in Fig. 6.4.

6.5 Finding 10th order product formulae

In this section we present the result of our numerics for 10th order product formulae.. We have generalised Yoshida's method, and find 15 independent equations to be solved. This gives the minimum m required to find new product formulae. This derivation is quite lengthy, so is given in Appendix B.1. We performed searches for solutions both with $m = 15$ (the minimal number) and $m = 16$. Again this gives the flexibility to adjust the solution to reduce the error. We report the best 10th order product formulae for $m = 15$ and 16 in Table 6.4 in extended precision.

As in Section 6.4, we compare the performance of product formulae of 10th order by computing the constant factor χ in the error χt^{11} for random Hamiltonians. For the best solution with $m = 16$ we have a constant factor of $\chi = 1.9 \times 10^{-8}$, and the best solution with $m = 15$ has $\chi = 9.4 \times 10^{-7}$, which is about a factor of 50 times worse. The far better constant factor for $m = 16$ is far more significant than the slightly larger number of exponentials in the product formula. For this reason we consider cases with $m = 16$ in the remainder of this discussion.

We compare our best-performing 10th order product formula to our best 8th order formula, and Yoshida's best, in Fig. 6.5. We also compare in Fig. 6.6 the best product formulae when the total Hamiltonian is given by a sum of ten terms. As explained in Section 6.2.2, the product formulae are also valid for Hamiltonians that are sums of arbitrary numbers of terms. This plot demonstrates that the correct scalings are still obtained with larger numbers of terms.

As in the 8th order case, we compute the total evolution time and compare with the constant factor error χ for a set of 10th product formulae found by the optimisation procedure. The result is shown in Fig. 6.7. The correlation factor is smaller than in the 8th order case, but still shows a relationship between these two quantities with a correlation factor of 0.5.

In the search for 10th order product formulae, unlike in the case of 8th order, we find that almost all new solutions found are different from those found before. That indicates

	Best 10th order sol. with $m = 15$	Best 10th order sol. with $m = 16$
w_1	0.14552859955499429739088135596618	−0.49450131799555718563471479776 44
w_2	−0.48773512068133537309419933740564	0.29043172229701214798784142920 93
w_3	0.12762011242429535909727342301656	0.34781541068705330937913890281003
w_4	0.70225450019485751220143080587959	−0.98828132118546184603769781410676
w_5	−0.62035679146761710925756521405042	0.98855187532756405235733957305613
w_6	0.39099152412786178133688869373114	−0.34622976933123177430694714630668
w_7	0.17860253604355465807791041367045	0.20218952619073117554714280367018
w_8	−0.80455783177921776295588528272593	0.13064273069786247787208895471461
w_9	0.05308721644275824211868738564628 3	−0.26441199183146805554735845490359
w_{10}	0.86836307910275556258687030904753	0.06099914055921040886909699229153 1
w_{11}	−0.85326297197907834671536254437991	−0.68554424896061413591089732670 28
w_{12}	−0.11732457198874083224967699358383	−0.15843692473786584550599206557006
w_{13}	0.03827345494186056632406947772047	0.15414691779958299150286452215575
w_{14}	0.74843529029532498233997793305357	0.66715205827214320371061839297055
w_{15}	0.30208715621975773712410948025906	0.20411874474696598289603677693511
w_{16}	NA	0.08120731821027259322508771144168 4

Table 6.4: Our best performing solutions for 10th order with $m = 15$ and $m = 16$.

that there is an extremely large number of solutions, and we have only found a very small proportion of them. Potentially there are solutions with even better performance still to be found.

6.6 Comparison of product formulae

In this section we provide a comparison of the product formulae we have found and those found previously in the literature. To compare the product formulae we've obtained, we make the following considerations. A k order integrator for time t will have an error $\delta = \chi t^{k+1}$ where χ is a real constant. Let T be the total evolution time for an integrator of order k, and ε be the maximum allowable error. Subdivide the evolution time T into r subintervals, so $t = T/r$ is the length of each time subinterval. We thus have $\chi\,(T/r)^{k+1} \approx \varepsilon/r$, which gives

$$r \approx \left(\frac{\chi T}{\epsilon}\right)^{1/k} T. \tag{6.31}$$

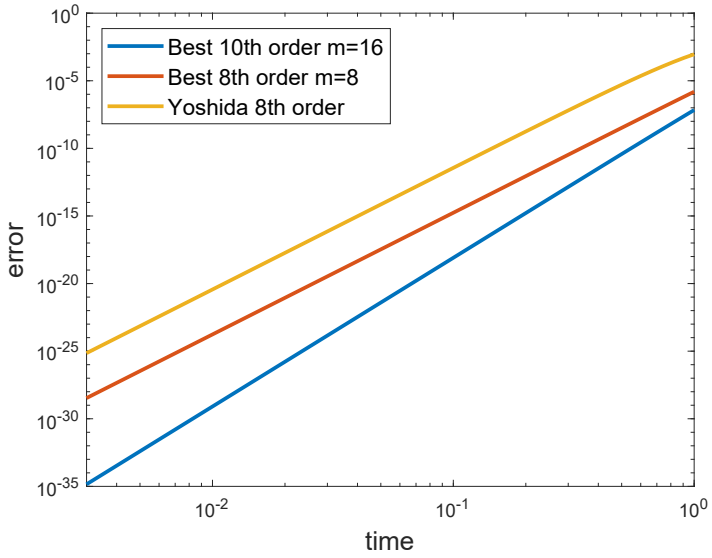

Figure 6.5: Error of the best 8th and 10th product formula obtained by our optimisation procedure together with the best 8th order product formula from Yoshida. To compute the error, two pairs of random Hamiltonians A and B were generated and the error was evaluated comparing to the total evolution $e^{-it(A+B)}$.

As explained above, the number of exponentials in the product is $(4m+2)(J-1)+1$. When applying products of these product formulae, two exponentials can be combined, so the effective number for each is $(4m+2)(J-1)$. As a result, the total number of exponentials can be given as proportional to

$$(2m+1)\left(\frac{\chi T}{\epsilon}\right)^{1/k} T \tag{6.32}$$

where we have ignored a common factor of $2(J-1)$.

If we wish to compare product formulae of the same order, then we need only compare the values of $(2m+1)\chi^{1/k}$, and the one with the smaller value is the more efficient product formula. This clearly demonstrates that our best product formula with $m=16$ is better than our best with $m=15$. We also evaluated product formulae derived using Suzuki's fractal method via a number of approaches, such as constructing 10th order from the best 8th order, or simply iterating the fractal method. Although some of these gave better χ values, these were more than outweighed by the significantly

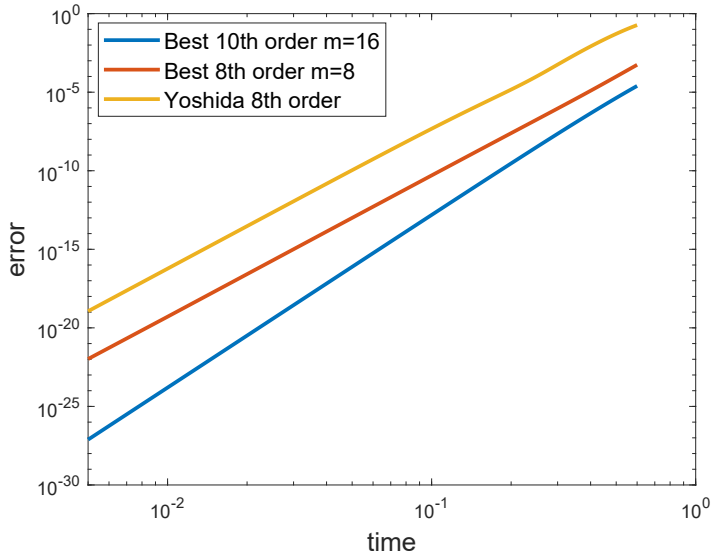

Figure 6.6: Error in the case of the total Hamiltonian decomposed into 10 terms. We compare the best 8th and 10th product formulae obtained by our optimisation procedure together with the best 8th order product formula from Yoshida. We generate a random tuple of Hamiltonian terms $(H_1, \cdots, H_{10})$ and compute the error comparing the product formula with the total evolution $e^{-it\sum_j H_j}$.

larger values of m, meaning that they were not as efficient as our best product formulae (both for 8th and 10th order). If we wish to compare product formulae of *different* order, then we need to take account of the values of T and ε. Assume we have two integrators of order k_1 and k_2, with corresponding constants χ_1, χ_2. Given T and ε, when the two integrators use the same number of exponentials we have $(2m_1 + 1)r_1 = (2m_2 + 1)r_2$, thus

$$(2m_1 + 1)\left(\frac{\chi_1 T}{\varepsilon}\right)^{1/k_1} T = (2m_2 + 1)\left(\frac{\chi_2 T}{\varepsilon}\right)^{1/k_2} T \tag{6.33}$$

$$\implies \frac{T}{\varepsilon} = \left(\frac{(2m_2 + 1)\chi_2^{1/k_2}}{(2m_1 + 1)\chi_1^{1/k_1}}\right)^{\frac{1}{\frac{1}{k_1} - \frac{1}{k_2}}} . \tag{6.34}$$

For $k_2 > k_1$, this gives the threshold beyond which we should use the higher-order product formula.

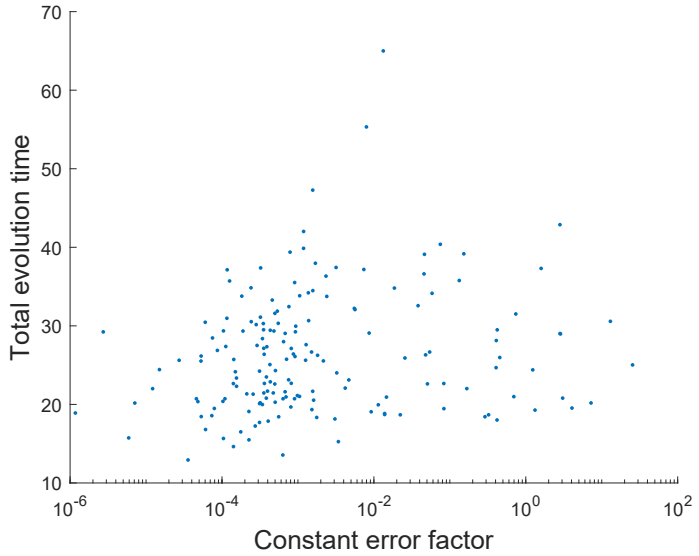

Figure 6.7: Comparison of the total evolution time $|w_0| + 2\sum_{i=1}^{m} |w_i|$ and the constant factor χ in the product formula error for the 10th order product formulae with $m = 16$ we have found. Each point corresponds to a different product formula for which the constant error factor and total evolution time is computed.

As mentioned before, we report comparisons between several 8th and 10th order product formulae as our numerics show that this will be most relevant in quantum simulation. From the literature, we consider the following 6th order:

- Y6m3a: product formula with $m = 3$ (7 stages). Reported in Table 1 of [Yos90] as Solution A.
- KL6s9a: product formula with 9 stages. Reported as s9odr6a in Appendix A of [KL97].
- KL6s9b: product formula with 9 stages. Reported as s9odr6b in Appendix A of [KL97].
- SS6s11: product formula with 11 stages. Reported in Section 4.2 of [SS05].
- SS6s13: product formula with 13 stages. Reported in Section 4.2 of [SS05].

We consider the following 8th order product formulae:

- KL8s15: product formula with 15 stages. Reported as s15odr8 in Appendix A of [KL97]. Corresponds to best solution found in Table 6.1 with $m = 7$.

- KL8s17: product formula with 17 stages. Reported as s17odr8 in Appendix A of [KL97]. Corresponds to best solution found in Table 6.2 with $m = 8$.
- SS8s19: product formula with 19 stages. Reported in Section 4.3 of [SS05].
- SS8s21: product formula with 21 stages. Reported in Section 4.3 of [SS05].
- PP8s13: post processing product formulae with 13 stages without considering processing. Reported as $P_{13}8$ in Table 6 of [BCM06].
- Y8m10: product formulae based on Yoshida's method with $m = 10$. Table 6.3.

For 10th order product formulae, we consider the following:

- KL10s31a: product formulae with 31 stages. Reported as s31odr10a in Appendix A of [KL97].
- KL10s31b: product formulae with 31 stages. Reported as s31odr10b in Appendix A of [KL97].
- SS10s31: product formulae with 31 stages. Reported in Section 4.4 of [SS05].
- SS10s33: product formulae with 33 stages. Reported in Section 4.4 of [SS05].
- SS10s35: product formulae with 35 stages. Reported in Section 4.4 of [SS05].
- PP10s23: post processing product formulae with 23 stages without considering processing. Reported as $P_{23}10$ in Table 6 of [BCM06].
- Y10m15: 10th order product formula based on Yoshida's method with $m = 15$. Table 6.4.
- Y10m16: 10th order product formula based on Yoshida's method with $m = 16$. Table 6.4.

We provide the constant factors in the error χ for 6th order in Table 6.5, in Table 6.6 we give the constant factors for the 8th order product formulae and we give the constant factors for 10th order in Table 6.7.

To compare among product formulae of the same order, we use Eq. (6.34) and set $k_1 = k_2$. Then, we need to compute the ratio $\frac{(2m_2+1)\chi_2}{(2m_1+1)\chi_1}$. Note that the m_i refer to the m in Yoshida's method but we can instead write the ratio in terms of number of S_2 operators as $\frac{s_2\chi_2}{s_1\chi_1}$ where s_i corresponds to the number of S_2 in the product formula. By comparing this ratio among the 6th order product formulae, we find that the best performing one is

	χ
KL6s9a	6.8×10^{-4}
KL6s9b	6.7×10^{-4}
SS6s11	1.3×10^{-4}
SS6s13	7×10^{-5}
Y6m3a	1.6×10^{-3}

Table 6.5: Constant factor in the error for 6th order product formulae.

	χ
KL8s15	5.8×10^{-6}
KL8s17	5.7×10^{-7}
SS8s19	6.2×10^{-8}
SS8s21	1.6×10^{-7}
PP8s13	2.7
Y8m10	2.1×10^{-8}

Table 6.6: Constant factor in the error for 8th order product formulae.

	χ
KL10s31a	5.8×10^{-6}
KL10s31b	4.7×10^{-5}
SS10s31	4.3×10^{-7}
SS10s33	8.7×10^{-9}
SS10s35	6.3×10^{-9}
PP10s23	0.23
Y10m15	9.4×10^{-7}
Y10m16	1.9×10^{-8}

Table 6.7: Constant factor in the error for 10th order product formulae.

SS6s13, which despite using more stages, the threshold is very small when comparing with other 6th order formulae. In the 8th order case, the best performing product formulae is Y8m10 (which we have reported in this paper) and in the 10th order case SS10s31, SS10s33 and SS10s35 are the best performing depending on the threshold which is rather low. For example the threshold between SS10s35 SS10s31 is around 60.4 which indicates that SS10s35 would be preferred for most practical applications of quantum simulation.

The threshold T/ϵ for 8th order to improve over 6th order is only about 0.08 when comparing the best 6th and 8th order product formula. The value of χ for the 4th order product formula (using Suzuki's method with 5 copies of S_2) is about 2.5×10^{-3}. As a result, the threshold for 6th order to improve over 4th order is about 9.4×10^3, which is well above that for 8th order to beat 6th order. This means that, as T/ϵ is increased, one should change over directly from 4th order to 8th order, and not use 6th order. The threshold for using 8th order instead of 4th order is about 1.6×10^3.

To compare our best 8th order and 10th order product formulae, we took $m_1 = 8, m_2 = 16, k_1 = 8, k_2 = 10$ in Eq. (6.34). When comparing SS8s19 and SS10s35, the threshold for shifting to 8th to 10th order is 7×10^{13} which is very high for practical applications.

This threshold calculation is for Hamiltonians that are sums of two terms, each of which is normalised. In the case of quantum chemistry, the Hamiltonians have far more structure in them. In particular, fermionic Hamiltonians encountered in quantum chemistry often have the form

$$\sum_{p,q}^{d} \tau_{pq} a_p^\dagger a_q + \sum_{p,q}^{d} v_{pq} a_p^\dagger a_p a_q^\dagger a_q \tag{6.35}$$

where $a_p^\dagger$ and a_p are the fermionic creation and destruction operators acting on orbital j and there are a total of d orbitals. Each entry τ_{pq}, v_{pq} is real and there is symmetry in exchanging indices. We compute χ for our best 8th order, best 10th order and the best 8th order from Yoshida. The behaviour of χ as the size of system is changed can be predicted based on the result in Theorem 4 of [LSTT22].

Theorem 6.5 (Theorem 4 in [LSTT22]). *Let* $H = T + V = \sum_{p,q}^{d} \tau_{pq} a_p^\dagger a_q + \sum_{p,q}^{d} v_{pq} a_p^\dagger a_p a_q^\dagger a_q$ *be an interacting-electronic Hamiltonian,* $S_k(t)$ *be a kth order product formula splitting the evolutions under* T *and* V*. Then*

$$\left\| S_k(t) - e^{-itH} \right\|_{W_\eta} = O\left((\|\tau\|_1 + \|v\|_{1,[\eta]})^{k-1} \|\tau\|_1 \|v\|_{1,[\eta]} \eta t^{k+1} \right). \tag{6.36}$$

where $\|\cdot\|_{W_\eta}$ *corresponds to the operator norm on the operator acting in the* η*-electron subspace,* $\|\tau\|_1 = \max_p \sum_q |\tau_{pq}|$ *and* $\|v\|_{1,[\eta]} = \max_p \max_{q_1 < q_1 < \ldots < q_\eta} (|v_{pq_1}| + \cdots + |v_{pq_\eta}|)$.

With the errors computed numerically, we can divide χ by the expression on the right of Eq. (6.36). In Table 6.8 we give the computed result for $d = 4, 6, 8$ orbitals, assuming half-filling of the orbitals. To compute χ, we generate 100 random instances of Hamiltonians and then compute the geometric mean over the χ obtained. Our numerics indicate that the error is roughly proportional to the bound in Eq. (6.36), independent of the number of orbitals, though the constant factors are small. Moreover, out product formulae still performs much better than Yoshida's, with about a factor of 1000 less error.

Num. orbitals	8th order ($m = 8$)	10th order ($m = 16$)	Yoshida 8th order
4	5.9×10^{-10}	2.7×10^{-12}	7.1×10^{-7}
6	5.8×10^{-10}	2.8×10^{-12}	6.4×10^{-7}
8	5.4×10^{-10}	2.6×10^{-12}	5.4×10^{-7}

Table 6.8: Comparison of constant factor χ for our best product formulae and the best 8th order product formula by Yoshida. We generate 100 random Hamiltonians as in Eq. (6.35) and compute the average χ.

To estimate thresholds for quantum chemistry, we can define ξ so that

$$\chi = \xi \left(\|\tau\|_1 + \|v\|_{1,[\eta]}\right)^{k-1} \|\tau\|_1 \|v\|_{1,[\eta]} \eta. \tag{6.37}$$

Then the formula for the threshold T/ε becomes

$$\begin{aligned}
\frac{T}{\varepsilon} &= \left(\frac{(2m_2+1)[\xi_2(\|\tau\|_1 + \|v\|_{1,[\eta]})^{k_2-1}\|\tau\|_1\|v\|_{1,[\eta]}\eta]^{1/k_2}}{(2m_1+1)[\xi_1(\|\tau\|_1 + \|v\|_{1,[\eta]})^{k_1-1}\|\tau\|_1\|v\|_{1,[\eta]}\eta]^{1/k_1}} \right)^{\frac{1}{\frac{1}{k_1}-\frac{1}{k_2}}} \\
&= \left(\frac{(2m_2+1)[\xi_2\|\tau\|_1\|v\|_{1,[\eta]}\eta/(\|\tau\|_1 + \|v\|_{1,[\eta]})]^{1/k_2}}{(2m_1+1)[\xi_1\|\tau\|_1\|v\|_{1,[\eta]}\eta/(\|\tau\|_1 + \|v\|_{1,[\eta]})]^{1/k_1}} \right)^{\frac{1}{\frac{1}{k_1}-\frac{1}{k_2}}} \\
\frac{\|\tau\|_1\|v\|_{1,[\eta]}\eta}{\|\tau\|_1 + \|v\|_{1,[\eta]}} \frac{T}{\varepsilon} &= \left(\frac{(2m_2+1)\xi_2^{1/k_2}}{(2m_1+1)\xi_1^{1/k_1}} \right)^{\frac{1}{\frac{1}{k_1}-\frac{1}{k_2}}}.
\end{aligned} \tag{6.38}$$

Thus we see that the ratio $\|\tau\|_1\|v\|_{1,[\eta]}\eta/(\|\tau\|_1 + \|v\|_{1,[\eta]})$ governs the threshold where a 10th-order product formula will improve over an 8th-order product formula.

With the values of ξ_1 and ξ_2 above we have the right-hand-side of (6.38) approximately equal to 3×10^{11}. From Ref. [LSTT22] the norms can be expected to scale

as

$$\|v\|_{1,[\eta]} = O\left(\frac{\eta^{2/3}N^{1/3}}{\Omega^{1/3}}\right), \qquad \|\tau\|_1 = O\left(\frac{N^{2/3}}{\Omega^{2/3}}\right), \tag{6.39}$$

where N is the number of orbitals and Ω is the volume (denoted n and ω in [LSTT22]). For some order of magnitude estimates, the chemical accuracy required for the phase estimation is about 0.001 Hartree, which implies T of about 1000π, and ε can be taken to be of order 1. With high estimates $\eta \approx 100$ and $N/\Omega \approx 10^9$, the left-hand-side of (6.38) would still only be on the order of 10^8, about three orders of magnitude less than the threshold. Therefore the threshold to use 10th order instead of 8th order is well beyond the expected values needed for simulations in quantum chemistry.

6.7 Conclusion

We have extended the method of Yoshida to construct product formulae of 10th order with a minimum number of factors (for symmetric product formulae constructed from S_2). We have also constructed 10th order product formulae with more factors that are far more accurate. Yoshida only found five 8th order solutions, but we have found over 600 with a minimum number of factors. We have also found hundreds of examples with more factors that are again far more accurate. Our best 8th order product formula is more than a factor of 1000 times more accurate than the best product formula of Yoshida.

We have provided a method of fairly comparing product formulae with different numbers of terms and different orders. This demonstrates that our best solutions for 8th and 10th order also improve over those obtained using Suzuki's fractal method with lower error but many more terms. For comparing our best 10th order to our best 8th order, simulations with $T/\epsilon \gtrsim 6 \times 10^{11}$ would be required. This is larger than would be expected for most applications, indicating that our best 8th order solution would be best for most simulations.

The analysis of the threshold of T/ε depends on the form of the Hamiltonian. To more specifically analyse the performance for Hamiltonians for quantum chemistry, we

have considered fermionic Hamiltonians, together with the result for the error given in [LSTT22]. The results are qualitatively similar to those obtained simply by using random Hamiltonians. The threshold for there to be an improvement in performance by using the 10th order product formula is still orders of magnitude larger than would be expected for typical ranges of parameters.

In further work one could increase the number of factors in the product formulae in order to further reduce the error. There were large reductions in the error just from increasing m by 1 from its minimal value. There is also the possibility to extend the solutions to even higher-order product formulae, though those would likely need much higher numbers of terms, and so the threshold for them to provide an advantage would be even higher.

Chapter 7

Conclusion

In this thesis we have explored three topics whose unifying motivation is that of understanding tractability or intractability of physically relevant problems for quantum computers. We began in Chapter 3 and Chapter 4 by studying the intractability of the Weighted Local Hamiltonian Problem. This likely intractability is tied to the validity of the quantum exponential time hypothesis (QETH). As remarked in Chapter 4 it remains open whether this problem is complete for the class QW[1]. An important future direction is that of determining if this problem is in fact complete and whether there are other important problems which are complete for this class. Moreover it would be interesting to develop a quantum version of the Sparsification Lemma as stated in Lemma 2.13.

Chapter 5 introduced a sampling scheme based on fermionic particles. We have shown that such scheme has comparable hardness guarantees as Boson Sampling and Random Circuit Sampling. Possible future work includes reducing the depth at which anticoncentration holds for random free Fermionic circuits. Some numerical work in the original paper [ODMZ22] shows that these circuits could anticoncentrate at logarithmic depth. This has already been shown for Random Circuit Sampling in [DHJBa22], the techniques used to prove this could be adapted for free Fermionic circuits and prove a similar result.

Finally, in Chapter 6 we give new 8th and 10th order product formulae. We show that these improve over previous formulae in the literature by numerically testing their performance. The numerical recipe we have established could be extended to the randomized Trotter version [Cam19] or to multiproduct formulae [ZRB23].

As discussed in Chapter 1, there were three motivating questions for the work presented in this thesis. The first question, regarding the limitations of quantum computers and their computational capabilities, has been studied in the setting of parameterized complexity theory has been studied in Chapter 3 and Chapter 4. We hope this work to be the first steps in fully developing a quantum parameterized complexity theory. The second question, on the conditions in which quantum algorithms can exhibit superiority over classical counterparts, has been studied in Chapter 5. We found that Fermionic Linear Optics provide a setting in which a robust quantum advantage can be found. Whether this advantage can be transferred to other problems such as in quantum chemistry is an intriguing possibility. Finally, we have studied better implementation of product formulae for quantum simulation. We expect these to be useful in quantum chemistry and also inspire more work in improving the product formulae technique.

Appendix A

Some proofs of results for Part II

These proofs have been included in this appendix so the main text is easier to follow.

A.1 #P-Hardness of probabilities in shallow depth active FLO circuits

We argued in section 5.5 that amplitudes of active FLO circuits are #P-hard to compute. Here we show that similarly strong simulation (i.e., computing output probabilities) of constant-depth active FLO circuits is hard. It has been proven in previous work [BJS11] that under certain conditions, non-universal circuit families of shallow depth are hard to simulate under plausible conjectures which in addition implies that the output probabilities are #P-hard. In concrete, it is required that the postselected version of the circuit family is universal for quantum computation. This method is not robust as it only shows that exactly computing the output probabilities are hard, nonetheless it may be of interest that such hardness results can be obtained for constant-depth active FLO circuits. The required theorem is as follows

Theorem A.1. *Let $\mathcal{F}$ be a restricted family of quantum circuits. If circuits from $\mathcal{F}$ with the added power of postselection can simulate the output probability distributions of*

universal quantum circuits with postselection (i.e., $\mathcal{F}$ is universal with postselection) then computing the output probabilities (strong simulation) of circuits in $\mathcal{F}$ is #P-hard.

Proof. Similar results have been proven in [AA11, BJS11] and later in other works related to active FLO [HJKS20]. Let C be some circuit with gates from a universal gate set and let $P_C(\mathbf{y})$ be the output probability of result $\mathbf{y}$. By hypothesis, with the power of postselection we can use a circuit F from $\mathcal{F}$ to simulate C and thus $P_C(\mathbf{y}) = P_F(\mathbf{y}_*|00\cdots0) = \frac{P_F(\mathbf{y}_*00\cdots0)}{P_F(00\cdots0)}$, where $\mathbf{y}_*$ is potentially a bitstring encoding $\mathbf{y}$ (which will be our case below). This directly implies that if we could compute the output probabilities of F then this would allow for computing the output probabilities of C. Since universal circuits are known to include #P-hard instances, the result follows. ■

In what follows, we will always assume that the active FLO circuits are supplied with auxiliary states $|\Psi_4\rangle$. Throughout this section we will consider the encoding $|0_L\rangle = |00\rangle$ and $|1_L\rangle = |11\rangle$. To prove that computing the probabilities of shallow depth active FLO circuits is #P-hard, we prove now Lemma A.2.

Lemma A.2. *Constant-depth active FLO circuits supplied with auxiliary states $|\Psi_4\rangle$ with the added power of postselection are universal.*

To prove this, we follow Ref. [Bro15], which showed similar results in the context of Boson Sampling. The starting point is the brickwork graph state which allows for universal computation on the measurement based quantum computation (MBQC) scheme. We can write the preparation of the brickwork graph state plus measurements on the state as a single circuit with adaptive measurements. If we are given the power to postselect measurements, then the preparation of the graph state requires a constant depth circuit with single qubit gates and CZ gates. If we can simulate these gates with constant-depth active FLO circuits and postselection, then this would imply Lemma A.2. Using the encoding defined above, we show Theorem A.3 which directly implies Lemma A.2.

Theorem A.3. *Active FLO acting on an initial state consisting of tensor products of* $|\Psi_4\rangle$ *with the added power of postselection can simulate single qubit gates and* CZ *with constant-depth circuits. These simulations are at the logical level using the encoding above.*

Proof. As explained before, the circuit induced by the brickwork state with post selection is universal and of constant depth, consisting of single qubit gates and CZ gates. Using the encoding above we can simulate single qubit gates and CZ gates in constant depth, then we can simulate the whole universal constant-depth circuit with a circuit from C_{act} and postselection.

That single qubit gates at the logical level can be implemented with this encoding is already known [BK02]. Implementing CZ at the logical level will require the use of post selection and the auxiliary states $|\Psi_4\rangle$. First, we note that the state $|\Psi_4\rangle$ can be transformed into the state $|a_8\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$ using only active FLO operations. Second, in Lemma 1 of [Bra06] it is shown that using a single copy of $|a_8\rangle$ and particle number measurements it is possible to implement a CZ at the logical level using the same encoding we use here. This two facts together imply that CZ can be implemented with active FLO circuits supplied by $|\Psi_4\rangle$ states and postselection. The auxiliary states can be swapped to the desired position when implementing a gate without incurring on extra negative signs with our encoding since the auxiliary states used are fermionic as for example argued in [HJK+19].

A.2 Bounding K_{pas} and K_{act}

In this section we prove that in the passive case $\frac{K_{\text{pas}}}{2N+1} \leq \frac{C_{\text{pas}}}{N}$ and in the active case $\frac{K_{\text{act}}}{2^{8N}} \leq \frac{C_{\text{act}}}{\sqrt{\pi N}}$, which is required for the proof of anticoncentration. To prove this, we will use a known bound for the binomial coefficients.

Lemma A.4 (Bounds for binomial and trinomial coefficients). *Let n, k be a natural numbers such that $k \in \{1, \dots, n-1\}$. Let $x = \frac{k}{n}$. Then we have*

$$c \cdot \sqrt{\frac{n}{k(n-k)}} \exp\left(n\, h(x)\right) \leq \binom{n}{k} \leq C \cdot \sqrt{\frac{n}{k(n-k)}} \exp\left(n\, h(x)\right) , \tag{A.1}$$

where $c = \frac{1}{2\sqrt{2}}$, $C = \frac{1}{\sqrt{2\pi}}$, and $h(x) = -x\log(x) - (1-x)\log(1-x)$ is the binary entropy.

Moreover, let k, l, m be nonzero natural numbers such that $k + l + m = n$. Let $x = \frac{k}{n}, y = \frac{l}{n}, z = \frac{m}{n}$. Then we have

$$a \sqrt{\frac{n}{k \cdot l \cdot m}} \exp\left(n\, h(x,y,z)\right) \leq \binom{n}{k,l,m} \leq A \sqrt{\frac{n}{k \cdot l \cdot m}} \exp\left(n\, h(x,y,z)\right) , \tag{A.2}$$

where $a = \frac{1}{8}$, $A = \frac{1}{2\pi}$ and $h(x,y,z) = -x\log(x) - y\log(y) - z\log(z)$ is the entropy of three-outcome probability distribution.

The inequality (A.1) can be found in Lemma 7 in Chapter 10 of [MS83] while (A.2) follows from it due to identity $\binom{n}{k,l,m} = \binom{n}{k}\binom{l+m}{m}$.

Lemma A.5. *With the notation from Proposition 5.5, we have that $\frac{K_{\text{pas}}}{2N+1} \leq \frac{C_{\text{pas}}}{N}$. Where $C_{\text{pas}} = 5.7$.*

Proof. We have that

$$\begin{aligned} \frac{K_{\text{pas}}}{2N+1} &= \frac{1}{2N+1}\left[2\sum_{k=0}^{N-1}\frac{1}{\binom{2N}{k}}\sum_{l=0}^{\lfloor k/2\rfloor}\frac{N!}{l!(k-2l)!(N-k+l)!} + \frac{1}{\binom{2N}{N}}\sum_{l=0}^{\lfloor N/2\rfloor}\frac{N!}{l!(N-2l)!l!}\right] \\ &= \frac{2}{2N+1}\sum_{k=0}^{N}\sum_{l=0}^{\lfloor k/2\rfloor}\frac{\binom{N}{l,k-2l,N-k+l}}{\binom{2N}{k}}. \end{aligned} \tag{A.3}$$

Let us denote

$$f_N(k,l) := \frac{\binom{N}{l,k-2l,N-k+l}}{\binom{2N}{k}} . \tag{A.4}$$

Then we have

$$\frac{K_{\text{pas}}}{2N+1} = \frac{2}{2N+1}\sum_{k=0}^{N}\sum_{l=0}^{\lfloor k/2\rfloor} f_N(k,l) \tag{A.5}$$

$$= \frac{1}{N}\left(\mathcal{A}_{k=0} + \mathcal{A}_{l=0} + \mathcal{A}_{k=2l} + \mathcal{A}_{gen}\right) \tag{A.6}$$

where

$$\mathcal{A}_{k=0} = f_N(0,0) = 1\,, \tag{A.7}$$

$$\mathcal{A}_{l=0} = \sum_{k=1}^{N} \frac{\binom{N}{k}}{\binom{2N}{k}}\,, \tag{A.8}$$

$$\mathcal{A}_{k=2l} = \sum_{\substack{k>1 \\ k \text{ even}}}^{N} \frac{\binom{N}{k/2}}{\binom{2N}{k}}\,, \tag{A.9}$$

$$\mathcal{A}_{gen} = \sum_{k=1}^{N} \sum_{l=1}^{l<k/2} f_N(k,l)\,. \tag{A.10}$$

We upper bound each term above separately (except for the trivial case of $\mathcal{A}_{k=0}$). The following analytical proof for the bound requires $N \geq 130$. In particular, the bound for $\mathcal{A}_{l=0}$ is valid for $N \geq 40$, and the bound for $\mathcal{A}_{gen}$ is valid for $N \geq 130$. At the end of the proof, we show in Fig. A.2 that the bound also holds for all smaller values of N.

Upper bound on $\mathcal{A}_{l=0}$. In this case, we derive a bound valid for $N > 40$. The bounds from Lemma A.4 gives

$$\mathcal{A}_{l=0} \leq \frac{1}{\binom{2N}{N}} + \frac{C}{c} \sum_{k=1}^{N-1} \sqrt{\frac{2N-k}{2(N-k)}} \exp\left[N\left\{h(k/N) - 2h(k/2N)\right\}\right]\,. \tag{A.11}$$

We use now the inequality $h(x) - 2h(x/2) \leq -\frac{2}{3}x$, valid for $x \in [0,1]$ to obtain

$$\mathcal{A}_{l=0} \leq \frac{1}{\binom{2N}{N}} + \frac{C}{c} \sum_{k=1}^{N-1} \sqrt{\frac{2N-k}{2(N-k)}} \exp\left(-\frac{2k}{3}\right)\,. \tag{A.12}$$

We then apply the bound $\binom{2N}{N} \geq c 2^{2N}\sqrt{2/N}$ and divide the sum over k into two parts

$$\mathcal{A}_{l=0} \leq \frac{\sqrt{N}}{\sqrt{2}c} 2^{-2N} + \frac{C}{c}\left(\sum_{k=1}^{k\leq 1/2N} \sqrt{\frac{2N-k}{2(N-k)}} \exp\left(-\frac{2k}{3}\right) + \sum_{k>1/2N}^{N-1} \sqrt{\frac{2N-k}{2(N-k)}} \exp\left(-\frac{2k}{3}\right)\right) \tag{A.13}$$

For $k \leq N/2$ we have $\sqrt{\frac{2N-k}{2(N-k)}} \leq \sqrt{3/2}$ and therefore

$$\mathcal{A}_{l=0} \leq \frac{\sqrt{N}}{\sqrt{2}c} 2^{-2N} + \frac{C}{c}\left(\sqrt{\frac{3}{2}}\frac{1}{e^{2/3}-1} + \frac{N^{3/2}}{2}\exp\left(-\frac{N}{3}\right)\right)\,, \tag{A.14}$$

where we have utilized the expression for the sum of geometric progression and the upper bound $\sqrt{\frac{2N-k}{2(N-k)}} \leq \sqrt{N}$, valid for $k \leq N-1$. Using expression (A.14) it is easy to verify that for $N > 40$ we have

$$\mathcal{A}_{l=0} \leq \frac{3}{2} . \tag{A.15}$$

Upper bound on $\mathcal{A}_{k=2l}$. Estimates for binomials from Lemma A.4 yield

$$\mathcal{A}_{k=2l} \leq \frac{C\sqrt{2}}{c} \sum_{\substack{k>1 \\ k \text{ even}}}^{N} \exp\left[-Nh(k/2N)\right] . \tag{A.16}$$

Concavity of binary entropy $h(\cdot)$ implies that for $x \in [0,1]$ we have $\frac{\log(2)}{2} x \leq h(x/2)$ and consequently

$$\mathcal{A}_{k=2l} \leq \frac{C\sqrt{2}}{c} \sum_{\substack{k>1 \\ k \text{ even}}}^{N} \exp\left(-\frac{k\log(2)}{2}\right) = \frac{C\sqrt{2}}{c} \sum_{p=1}^{\lfloor N/2 \rfloor} 2^{-p} . \tag{A.17}$$

The sum of the geometric series in the above expression is upper bounded by 1 and therefore

$$\mathcal{A}_{k=2l} \leq \frac{C\sqrt{2}}{c} \leq \frac{8}{5} . \tag{A.18}$$

Upper bound on $\mathcal{A}_{gen}$. In the following proof, we require that $N \geq 130$. For the generic points in the sum of Eq. (A.5) inequalities from Lemma A.4 give

$$\mathcal{A}_{gen} \leq \frac{A}{\sqrt{2}c} \sum_{k=1}^{N} \sum_{l=1}^{l<k/2} \sqrt{\frac{k(2N-k)}{l(k-2l)(N-k+l)}} \exp\left(N\left\{h\left[x_l, y_k - 2x_l, 1-y_k+x_l\right] - 2h\left[y_k/2\right]\right\}\right) , \tag{A.19}$$

where $x_l = l/N$, $y_k = k/N$. Note that $k = 1$ and $k = 2$ are implicitly excluded from the above sum because of the constraints on l and hence

$$\mathcal{A}_{gen} \leq \frac{A}{\sqrt{2}c} \sum_{k=3}^{N} \sum_{l=1}^{l<k/2} \sqrt{\frac{k(2N-k)}{l(k-2l)(N-k+l)}} \exp\left(N\left\{h\left[x_l, y_k - 2x_l, 1-y_k+x_l\right] - 2h\left[y_k/2\right]\right\}\right) . \tag{A.20}$$

In order to upper bound the expression we maximize the function

$$F(x,y) = h\left(x, y-2x, 1-y+x\right) - 2h\left(y/2\right) \tag{A.21}$$

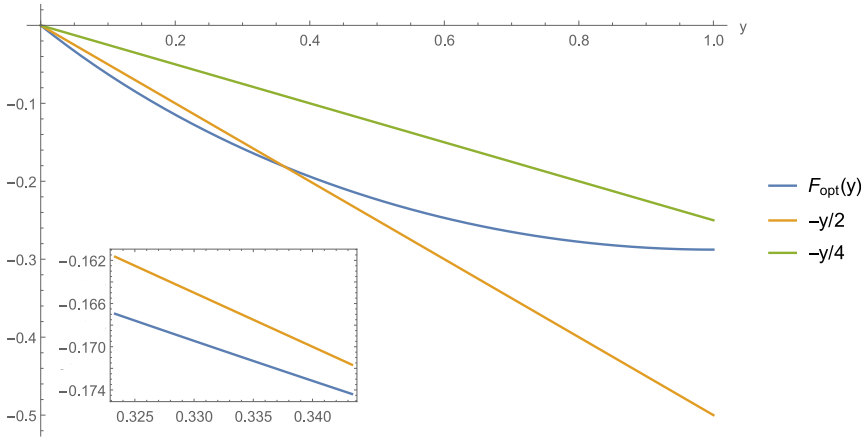

Figure A.1: Function $F_{opt}(y) := F(x_{opt}(y), y)$ where F is defined in Eq. (A.21) and $x_{opt}(y)$ is given in Eq. (A.22). The function is bounded by $-y/3$ in the interval $[0, 1/3]$ and by $-y/4$ in the interval $[1/3, 1]$. The inset plot shows that the inequality is also valid near $y = 1/3$. Figure from [ODMZ22].

over $x \in [0, y/2]$, for fixed value of $y \in [0, 1]$. Looking for critical points reduces the problem to solving quadratic equation which gives a unique solution in the interval $[0, y/2]$:

$$x_{opt}(y) = \frac{1}{6}\left(1 + 3y - \sqrt{1 + 6y - 3y^2}\right) . \tag{A.22}$$

Crucially, the function $F_{opt}(y) := F(x_{opt}(y), y)$ is a continuous function of parameter y, which is also analytic in the interior the interval $(0, 1)$. Moreover, $F_{opt}(y)$ satisfies (see Fig. A.1):

$$F_{opt}(y) \le -\frac{1}{2}y \ \text{ for } y \in [0, 1/3] \ , \ F_{opt}(y) \le -\frac{1}{4}y \ \text{ for } y \in [0, 1] \ . \tag{A.23}$$

It follows that

$$N\left(h\left[x_l, y_k - 2x_l, 1 - y_k + x_l\right] - 2h\left[y_k/2\right]\right) \le -\frac{1}{2}k \ \text{ for } \ 1 \le k \le N/3 \, , \tag{A.24}$$

$$N\left(h\left[x_l, y_k - 2x_l, 1 - y_k + x_l\right] - 2h\left[y_k/2\right]\right) \le -\frac{1}{4}k \ \text{ for } \ 1 \le k \le N. \tag{A.25}$$

Moreover, for integer l satisfying $1 \le l < k/2$ we have $l(k - 2l) \ge (k - 2)/2$ and consequently for $k \ge 3$ we have $\frac{k}{l(k-2l)} \le \frac{2k}{k-2} \le 6$. As a result we have

$$\sum_{l=1}^{l<k/2} \sqrt{\frac{k(2N-k)}{l(k-2l)(N-k+l)}} \le \frac{\sqrt{6}k}{2}\sqrt{\frac{2N-k}{N-k+1}} \, . \tag{A.26}$$

Inserting Eq. (A.24) and Eq. (A.26) into Eq. (A.20) gives

$$\mathcal{A}_{gen} \leq \frac{\sqrt{3}A}{2c}\left(\sum_{k=3}^{k\leq N/3}\sqrt{\frac{2N-k}{N-k+1}}\,k\,\exp\left(-\frac{k}{2}\right)+\sum_{k>N/3}^{N}\sqrt{\frac{2N-k}{N-k+1}}\,k\,\exp\left(-\frac{k}{4}\right)\right) \tag{A.27}$$

Observing that for $k \leq N/3$ we have $\sqrt{\frac{2N-k}{N-k+1}} \leq \sqrt{\frac{5}{2}}$, while and for general $k \leq N$ $\sqrt{\frac{2N-k}{N-k+1}} \leq \sqrt{N}$, we obtain

$$\mathcal{A}_{gen} \leq \frac{\sqrt{3}A}{2c}\left(\sqrt{\frac{5}{2}}\sum_{k=3}^{k\leq N/3}k\,\exp\left(-\frac{k}{2}\right)+\frac{2N^{\frac{3}{2}}}{3}\exp\left(-\frac{N}{12}\right)\right)\,. \tag{A.28}$$

We bound the first summand as follows

$$\sum_{k=3}^{k\leq N/3}k\,\exp\left(-\frac{k}{2}\right)\leq\sum_{k=3}^{\infty}k\,\exp\left(-\frac{k}{2}\right)=\frac{3\sqrt{e}-2}{(\sqrt{e}-1)^2e}\,. \tag{A.29}$$

This finally gives us

$$\mathcal{A}_{gen} \leq \frac{\sqrt{15}A}{2\sqrt{2}c}\frac{3\sqrt{e}-2}{(\sqrt{e}-1)^2e}+\frac{A}{\sqrt{3}c}N^{\frac{3}{2}}\exp\left(-\frac{N}{12}\right)\,. \tag{A.30}$$

Using the above expression we get that for $N \geq 130$ we have

$$\mathcal{A}_{gen} \leq \frac{8}{5}\,. \tag{A.31}$$

Finally, combining bounds (A.15), (A.18) and (A.31) together with $\mathcal{A}_{k=0} = 1$ we see that for $N \geq 130$,

$$\mathcal{A}_{k=0}+\mathcal{A}_{l=0}+\mathcal{A}_{k=2l}+\mathcal{A}_{gen} \leq 5.7\,. \tag{A.32}$$

Inserting this into Eq. (A.5) proves the lemma for $N \geq 130$. For $N \leq 130$, the validity of the bound can be verified numerically as shown in Fig. A.2, which completes the proof.

■

We prove the analogous result for active FLO. The proof follows a similar structure to the passive case.

Lemma A.6. *With the notation from Proposition 5.5, we have that* $\frac{K_{\text{act}}}{2^{8N}} \leq \frac{C_{\text{act}}}{\sqrt{\pi N}}$. *Where* $C_{\text{act}} = 16.2$.

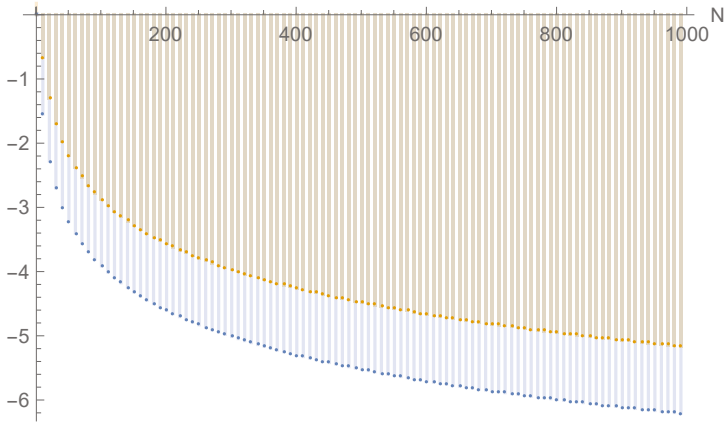

Figure A.2: Plots of the logarithm of the expression (A.5) (blue) and $\log\big(C_{\text{pas}}/N\big) = \log(5.7/N)$ (orange), which constitutes a valid upper bound for all $N \leq 1000$. Figure from [ODMZ22].

Proof. Analogously for the active FLO case, we have that

$$\frac{K_{\text{act}}}{2^{8N}} = \frac{1}{2^{8N}}\left[2\sum_{q}^{N-1} C_{2q} \sum_{l=0}^{\lfloor \frac{q}{2} \rfloor} \frac{N!}{l!(q-2l)!(N-q+l)!} 14^{q-2l} + C_{2N} \sum_{l=0}^{N} \frac{N!}{(l!)^2(4N-2l)!} 14^{N-2l}\right] \tag{A.33}$$

$$= \frac{\binom{8N}{4N}}{2^{8N-1}} \sum_{q=0}^{N} \sum_{l=0}^{\lfloor \frac{q}{2} \rfloor} \frac{\binom{4N}{2q}\binom{N}{l,q-2l,N-q+l}}{\binom{8N}{4q}} 14^{q-2l}, \tag{A.34}$$

where

$$C_{2q} = \frac{(4q)!(8N-4q)!}{((4N)!)^2}\binom{4N}{2q}. \tag{A.35}$$

Let us denote

$$g_N(q,l) := \frac{\binom{4N}{2q}\binom{N}{l,q-2l,N-q+l}}{\binom{8N}{4q}} 14^{q-2l}. \tag{A.36}$$

It follows from (A.33) and the entropic bound for binomial coefficients in Lemma A.4,

$$\frac{\binom{8N}{4N}}{2^{8N-1}} \leq \frac{1}{\sqrt{\pi N}}, \tag{A.37}$$

that

$$\frac{K_{\text{act}}}{2^{8N}} = \frac{1}{\sqrt{\pi N}} \sum_{q=0}^{N} \sum_{l=0}^{\lfloor \frac{q}{2} \rfloor} g_N(q,l) \leq \frac{1}{\sqrt{\pi N}}(\mathcal{B}_{q=0} + \mathcal{B}_{l=0} + \mathcal{B}_{q=2l} + \mathcal{B}_{gen}), \tag{A.38}$$

where

$$\mathcal{B}_{q=0} = g_N(0,0) = 1, \tag{A.39}$$

$$\mathcal{B}_{l=0} = \sum_{q=1}^{N} \frac{\binom{4N}{2q}\binom{N}{q}}{\binom{8N}{4q}} 14^q, \tag{A.40}$$

$$\mathcal{B}_{q=2l} = \sum_{\substack{q>1 \\ q \text{ even}}}^{N} \frac{\binom{4N}{2q}\binom{N}{q/2}}{\binom{8N}{4q}}, \tag{A.41}$$

$$\mathcal{B}_{gen} = \sum_{q=1}^{N} \sum_{l=1}^{l<q/2} g_N(q,l) \ . \tag{A.42}$$

We upper bound each term above separately (except for the trivial case of $\mathcal{B}_{q=0}$). The following analytical proof for the bound requires $N \geq 7000$. In particular, the bound for (A.15) $\mathcal{B}_{l=0}$ is valid for $N \geq 1000$, and the bound (A.31) for $\mathcal{B}_{gen}$ is valid for $N \geq 7000$. At the end of the proof, we show in Fig. A.4 that the bound also holds for all smaller values of $N \leq 8000$ by numerically evaluating right-hand side of (A.33).

Upper bound on $\mathcal{B}_{l=0}$. For this term, we require that $N \geq 1000$. The entropic bound in Lemma A.4 implies that

$$\mathcal{B}_{l=0} \leq \frac{\binom{4N}{2N}}{\binom{8N}{4N}} 14^N + \frac{C^2\sqrt{2}}{c} \sum_{q=1}^{N-1} \sqrt{\frac{N}{q(N-q)}} \exp[N\{h(q/N) - 4h(q/2N) + \log(14)q/N\}] \ . \tag{A.43}$$

To upper bound the sum, we split the sum into two sums: one from $q = 1$ to $q \leq N/5$ and another from $q > N/5$ to $q = N-1$, and upper bound the function

$$H(x) \coloneqq h(x) - 4h(x/2) + x\log(14), \tag{A.44}$$

$x \in [0,1]$ in the intervals $[0, 1/5]$ and $(1/5, 1]$ separately. In particular, we have that (See also Fig. A.3)

$$H(x) \leq -\frac{4}{3}x \ \text{ for } x \in [0, 2/5], \qquad H(x) \leq -\frac{1}{18}x \ \text{ for } x \in [0,1] \tag{A.45}$$

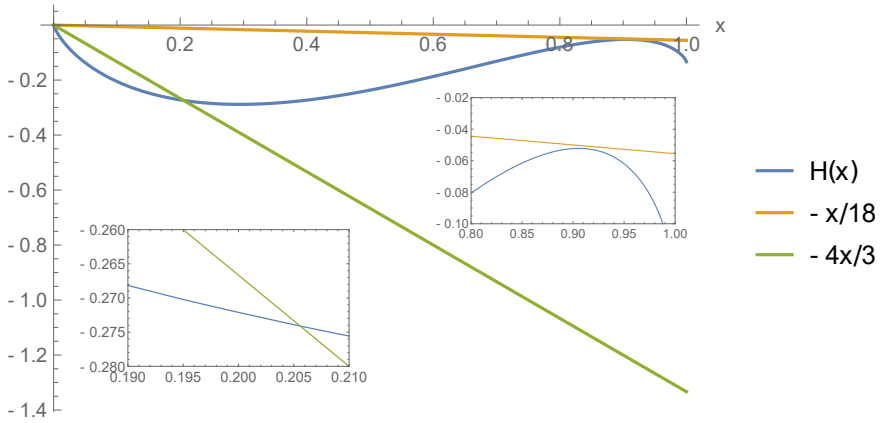

Figure A.3: Function $H(x)$ defined in (A.45). The function is bounded above by $-4x/3$ in the interval $[0, 1/5]$ and by $-x/18$ in the interval $[0, 1]$. The inset plot shows the validity of the upper bound in each interval. Figure from [ODMZ22].

Together with the bound $\binom{4N}{2N}/\binom{8N}{4N} \leq \frac{C\sqrt{2}}{c} 2^{-4N}$ and $\sqrt{N/(q(N-q))} \leq \sqrt{2}$ valid for $N \geq 2$ (this is because $\sqrt{\frac{N}{q(N-q)}}$ is convex for $q \in [1, N-1]$ and thus the expression takes the maximum values at the end points), we obtain

$$\mathcal{B}_{l=0} \leq \frac{C\sqrt{2}}{c}\left(\frac{14}{16}\right)^N + \frac{2C^2}{c}\left(\sum_{q=1}^{q\leq N/5} \exp(-4q/3) + \sum_{q>N/5}^{N-1} \exp(-q/18)\right) \tag{A.46}$$

$$\leq \frac{C\sqrt{2}}{c}\left(\frac{14}{16}\right)^N + \frac{2C^2}{c}\left(\frac{1}{e^{4/3}-1} + \frac{4N}{5}\exp\left[-\frac{N}{18\cdot 5}\right]\right), \tag{A.47}$$

where we have used the sum of the geometric series to arrive at the final expression. Using the expression (A.47), it can be verified that

$$\mathcal{B}_{l=0} \leq \frac{1}{3} \tag{A.48}$$

holds for $N \geq 1000$.

Upper bound on $\mathcal{B}_{q=2l}$. From Lemma A.4 we see that

$$\mathcal{B}_{q=2l} \leq \frac{C^2\sqrt{2}}{c} \sum_{\substack{q>1 \\ q \text{ even}}}^{N} \sqrt{\frac{N}{\frac{q}{2}(N-\frac{q}{2})}} \exp[-3Nh(q/2N)] \tag{A.49}$$

Now by concavity of $h(x)$ for $x \in [0, \frac{1}{2}]$ we have $\log(2)x/2 \leq h(x/2)$ for $x \in [0, 1]$. Then

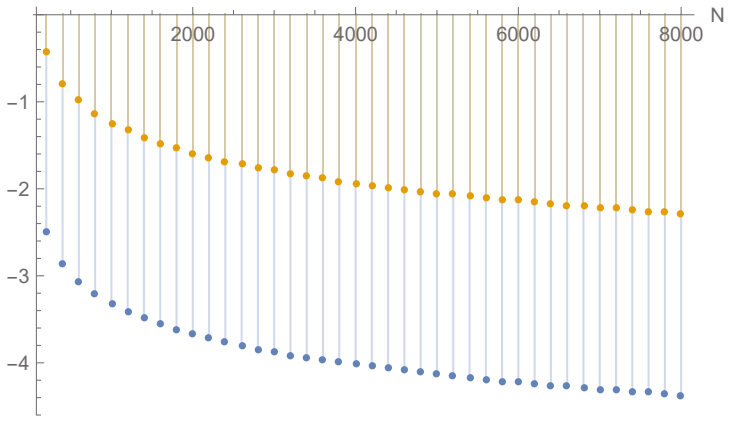

Figure A.4: Plots of the logarithm of the the expression (A.33) (blue) and $\log\left(C_{\text{act}}/\sqrt{N}\right) = \log\left(16.2/\sqrt{\pi N}\right)$ (orange), which is a valid upper bound for all $N \leq 8000$. Figure from [ODMZ22].

$$\mathcal{B}_{q=2l} \leq \frac{C^2\sqrt{2}}{c} \sum_{\substack{q>1 \\ q \text{ even}}}^{N} \sqrt{\frac{N}{\frac{q}{2}(N-\frac{q}{2})}} \exp[-3q\log(2)/2] \tag{A.50}$$

$$= \frac{C^2\sqrt{2}}{c} \sum_{p=1}^{\lfloor N/2 \rfloor} \sqrt{\frac{N}{p(N-p)}} 2^{-3p} \tag{A.51}$$

We can bound $\sqrt{\frac{N}{p(N-p)}} \leq \sqrt{2}$ the same way as in the passive case. Then we obtain

$$\mathcal{B}_{q=2l} \leq \frac{2C^2}{7c} \leq 0.13 \tag{A.52}$$

where we used that the geometric sum of 2^{-3p} is bounded by $1/7$.

■

Upper bound on $\mathcal{B}_{gen}$. Following bounds form Lemma A.4 and defining $x_l = \frac{l}{N}$ and $y_q = \frac{q}{N}$ we obtain

$$\mathcal{B}_{gen} \leq \frac{\sqrt{2}CA}{c} \sum_{q=1}^{N} \sum_{l=1}^{l<q/2} \sqrt{\frac{N}{l(q-2l)(N-q+l)}} \exp\left[NG(x_l, y_q)\right], \tag{A.53}$$

where, following the analogous construction in Lemma A.5, we introduced

$$G(x,y) := -4h(y/2) + h(x, y-2x, 1-y+x) + (y-2x)\log(14). \tag{A.54}$$

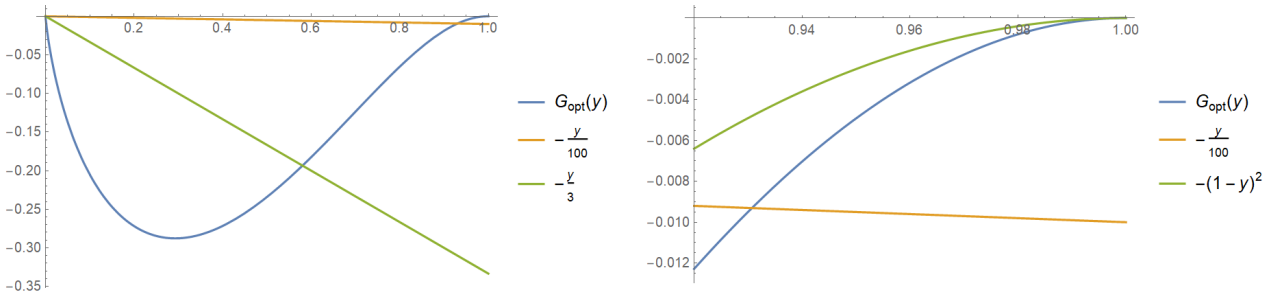

Figure A.5: Function $G_{opt}(y) = G(x_{opt}(y), y)$ where G is defined in Eq. (A.54) and $x_{opt}(y)$ is defined in Eq. (A.55). The function is presented alongside simple analitical lower bounds are valid in specific intervals formulated in Eq. (A.56).

As in the case of passive FLO, our strategy is to upper bound $G(x, y)$ by a function that allows for analytical treatment. To this end, we first optimize $G(x, y)$ over $x \in [0, y/2]$ for fixed $y \in [0, 1]$. Solving for the critical points gives the following optimal solution $x_{opt} \in [0, y/2]$ (at the extremal points of this interval function $G(x, y)$, treated as a function of x for fixed y, takes smaller values)

$$x_{opt}(y) = \frac{1}{96}\left(-49 + 48y + 7\sqrt{40 - 96y + 48y^2}\right) . \tag{A.55}$$

The maximum of $G(x, y)$ over $x \in [0, y/2]$, $G_{opt}(y) \coloneqq G(x_{opt}(y), y)$ is a continuous function of $y \in [0, 1]$ and also analytic for $y \in (0, 1)$. We can bound $G_{opt}(y)$ in the following way (see Fig. A.5)

$$G_{opt}(y) \leq -y/3 \quad \text{for } y \in [0, 1/2] , \tag{A.56}$$

$$G_{opt}(y) \leq -y/100 \quad \text{for } y \in [1/5, 0.925] , \tag{A.57}$$

$$G_{opt}(y) \leq -(1 - y)^2 \quad \text{for } y \in [0.925, 1] . \tag{A.58}$$

We shall need much more refined information about $G(x, y)$ than in the case of analogous considerations for passive FLO. Namely, we will need to control how fast $G(x, y)$ decays as a function of $x - x_{opt}(y)$, for fixed y. To this end we compute for $x \in (0, y/2)$, $y \in (0, 1)$

$$\partial_x^2 G(x, y) = -\left(\frac{1}{x} + \frac{1}{1 - y + x} + \frac{4}{y - 2x}\right) . \tag{A.59}$$

From the above expression we get[1]

$$\partial_x^2 G(x,y) \leq -16 \text{ for } x \in (0, y/2) \text{ and } \partial_x^2 G(x,y) \leq -\frac{2}{3x_{opt}(y)} \text{ for } x \in \left[\frac{x_{opt}(y)}{2}, \frac{3x_{opt}(y)}{2}\right] . \tag{A.60}$$

Using the analyticity of $G(x,y)$ as a function of x inside the interval $(0, y/2)$, we can Taylor expand it around $x_{opt}(y)$ (for fixed value of y):

$$G(x,y) = G_{opt}(y) + (\partial_x G(x_{opt}(y), y))(x - x_{opt}(y)) + \int_{x_{opt}(y)}^{x} d\tau \partial_\tau G(\tau, y) . \tag{A.61}$$

Using the fact that $x_{opt}(y)$ is a critical point and bounds, identity

$$\partial_\tau G(\tau, y) = \int_{x_{opt}(y)}^{\tau} dx \partial_x^2 G(x,y) \tag{A.62}$$

and bounds from Eq. (A.60) we get finally get

$$G(x,y) \leq G_{opt}(y) - 8(x - x_{opt}(y))^2 \qquad \text{for } x \in [0, y/2] , \ y \in [0,1] , \tag{A.63}$$

$$G(x,y) \leq G_{opt}(y) - \frac{1}{3x_{opt}(y)}(x - x_{opt}(y))^2 \quad \text{for } x \in \left[\frac{x_{opt}(y)}{2}, \frac{3x_{opt}(y)}{2}\right] , \ y \in [0,1] . \tag{A.64}$$

Coming back to the bound on $\mathcal{B}_{gen}$ from (A.53), similarly to the case of passive FLO, due to constrains on l, the sum appearing in (A.53) effectively starts from $q = 3$. Moreover, we also note that $l(q - 2l) \geq (q-2)/2$ and therefore

$$\sqrt{\frac{N}{l(q-2l)(N-q+l)}} \leq \sqrt{\frac{2N}{(q-2)(N-q+l)}} \leq \sqrt{\frac{2N}{N-2}} , \tag{A.65}$$

where in the second inequality we used the fact that $q \in [3, N]$ and $l \geq 1$. Using the above and expanding the expression in (A.53) in the different intervals defined in (A.56) we obtain

$$\mathcal{B}_{gen} \leq \frac{2CA}{c}\sqrt{\frac{N}{N-2}}\left(\sum_{q=3}^{q\leq N/2}\sum_{l=1}^{l<q/2} \exp[-q/2] + \sum_{q>N/2}^{q<0.925N}\sum_{l=1}^{l<q/2} \exp[-q/100]\right) \tag{A.66}$$

$$+ \frac{\sqrt{2}CA}{c}\sum_{q>0.925N}^{N}\sum_{l=1}^{l<q/2}\sqrt{\frac{N}{l(q-2l)(N-q+l)}}\exp\left[NG(x_l, y_q)\right] . \tag{A.67}$$

[1] It is easy to check that $3/2x_{opt}(y) \leq y/2$.

Two sums from Eq. (A.66) can be handled analogously as in the case of passive FLO:

$$\frac{2CA}{c}\sqrt{\frac{N}{N-2}}\left(\sum_{q=3}^{q\le N/5}\sum_{l=1}^{l<q/2}\exp[-q/3]+\sum_{q>N/2}^{q<0.925N}\sum_{l=1}^{l<q/2}\exp[-q/100]\right) \tag{A.68}$$

$$\le \frac{2CA}{c}\sqrt{\frac{N}{N-2}}\left(\sum_{q=3}^{\infty}(q/2)\exp(-q/3)++(N^{3/2}/2)\exp\left[-\frac{N}{200}\right]\right). \tag{A.69}$$

$$= \frac{2CA}{c}\sqrt{\frac{N}{N-2}}\left(\frac{3e^{1/3}-2}{2e^{2/3}(e^{1/3}-1)}+(N^{3/2}/4)\exp\left[-\frac{N}{200}\right]\right) \le 2\,, \tag{A.70}$$

where the last inequality is valid for $N \ge 1800$. The sum in (A.67) will be analyzed using inequalities (A.63) and (A.64). For fixed y_q (Which corresponds to $q = y_q N$) we set $l_{opt}(y_q) = x_{opt}(y_q)N$ and divide the range of summation over l in (A.67) into two parts that corresponds to intervals in bounds (A.63) and (A.64) respectively :

$$\mathcal{L}_q^{\max} = \left\{ l \;\middle|\; \frac{1}{2}l_{opt}(y_q) \le l \le \frac{3}{2}l_{opt}(y_q) \right\}\,, \tag{A.71}$$

$$\mathcal{L}_q^{\text{gen}} = \left\{ l \;\middle|\; 1 \le l < \frac{1}{2}l_{opt}(y_q) \text{ or } \frac{3}{2}l_{opt}(y_q) < l < q/2 \right\}\,. \tag{A.72}$$

$$\tag{A.73}$$

It is now straightforward to verify that:

$$\sqrt{\frac{N}{l(q-2l)(N-q+l)}} \le \sqrt{\frac{4N}{l_{opt}(q-3l_{opt})l_{op}}} \text{ for } l \in \mathcal{L}_q^{\max}\,, \tag{A.74}$$

It is now straightforward to verify that:

$$\sqrt{\frac{N}{l(q-2l)(N-q+l)}} \le \sqrt{\frac{4N}{l_{opt}(q-3l_{opt})l_{op}}} \text{ for } l \in \mathcal{L}_q^{\max}\,, \tag{A.75}$$

Finally, we arrive at the following bound

$$\frac{\sqrt{2}CA}{c}\sum_{q>0.925N}^{N}\sum_{l=1}^{l<q/2}\sqrt{\frac{N}{l(q-2l)(N-q+l)}}\exp\left[NG(x_l,y_q)\right] \tag{A.76}$$

$$\le \frac{\sqrt{2}CA}{c}\sum_{q>0.925N}^{N}\exp\left[NG_{opt}(y_q)\right]\sqrt{\frac{4N}{l_{opt}(q-3l_{opt})l_{opt}}}\sum_{l\in\mathcal{L}_q^{\max}}\exp\left(-\frac{(l-l_{opt})^2}{3l_{opt}}\right) \tag{A.77}$$

$$+\sqrt{\frac{N}{N-2}}\frac{CA}{c}\sum_{q>0.925N}^{N} q\exp\left[NG_{opt}(y_q)\right]\exp\left(-2x_{opt}^2N\right)\ , \tag{A.78}$$

where we used (A.65) to get (A.78). We first analyze the second sum. We using (A.56) we obtain

$$\sum_{q>0.925N}^{N} q\exp\left[NG_{opt}(y_q)\right] \leq N\sum_{q>0.925N}^{N}\exp\left[\frac{(N-q)^2}{N}\right] \leq N(1+\frac{\sqrt{\pi N}}{2}) \leq N^{\frac{3}{2}}\ . \tag{A.79}$$

where

$$\sum_{x=0}^{\infty}\exp\left(-\frac{x^2}{a}\right) \leq 1+\int_0^{\infty} dx\exp\left(-\frac{x^2}{a}\right) = 1+\frac{\sqrt{\pi a}}{2}\ , \tag{A.80}$$

valid for all $a > 0$, and $N \geq 100$. Importantly, for $q > 0.925N$ (which corresponds to $y \geq 0.925$), we have $x_{opt} \geq 0.03$. Using this and assuming $N \geq 7000$, we finally obtain

$$\sqrt{\frac{N}{N-2}}\frac{CA}{c}\sum_{q>0.925N}^{N} q\exp\left[NG_{opt}(y_q)\right]\exp\left(-2x_{opt}^2N\right) \leq \sqrt{\frac{N}{N-2}}\frac{CA}{c}N^{\frac{3}{2}}\exp\left(-\frac{9}{5000}N\right) \leq 1\ . \tag{A.81}$$

We use similar methods to bound (A.77). First, we upper bound the exponential sum

$$\sum_{l\in\mathcal{L}_q^{\max}}\exp\left(-\frac{(l-l_{opt})^2}{3l_{opt}}\right) \leq 1+\sqrt{\pi 3l_{opt}} \leq \frac{10}{3}\sqrt{l_{opt}}\ , \tag{A.82}$$

which allows estimate

$$\sqrt{\frac{4N}{l_{opt}(q-3l_{opt})l_{opt}}}\sum_{l\in\mathcal{L}_q^{\max}}\exp\left(-\frac{(l-l_{opt})^2}{3l_{opt}}\right) \leq \frac{10}{3}\sqrt{\frac{4N}{l_{opt}(q-3l_{opt})}} \leq \frac{10}{3}\sqrt{\frac{4}{(0.03)(0.7N)}} = \frac{20}{3}\sqrt{\frac{1000}{21N}}\ , \tag{A.83}$$

where in the second inequality we used that for $q \geq 0.925N$ we have $l_{opt}(y_q) \geq 0.03N$ and $q - 3l_{opt}(y_q) \geq 0.7N$. Inserting thin inequality to (A.77) and again using (A.79) gives that for $N \geq 7000$

$$\frac{\sqrt{2}CA}{c}\sum_{q>0.925N}^{N}\sum_{l=1}^{l<q/2}\sqrt{\frac{N}{l(q-2l)(N-q+l)}}\exp\left[NG(x_l,y_q)\right] \leq 1+\frac{\sqrt{2}CA}{c}\sqrt{\frac{1000}{21}} \leq 12.7\ . \tag{A.84}$$

Combining this estimate with the bound (A.70) and using (A.66), we finally obtain that for $N \geq 7000$

$$\mathcal{B}_{gen} \leq 14.7\ . \tag{A.85}$$

Finally, combining bounds (A.48), (A.52) and (A.85) together with $\mathcal{B}_{k=0} = 1$ in inequality (A.38) we see that for $N \geq 7000$,

$$\frac{K_{\text{act}}}{2^{8N}} \leq \frac{1}{\sqrt{\pi N}}(\mathcal{B}_{q=0} + \mathcal{B}_{l=0} + \mathcal{B}_{q=2l} + \mathcal{B}_{gen}) \leq \frac{16.2}{\sqrt{\pi N}} \,. \tag{A.86}$$

For $N \leq 7000$, the validity of the bound can be verified numerically as shown in Fig. A.4, which completes the proof.

■

Appendix B

Some proof of results for Part III

These proofs have been included in this appendix so the main text is easier to follow.

B.1 Extending Yoshida's method to 10th order

Here we explain how to extend the method of Yoshida to obtain the equations for a 10th order integrator. We find that the number of equations required is at least 15, we use this fact in our optimization procedure to find new product formulae. We will follow the notation from Corollary 6.3, noting that

$$S_2(t) = e^{t\alpha_1+t^3\alpha_3+t^5\alpha_5+t^7\alpha_7+t^9\alpha_9+O(t^{11})} \tag{B.1}$$

where the α_j are defined as commutators of operators. It will be useful to define the following commutators

$$\beta_9 = [\alpha_1, \alpha_1, \alpha_7], \tag{B.2}$$

$$\gamma_9^{(1)} = [\alpha_1, \alpha_3, \alpha_5], \tag{B.3}$$

$$\gamma_9^{(2)} = [\alpha_3, \alpha_1, \alpha_5], \tag{B.4}$$

$$\gamma_9^{(3)} = [\alpha_5, \alpha_1, \alpha_3], \tag{B.5}$$

$$\delta_9^{(1)} = [\alpha_1^4, \alpha_5], \tag{B.6}$$

$$\delta_9^{(2)} = [\alpha_3, \alpha_1^3, \alpha_3], \tag{B.7}$$

$$\delta_9^{(3)} = [\alpha_1, \alpha_3, \alpha_1^2, \alpha_3], \tag{B.8}$$

$$\epsilon_9 = [\alpha_1^6, \alpha_3]. \tag{B.9}$$

Yoshida also defined the commutators

$$\beta_5 = [\alpha_1, \alpha_1, \alpha_3], \tag{B.10}$$

$$\beta_7 = [\alpha_1, \alpha_1, \alpha_5], \tag{B.11}$$

$$\gamma_7 = [\alpha_3, \alpha_3, \alpha_1], \tag{B.12}$$

$$\delta_7 = [\alpha_1, \alpha_1, \alpha_1, \alpha_1, \alpha_3]. \tag{B.13}$$

To find the equations we need to solve to find 10th order product formulae, we are also required to apply the symmetric BCH formula from Corollary 6.3 up to commutators with 7 operators. The reason we do not consider an expansion up to 9 operator commutators is that the only commutator contributing terms of 9th order and with 9 operators in the commutator is of the form $[\alpha_1, \alpha_1, \cdots, \alpha_1]$ (considering commutators with other α_i will only contribute higher orders than 9). To obtain the coefficients multiplying the commutators with 7 operators in the symmetric BCH expansion, we use the algorithm defined in Section V of Ref. [VV16]. Note that the algorithm in that work generates the scalar coefficients multiplying products of operators rather than their commutators. We need to express the symmetric BCH expansion in the so called Ph. Hall basis, which is a basis for writing Lie monomials consisting of commutators of the generators of the Lie algebra (for a list of operators in this basis, see Table 1 in [DKD20]); we obtain the coefficients for the Ph. Hall basis by solving the corresponding linear problem of changing from one basis to another. As an example, consider the term with 3 operators in the symmetric BCH expansion from Corollary 6.3, given as $\alpha_3 = \frac{1}{12}[Y,[Y,X]] - \frac{1}{24}[X,[X,Y]]$. We can also express the commutators as products by expanding out the commutators, which gives $\tilde{\alpha}_3 = \frac{1}{24}\left\{2Y^2X - 4YXY - 2XY^2 - X^2Y + 2XYX + YX^2\right\}$. The algorithm in [VV16] outputs expressions with the commutators expanded out as in $\tilde{\alpha}_3$. In order to obtain the original expression α_3, we write $\tilde{\alpha}_3 = a[Y,[Y,X]] + b[X,[X,Y]]$ with $a, b \in \mathbb{R}$ and expand the commutators $[Y,[Y,X]]$ and $[X,[X,Y]]$. This gives several

linear equations that can be written in terms of a matrix. By inverting this matrix, we obtain the coefficients a and b.

Thus, by using C and D as generic variables for operators we have

$$e^C e^D e^C = e^Z, \tag{B.14}$$

where

$$\begin{aligned} Z = {} & 2C + D + \frac{1}{6}([D,D,C] - [C,C,D]) \\ & + \frac{7}{360}[C,C,C,C,D] - \frac{1}{360}[D,D,D,D,C] \\ & + \frac{1}{90}[C,D,D,D,C] + \frac{1}{45}[D,C,C,C,D] \\ & - \frac{1}{60}[C,C,D,D,C] + \frac{1}{30}[D,D,C,C,D] \\ & - \frac{31}{15120}[C,C,C,C,C,C,D] - \frac{31}{5040}[D,C,C,C,C,C,D] \\ & - \frac{13}{1890}[D,D,C,C,C,C,D] - \frac{53}{15120}[D,D,D,C,C,C,D] \\ & - \frac{1}{1260}[D,D,D,D,C,C,D] - \frac{1}{15120}[D,D,D,D,D,C,D] + \mathcal{R}_{(9\leq)}. \end{aligned} \tag{B.15}$$

Where $\mathcal{R}_{(9\leq)}$ is an infinite sum with commutators of an odd number of operators (equal to or higher than 9). Now we prove the following Lemma which will allow us to derive the equations for 10th order product formulae.

Lemma B.1. *Following the notation of the procedure given by Yoshida in Eq. (6.15), we have that for all $m \in \mathbb{N}$*

$$\begin{aligned} S^{(m)}(\tau) = \exp\Big\{ & \tau A_{1,m}\alpha_1 + \tau^3 A_{3,m}\alpha_3 + \tau^5(A_{5,m}\alpha_5 + B_{5,m}\beta_5) \\ & + \tau^7(A_{7,m}\alpha_7 + B_{7,m}\beta_7 + C_{7,m}\gamma_7 + D_{7,m}\delta_7) \\ & + \tau^9(A_{9,m}\alpha_9 + B_{9,m}\beta_9 + C^{(1)}_{9,m}\gamma^{(1)}_9 + C^{(2)}_{9,m}\gamma^{(2)}_9 + C^{(3)}_{9,m}\gamma^{(3)}_9 \\ & + D^{(1)}_{9,m}\delta^{(1)}_9 + D^{(2)}_{9,m}\delta^{(2)}_9 + D^{(3)}_{9,m}\delta^{(3)}_9 + E_{9,m}\epsilon_9) + O(\tau^{11})\Big\}, \end{aligned} \tag{B.16}$$

where the variables in upper case denote polynomials in the variables $(w_1, \cdots, w_m)$.

Proof. We proceed by induction. First, note that the statement is true for the case $m = 0$,

$$S^{(m=0)}(\tau) = S_2(w_0\tau), \tag{B.17}$$

$$= \exp\{t w_0\alpha_1 + t^3 w_0^3\alpha_3 + t^5 w_0^5\alpha_5 + t^7 w_0^7\alpha_7 + t^9 w_0^9\alpha_9 + O(t^{11})\}. \tag{B.18}$$

This clearly has the form of Eq. (B.16) by taking $A_{j,m=0} = w_0^j$ and all other scalar variables as 0.

Assume now that Eq. (B.16) is correct, we want to derive an expression for $S^{(m+1)}$. We then have $S^{(m+1)}(\tau) = S_2(w_{m+1}\tau)S^{(m)}(\tau)S_2(w_{m+1}\tau)$ and thus

$$\begin{aligned}
S_2(w_{m+1}\tau)S^{(m)}(\tau)S_2(w_{m+1}\tau) = \exp&\left\{\tau w_{m+1}\alpha_1 + \tau^3 w_{m+1}^3\alpha_3 + \tau^5 w_{m+1}^5\alpha_5 + \tau^7 w_{m+1}^7\alpha_7 + \tau^9 w_{m+1}^9\alpha_9 + O(\tau^{11})\right\} \\
&\times \exp\left\{\tau A_{1,m}\alpha_1 + \tau^3 A_{3,m}\alpha_3 + \tau^5(A_{5,m}\alpha_5 + B_{5,m}\beta_5)\right. \\
&+ \tau^7(A_{7,m}\alpha_7 + B_{7,m}\beta_7 + C_{7,m}\gamma_7 + D_{7,m}\delta_7) \\
&+ \tau^9(A_{9,m}\alpha_9 + B_{9,m}\beta_9 + C_{9,m}^{(1)}\gamma_9^{(1)} + C_{9,m}^{(2)}\gamma_9^{(2)} + C_{9,m}^{(3)}\gamma_9^{(3)} \\
&\left.+ D_{9,m}^{(1)}\delta_9^{(1)} + D_{9,m}^{(2)}\delta_9^{(2)} + D_{9,m}^{(3)}\delta_9^{(3)} + E_{9,m}\epsilon_9) + O\left(\tau^{11}\right)\right\} \\
&\times \exp\left\{\tau w_{m+1}\alpha_1 + \tau^3 w_{m+1}^3\alpha_3 + \tau^5 w_{m+1}^5\alpha_5 + \tau^7 w_{m+1}^7\alpha_7 + \tau^9 w_{m+1}^9\alpha_9 + O\left(\tau^{11}\right)\right.
\end{aligned} \tag{B.19}$$

We compute the right-hand-side (RHS) of Eq. (B.19) applying the symmetric BCH formula from Corollary 6.3. Writing the RHS as $e^C e^D e^C$, we have that

$$C = \tau w_{m+1}\alpha_1 + \tau^3 w_{m+1}^3\alpha_3 + \tau^5 w_{m+1}^5\alpha_5 + \tau^7 w_{m+1}^7\alpha_7 + \tau^9 w_{m+1}^9\alpha_9 + O(\tau^{11}) \tag{B.20}$$

$$\begin{aligned}
D = &\tau A_{1,m}\alpha_1 + \tau^3 A_{3,m}\alpha_3 + \tau^5(A_{5,m}\alpha_5 + B_{5,m}\beta_5) + \tau^7(A_{7,m}\alpha_7 + B_{7,m}\beta_7 + C_{7,m}\gamma_7 + D_{7,m}\delta_7) \\
&+ \tau^9(A_{9,m}\alpha_9 + B_{9,m}\beta_9 + C_{9,m}^{(1)}\gamma_9^{(1)} + C_{9,m}^{(2)}\gamma_9^{(2)} + C_{9,m}^{(3)}\gamma_9^{(3)} + D_{9,m}^{(1)}\delta_9^{(1)} + D_{9,m}^{(2)}\delta_9^{(2)} + D_{9,m}^{(3)}\delta_9^{(3)} + E_{9,m}\epsilon_9) \\
&+ O\left(\tau^{11}\right).
\end{aligned} \tag{B.21}$$

We then compute the commutators of C and D that appear in the symmetric BCH formula, here we give the resulting 9th order operators after applying the commutators. When we write $[C, D, \cdots, C]_9$, the subscript indicates that we are only keeping the 9th

order terms when expanding the commutator. We will explain in detail how to compute the commutator $C = [D, D, C]_9$, the other commutators are computed in a similar way. Since we only need to consider terms of 9th order when computing C, each term will have contributions from each operator inside C (in this case two operators D and one C) which is comprised of odd numbers that sum up to 9 such that the commutator is non-zero. We then have that

$$\begin{aligned}
[D, D, C]_9 = {} & \tau^9(A_{1,m}^2 w_{m+1}^7 - A_{1,m}A_{7,m}w_{m+1})[\alpha_1, \alpha_1, \alpha_7] \\
& + \tau^9 A_{1,m}B_{7,m}w_{m+1}[\alpha_1, \beta_7, \alpha_1] \\
& + \tau^9 A_{1,m}C_{7,m}w_{m+1}[\alpha_1, \gamma_7, \alpha_1] \\
& + \tau^9 A_{1,m}D_{7,m}w_{m+1}[\alpha_1, \delta_7, \alpha_1] \\
& + \tau^9(A_{1,m}A_{3,m}w_{m+1} - A_{1,m}A_{5,m}w_{m+1}^3)[\alpha_1, \alpha_3, \alpha_5] \\
& + \tau^9 A_{1,m}B_{5,m}w_{m+1}^3[\alpha_1, \beta_5, \alpha_3] \\
& + \tau^9(A_{3,m}A_{1,m}w_{m+1}^5 - A_{3,m}A_{5,m}w_{m+1})[\alpha_3, \alpha_1, \alpha_5] \\
& + \tau^9 A_{1,m}B_{5,m}w_{m+1}[\alpha_3, \beta_5, \alpha_1] \\
& + \tau^9(A_{5,m}A_{1,m}w_{m+1}^3 - A_{5,m}A_{3,m}w_{m+1})[\alpha_5, \alpha_1, \alpha_3] \\
& + \tau^9 B_{5,m}A_{3,m}w_{m+1}[\beta_5, \alpha_3, \alpha_1]. \qquad \text{(B.22)}
\end{aligned}$$

Given how we have defined the commutator, we have then

$$\begin{aligned}
[D, D, C]_9 = {} & \tau^9(A_{1,m}^2 w_{m+1}^7 - A_{1,m}A_{7,m}w_{m+1})\beta_9 - \tau^9 A_{1,m}B_{7,m}w_{m+1}\delta_9^{(1)} \\
& + \tau^9 A_{1,m}C_{7,m}w_{m+1}\delta_9^{(3)} - \tau^9 A_{1,m}D_{7,m}w_{m+1}\epsilon_9 \\
& + \tau^9(A_{1,m}A_{3,m}w_{m+1}^5 - A_{1,m}A_{5,m}w_1^3)\gamma_9^{(1)} - \tau^9 A_{1,m}B_{5,m}w_{m+1}^3\delta_9^{(3)} \\
& + \tau^9(A_{3,m}A_{1,m}w_{m+1}^5 - A_{3,m}A_{5,m}w_{m+1})\gamma_9^{(2)} - \tau^9 A_{3,m}B_{5,m}w_{m+1}\delta_9^{(2)} \\
& + \tau^9(A_{5,m}A_{1,m}w_{m+1}^3 - A_{5,m}A_{3,m}w_{m+1}^2)\gamma_9^{(3)} \\
& + \tau^9(B_{5,m}A_{1,m}w_{m+1}^3 - B_{5,m}A_{3,m}w_{m+1})(\delta_9^{(2)} - \delta_9^{(3)}) \qquad \text{(B.23)} \\
[C, C, D]_9 = {} & \tau^9(w_{m+1}^2 A_{7,m} - w_{m+1}^8 A_{1,m})\beta_9 + \tau^9 w_{m+1}^2 B_{7,m}\delta_9^{(1)} \\
& - \tau^9 w_{m+1}^2 C_{7,m}\delta_9^{(3)} + \tau^9 w_{m+1}^2 D_{7,m}\epsilon_9
\end{aligned}$$

$$
\begin{aligned}
&+ \tau^9(w_{m+1}^4 A_{5,m} - w_{m+1}^6 A_{3,m})\gamma_9^{(1)} + \tau^9 w_{m+1}^4 B_{5,m}\delta_9^{(3)} \\
&+ \tau^9(w_{m+1}^4 A_{5,m} - w_{m+1}^8 A_{1,m})\gamma_9^{(2)} + \tau^9 w_{m+1}^4 B_{5,m}\delta_9^{(2)} \\
&+ \tau^9(w_{m+1}^6 A_{3,m} - w_{m+1}^8 A_{1,m})\gamma_9^{(3)} \qquad \text{(B.24)}
\end{aligned}
$$

$$
\begin{aligned}
[C,C,C,C,D]_9 = {} & \tau^9(w_{m+1}^4 A_{5,m} - w_{m+1}^8 A_{1,m})\delta_9^{(1)} + \tau^9 A_{1,m}^3 B_{5,m} w_{m+1}\epsilon_9 \\
&+ \tau^9(w_{m+1}^6 A_{3,m} - w_{m+1}^8 A_{1,m})\delta_9^{(2)} \\
&+ \tau^9 2(w_{m+1}^6 A_{3,m} - w_{m+1}^8 A_{1,m})\delta_9^{(3)} \qquad \text{(B.25)}
\end{aligned}
$$

$$
\begin{aligned}
[D,D,D,D,C]_9 = {} & \tau^9(A_{1,m}^4 w_{m+1}^5 - A_{1,m}^3 A_{5,m} w_{m+1})\delta_9^{(1)} - \tau^9 A_{1,m}^3 B_{5,m} w_{m+1}\epsilon_9 \\
&+ \tau^9(A_{3,m} A_{1,m}^3 w_{m+1}^3 - A_{3,m}^2 A_{1,m}^2 w_{m+1})\delta_9^{(2)} \\
&+ \tau^9 2(A_{1,m}^3 A_{3,m} w_{m+1}^3 - A_{1,m}^2 A_{3,m}^2 w_{m+1})\delta_9^{(3)} \qquad \text{(B.26)}
\end{aligned}
$$

$$
\begin{aligned}
[C,D,D,D,C]_9 = {} & \tau^9(w_{m+1}^6 A_{1,m}^3 - w_{m+1}^2 A_{1,m}^2 A_{5,m})\delta_9^{(1)} - \tau^9 A_{1,m}^2 B_{5,m} w_{m+1}^2\epsilon_9 \\
&+ \tau^9(w_{m+1}^6 A_{1,m}^3 - w_{m+1}^4 A_{1,m}^2 A_{3,m})\delta_9^{(2)} \\
&+ \tau^9 2(w_{m+1}^4 A_{3,m} A_{1,m}^2 - w_{m+1}^2 A_{3,m}^2 A_{1,m})\delta_9^{(3)} \qquad \text{(B.27)}
\end{aligned}
$$

$$
\begin{aligned}
[D,C,C,C,D]_9 = {} & \tau^9(A_{1,m} A_{5,m} w_{m+1}^3 - A_{1,m}^2 w_{m+1}^7)\delta_9^{(1)} + \tau^9 w_{m+1}^3 A_{1,m} B_{5,m}\epsilon_9 \\
&+ \tau^9(A_{3,m}^2 w_{m+1}^3 - A_{1,m} A_{3,m} w_{m+1}^5)\delta^{(2)} \\
&+ \tau^9 2(A_{1,m} A_{3,m} w_{m+1}^5 - A_{1,m}^2 w_{m+1}^7)\delta_9^{(3)} \qquad \text{(B.28)}
\end{aligned}
$$

$$
\begin{aligned}
[C,C,D,D,C]_9 = {} & \tau^9(w_{m+1}^7 A_{1,m}^2 - w_{m+1}^3 A_{1,m} A_{5,m})\delta_9^{(1)} - w_{m+1}^3 A_{1,m} B_{5,m}\epsilon_9 \\
&+ \tau^9(w_{m+1}^7 A_{1,m}^2 - w_{m+1}^5 A_{1,m} A_{3,m})\delta_9^{(2)} \\
&+ \tau^9(w_{m+1}^5 A_{1,m} A_{3,m} - w_{m+1}^3 A_{3,m}^2)\delta_9^{(3)} \qquad \text{(B.29)}
\end{aligned}
$$

$$
\begin{aligned}
[D,D,C,C,D]_9 = {} & \tau^9(A_{1,m}^2 A_{5,m} w_{m+1}^2 - A_{1,m}^3 w_{m+1}^6)\delta_9^{(1)} + A_{1,m}^2 B_{5,m} w_{m+1}^2\epsilon_9 \\
&+ (A_{3,m}^2 A_{1,m} w_{m+1}^2 - A_{3,m} A_{1,m}^2 w_{m+1}^4)\delta_9^{(2)} \\
&+ \tau^9(A_{3,m}^2 A_{1,m} w_{m+1}^2 - A_{3,m} A_{1,m}^2 w_{m+1}^4)\delta_9^{(3)} + \tau^9(A_{1,m}^2 A_{3,m} w_{m+1}^4 - A_{1,m}^3 w_{m+1}^6)\delta_9^{(3)} \\
& \qquad \text{(B.30)}
\end{aligned}
$$

$$
[C,C,C,C,C,C,D]_9 = \tau^9(w_{m+1}^6 A_{3,m} - w_{m+1}^8 A_{1,m})\epsilon_9
$$

$$
[D,C,C,C,C,C,D]_9 = \tau^9(w_{m+1}^5 A_{1,m} A_{3,m} - w_{m+1}^7 A_{1,m}^2)\epsilon_9 \qquad \text{(B.31)}
$$

$$
[D,D,C,C,C,C,D]_9 = \tau^9(A_{1,m}^2 A_{3,m} w_{m+1}^4 - A_{1,m}^3 w_{m+1}^6)\epsilon_9 \qquad \text{(B.32)}
$$

$$[D,D,D,C,C,C,D]_9 = \tau^9(A^3_{1,m}A_{3,m}w^3_{m+1} - A^4_{1,m}w^5_{m+1})\epsilon_9 \tag{B.33}$$

$$[D,D,D,D,C,C,D]_9 = \tau^9(A^4_{1,m}A_{3,m}w^2_{m+1} - A^5_{1,m}w^4_{m+1})\epsilon_9 \tag{B.34}$$

$$[D,D,D,D,D,C,D]_9 = \tau^9(A^5_{1,m}A_{3,m}w_{m+1} - A^6_{1,m}w^3_{m+1})\epsilon_9 \tag{B.35}$$

Note that all the terms previously computed have terms that can be written as in Eq. (B.16), thus proving that $S^{(m+1)}$ can also be written in this way. ■

Having proved Lemma B.1, we can now compute the polynomials in Eq. (B.16). The polynomials are obtained from the recursion in Eq. (B.19), the left hand side corresponds to $S^{(m+1)}$ and can can be written as a single exponential, the same is true of the right side which is written as a single exponential. We have then the following polynomials:

$$A_{9,m+1} = A_{9,m} + 2w^9_{m+1} \tag{B.36}$$

$$\begin{aligned} B_{9,m+1} = B_{9,m} &+ \frac{1}{6}(A^2_{1,m}w^7_{m+1} - A_{1,m}A_{7,m}w_{m+1}) \\ &- \frac{1}{6}(A_{7,m}w^2_{m+1} - A_{1,m}w^8_{m+1}) \end{aligned} \tag{B.37}$$

$$\begin{aligned} C^{(1)}_{9,m+1} = C^{(1)}_{9,m} &+ \frac{1}{6}(A^2_{3,m}A_{1,m}w^5_{m+1} - A_{1,m}A_{5,m}w^3_{m+1}) \\ &- \frac{1}{6}(A_{5,m}w^4_{m+1} - A_{3,m}w^6_{m+1}) \end{aligned} \tag{B.38}$$

$$\begin{aligned} C^{(2)}_{9,m+1} = C^{(2)}_{9,m} &+ \frac{1}{6}(A^2_{3,m}A_{1,m}w^5_{m+1} - A_{3,m}A_{5,m}w_{m+1}) \\ &- \frac{1}{6}(A_{5,m}w^4_{m+1} - A_{1,m}w^8_{m+1}) \end{aligned} \tag{B.39}$$

$$\begin{aligned} C^{(3)}_{9,m+1} = C^{(3)}_{9,m} &+ \frac{1}{6}(A_{5,m}A_{1,m}w^3_{m+1} - A_{3,m}A_{5,m}w_{m+1}) \\ &- \frac{1}{6}(A_{3,m}w^6_{m+1} - A_{1,m}w^8_{m+1}) \end{aligned} \tag{B.40}$$

$$\begin{aligned} D^{(1)}_{9,m+1} = D^{(1)}_{9,m} &- \frac{1}{6}(A_{1,m}B_{7,m}w_{m+1} + w^2_{m+1}B_{7,m}) \\ &+ \frac{7}{360}(A_{5,m}w^4_{m+1} - w^8_{m+1}A_{1,m}) \\ &- \frac{1}{360}(A^4_{1,m}w^5_{m+1} - A^3_{1,m}A_{5,m}w_{m+1}) \\ &+ \frac{1}{90}(A^3_{1,m}w^6_{m+1} - A^2_{1,m}A_{5,m}w^2_{m+1}) \\ &+ \frac{1}{45}(A_{1,m}A_{5,m}w^3_{m+1} - A^2_{1,m}w^7_{m+1}) \end{aligned}$$

$$
\begin{aligned}
&- \frac{1}{60}(A_{1,m}^2 w_{m+1}^7 - A_{1,m}A_{5,m}w_{m+1}^3) \\
&+ \frac{1}{30}(A_{1,m}^2 A_{5,m} w_{m+1}^2 - A_{1,m}^3 w_{m+1}^6) \qquad \text{(B.41)}
\end{aligned}
$$

$$
\begin{aligned}
D_{9,m+1}^{(2)} = D_{9,m}^{(2)} &- \frac{1}{6}(A_{3,m}B_{5,m}w_{m+1} + w_{m+1}^4 B_{5,m}) \\
&+ \frac{7}{360}(A_{3,m}w_{m+1}^6 - w_{m+1}^8 A_{1,m}) \\
&- \frac{1}{360}(A_{1,m}^3 A_{3,m} w_{m+1}^3 - A_{1,m}^2 A_{3,m}^2 w_{m+1}) \\
&+ \frac{1}{90}(A_{1,m}^3 w_{m+1}^6 - A_{1,m}^2 A_{3,m} w_{m+1}^4) \\
&+ \frac{1}{45}(A_{3,m}^2 w_{m+1}^3 - A_{1,m}A_{3,m}w_{m+1}^5) \\
&- \frac{1}{60}(A_{1,m}^2 w_{m+1}^7 - A_{1,m}A_{3,m}w_{m+1}^5) \\
&+ \frac{1}{30}(A_{3,m}^2 A_{1,m} w_{m+1}^2 - A_{1,m}^2 A_{3,m} w_{m+1}^4) \qquad \text{(B.42)}
\end{aligned}
$$

$$
\begin{aligned}
D_{9,m+1}^{(3)} = D_{9,m}^{(3)} &- \frac{1}{6}(A_{1,m}B_{5,m}w_{m+1}^3 + w_{m+1}^4 B_{5,m}) \\
&+ \frac{14}{360}(A_{3,m}w_{m+1}^6 - w_{m+1}^8 A_{1,m}) \\
&- \frac{2}{360}(A_{1,m}^3 A_{3,m} w_{m+1}^3 - A_{1,m}^2 A_{3,m}^2 w_{m+1}) \\
&+ \frac{2}{90}(A_{1,m}^2 A_{3,m} w_{m+1}^4 - A_{1,m}A_{3,m}^2 w_{m+1}^2) \\
&+ \frac{2}{45}(A_{3,m}A_{1,m}w_{m+1}^5 - A_{1,m}^2 w_{m+1}^7) \\
&- \frac{1}{60}(A_{1,m}^2 w_{m+1}^7 - A_{1,m}A_{3,m}w_{m+1}^5) \\
&+ \frac{1}{30}(A_{3,m}^2 A_{1,m} w_{m+1}^2 - A_{1,m}^2 A_{3,m} w_{m+1}^4) \\
&+ \frac{1}{6}(A_{1,m}C_{7,m}w_{m+1} + w_{m+1}^2 c_{7,m}) \\
&- \frac{1}{60}(w_{m+1}^5 A_{1,m}A_{3,m} - w_{m+1}^3 A_{3,m}^2) \\
&+ \frac{1}{30}(A_{1,m}^2 A_{3,m} w_{m+1}^4 - A_{1,m}^3 w_{m+1}^6) \\
&- \frac{1}{6}(B_{5,m}A_{1,m}w_{m+1}^3 - B_{5,m}A_{3,m}w_{m+1}) \qquad \text{(B.43)}
\end{aligned}
$$

$$
\begin{aligned}
E_{9,m+1} = E_{9,m} &- \frac{1}{6}(A_{1,m}D_{7,m} - w_{m+1}^2 D_{7,m}) \\
&+ \frac{7}{360}w_{m+1}^4 B_{5,m}
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{1}{360} A_{1,m}^3 B_{5,m} w_{m+1} \\
&- \frac{1}{90} A_{1,m}^2 B_{5,m} w_{m+1}^2 \\
&+ \frac{1}{45} A_{1,m} B_{5,m} w_{m+1}^3 \\
&+ \frac{1}{60} A_{1,m} B_{5,m} w_{m+1}^3 \\
&+ \frac{1}{30} A_{1,m}^2 B_{5,m} w_{m+1}^2 \\
&- \frac{31}{15120} (w_{m+1}^6 A_{3,m} - w_{m+1}^8 A_{1,m}) \\
&- \frac{31}{5040} (w_{m+1}^5 A_{1,m} A_{3,m} - w_{m+1}^7 A_{1,m}^2) \\
&- \frac{13}{1890} (A_{1,m}^2 A_{3,m} w_{m+1}^4 - A_{1,m}^3 w_{m+1}^6) \\
&- \frac{53}{15120} (A_{1,m}^3 A_{3,m} w_{m+1}^3 - A_{1,m}^4 w_{m+1}^5) \\
&- \frac{1}{1260} (A_{1,m}^4 A_{3,m} w_{m+1}^2 - A_{1,m}^5 w_{m+1}^4) \\
&- \frac{1}{15120} (A_{1,m}^5 A_{3,m} w_{m+1} - A_{1,m}^6 w_{m+1}^3). \qquad \text{(B.44)}
\end{aligned}
$$

Then we obtain the polynomial equations for the tenth order product formula by imposing that $A_{1,m} = 1$ and all other terms are equal to zero. It can be seen that $C_{9,m}^2 = C_{9,m}^1 + C_{9,m}^3$, eliminating one equation and providing a set of 15 equations to solve.

B.2 Bounding error by total time evolution

In this section we show how the total evolution time appears in the error bound of a product formula. For an operator $W = X + Y$, with X and Y non-commuting, we want to bound the expression $\left\| S^{(m)}(t) - e^{tW} \right\|$. We assume $S^{(m)}(t)$ is a kth order product formula following Yoshida's ansatz.

We shall consider the Taylor expansion of both $S^{(m)}(t)$ and e^{tW}

$$S^{(m)}(t) = \mathcal{T}_k[S^{(m)}] + \mathcal{T}_{k<}[S^{(m)}] \qquad \text{(B.45)}$$

$$e^{tW} = \mathcal{T}_k[e^{tW}] + \mathcal{T}_{k<}[e^{tW}], \qquad \text{(B.46)}$$

where $\mathcal{T}_k$ is defined as the Taylor expansion up to order k and $\mathcal{T}_{k<}$ has the terms of the Taylor expansion from order $k+1$ to infinity. Note that since $S^{(m)}$ is a kth order product formula then $\mathcal{T}_k[S^{(m)}] = \mathcal{T}_k[e^{tW}]$ and thus

$$\left\| S^{(m)}(t) - e^{tW} \right\| = \left\| \mathcal{T}_{k<}[S^{(m)}] - \mathcal{T}_{k<}[e^{tW}] \right\| \tag{B.47}$$

$$\leq \left\| \mathcal{T}_{k<}[S^{(m)}] \right\| + \left\| \mathcal{T}_{k<}[e^{tW}] \right\|. \tag{B.48}$$

Now, it can be seen that the following bounds apply

$$\left\| \mathcal{T}_{k<}[S^{(m)}] \right\| \leq \sum_{r=2k+1}^{\infty} \frac{(\zeta t\Lambda)^r}{r!} \tag{B.49}$$

$$\leq \frac{(\zeta t\Lambda)^{2k+1}}{(2k+1)!} \sum_{r=0}^{\infty} \frac{(\zeta t\Lambda)^r}{r!} \tag{B.50}$$

$$= \frac{(\zeta t\Lambda)^{2k+1}}{(2k+1)!} \exp\left(\zeta t\Lambda\right), \tag{B.51}$$

where $\Lambda = \|X\| + \|Y\|$ and $\zeta = |w_0| + 2\sum_{i=1}^{m} |w_i|$ corresponds to the total evolution time. The first inequality comes from expanding out $\mathcal{T}_{k<}[S^{(m)}]$ and applying the triangle inequality to each term of the expansion, so that each operator X or Y in the sum becomes a scalar $\|X\|$ or $\|Y\|$ respectively. Similarly, we have

$$\left\| \mathcal{T}_{k<}[e^{tW}] \right\| = \sum_{r=2k+1}^{\infty} \frac{(t\|W\|)^r}{r!} \tag{B.52}$$

$$\leq \frac{(t\|W\|)^{2k+1}}{(2k+1)!} \sum_{r=0}^{\infty} \frac{(t\|W\|)^r}{r!} \tag{B.53}$$

$$= \frac{(t\|W\|)^{2k+1}}{(2k+1)!} \exp\{t\|W\|\}. \tag{B.54}$$

Therefore the total error for the product formula may be bounded as

$$\left\| S^{(m)}(t) - e^{tW} \right\| \leq \frac{(\zeta t\Lambda)^{2k+1}}{(2k+1)!} \exp\left(\xi t\Lambda\right) + \frac{(t\|W\|)^{2k+1}}{(2k+1)!} \exp\left(t\|W\|\right). \tag{B.55}$$

It can be seen from this expression that the total evolution time ζ appears in this bound and in the simulations of Section 6.4 the other quantities in the bound such as the norm of the operators are constant, then we can expect ξ to be correlated with the constant factor in the error.

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, pages 333–342. ACM, 2011, arXiv:1011.3245. doi:10.1145/1993636.1993682.

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019. doi:10.1038/s41586-019-1666-5.

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2):281–298, Aug 2014. doi:10.1007/s00224-013-9527-3.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009.

[ABNO22] Srinivasan Arunachalam, Sergey Bravyi, Chinmay Nirkhe, and Bryan O'Gorman. The Parametrized Complexity of Quantum Verification. In François Le Gall and Tomoyuki Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics*

(LIPIcs), pages 3:1–3:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2022.3.

[AHKK17] Winfried Auzinger, Harald Hofstätter, David Ketcheson, and Othmar Koch. Practical splitting methods for the adaptive integration of nonlinear evolution equations. part i: Construction of optimized schemes and pairs of schemes. *BIT Numerical Mathematics*, 57(1):55–74, 2017. doi:10.1007/s10543-016-0626-9.

[APMB20] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability deficits in quantum approximate optimization. *Phys. Rev. Lett.*, 124:090504, Mar 2020. doi:10.1103/PhysRevLett.124.090504.

[AS17] Guillaume Aubrun and Stanisław J Szarek. *Alice and Bob meet Banach: the interface of asymptotic geometric analysis and quantum information theory*. Mathematical surveys and monographs. American Mathematical Society, Providence, RI, 2017. URL https://cds.cern.ch/record/2296008.

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 20–29, New York, NY, USA, 2003. Association for Computing Machinery. doi:10.1145/780542.780546.

[BACS07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, Mar 2007. doi:10.1007/s00220-006-0150-x.

[BBC+95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald

Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995. doi:10.1103/PhysRevA.52.3457.

[BC04] Sergio Blanes and Fernando Casas. On the convergence and optimization of the baker–campbell–hausdorff formula. *Linear Algebra and its Applications*, 378:135–158, 2004. doi:https://doi.org/10.1016/j.laa.2003.09.010.

[BC12] Dominic W. Berry and Andrew M. Childs. Black-box hamiltonian simulation and unitary implementation. *Quantum Info. Comput.*, 12(1–2):29–62, jan 2012.

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 283–292, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591854.

[BCC+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015. doi:10.1103/PhysRevLett.114.090502.

[BCCM13] S. Blanes, F. Casas, P. Chartier, and A. Murua. Optimized high-order splitting methods for some classes of parabolic equations. *Mathematics of Computation*, 82(283):1559–1576, 2013. URL http://www.jstor.org/stable/42002709.

[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, 2015. doi:10.1109/FOCS.2015.54.

[BCM06] S. Blanes, F. Casas, and A. Murua. Composition methods for differential equations with processing. *SIAM Journal on Scientific Computing*, 27(6):1817–1843, 2006, arXiv:https://doi.org/10.1137/030601223. doi:10.1137/030601223.

[BCM08] Sergio Blanes, Fernando Casas, and Ander Murua. Splitting and composition methods in the numerical integration of differential equations. *arXiv e-prints*, page arXiv:0812.0377, December 2008, arXiv:0812.0377. doi:10.48550/arXiv.0812.0377.

[BFK18] Adam Bouland, Joseph F. Fitzsimons, and Dax Enshan Koh. Complexity Classification of Conjugated Clifford Circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference (CCC 2018)*, volume 102 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:25, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2018.21. ISSN: 1868-8969.

[BFLL22] Adam Bouland, Bill Fefferman, Zeph Landau, and Yunchao Liu. Noise and the frontier of quantum supremacy. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1308–1317, 2022. doi:10.1109/FOCS52979.2021.00127.

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, Feb 2019. doi:10.1038/s41567-018-0318-2.

[BGKS16] Alex Bredariol Grilo, Iordanis Kerenidis, and Jamie Sikora. QMA with Subset State Witnesses. *Chicago Journal of Theoretical Computer Science*, 2016(4):1–17, March 2016. doi:10.4086/cjtcs.2016.004.

[BIS+18] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing Quantum Supremacy in Near-Term Devices. *Nature Physics*, 14:595–600, Jul 2018, arXiv:1608.00263.

[BJL+22] Michael J. Bremner, Zhengfeng Ji, Xingjian Li, Luke Mathieson, and Mauro E. S. Morales. Parameterized complexity of weighted local hamiltonian problems and the quantum exponential time hypothesis, 2022. doi:10.48550/ARXIV.2211.05325.

[BJM+22] Michael J. Bremner, Zhengfeng Ji, Ryan L. Mann, Luke Mathieson, Mauro E. S. Morales, and Alexis T. E. Shaw. Quantum parameterized complexity, 2022. doi:10.48550/ARXIV.2203.08002.

[BJS11] M. J. Bremner, R. Jozsa, and D. J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. R. Soc. A*, 467(2126):459, 2011. doi:10.1098/rspa.2010.0301.

[BK02] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic Quantum Computation. *Annals of Physics*, 298(1):210–226, May 2002. doi:10.1006/aphy.2002.6254.

[BL08] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008. doi:10.1103/PhysRevA.78.012352.

[BMK20] Jacob D. Biamonte, Mauro E. S. Morales, and Dax Enshan Koh. Entanglement scaling in quantum advantage benchmarks. *Phys. Rev. A*, 101:012349, Jan 2020. doi:10.1103/PhysRevA.101.012349.

[BMS16] M. Bremner, A. Montanaro, and D. Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Phys. Rev. Lett.*, 117:080501, 2016. doi:10.1103/PhysRevLett.117.080501.

[BMW+15] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Physical Review A*, 91:022311, Feb 2015. doi:10.1103/PhysRevA.91.022311.

[Boo12] Adam D. Bookatz. QMA-complete problems. *arXiv e-prints*, page arXiv:1212.6312, December 2012, arXiv:1212.6312. doi:10.48550/arXiv.1212.6312.

[Bra06] Sergey Bravyi. Universal quantum computation with the $\nu = 5/2$ fractional quantum Hall state. *Phys. Rev. A*, 73(4):042313, Apr 2006. doi:10.1103/PhysRevA.73.042313.

[Bro15] Daniel J. Brod. Complexity of simulating constant-depth bosonsampling. *Phys. Rev. A*, 91:042316, Apr 2015. doi:10.1103/PhysRevA.91.042316.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997, arXiv:https://doi.org/10.1137/S0097539796300921. doi:10.1137/S0097539796300921.

[BVHS+18] J. Bermejo-Vega, D. Hangleiter, M. Schwarz, R. Raussendorf, and J. Eisert. Architectures for quantum simulation showing a quantum speedup. *Phys. Rev. X*, 8(2):021010, April 2018. doi:10.1103/PhysRevX.8.021010. arXiv:1703.00466.

[Cam19] Earl Campbell. Random compiler for fast hamiltonian simulation. *Phys. Rev. Lett.*, 123:070503, Aug 2019. doi:10.1103/PhysRevLett.123.070503.

[CB12] Andrew M. Childs and Dominic W. Berry. Black-box hamiltonian simulation and unitary implementation. *Quantum Information and Computation*, 12(1&2):29–62, January 2012. doi:10.26421/qic12.1-2-4.

[CCZZ21] Nai-Hui Chia, Chi-Ning Chou, Jiayu Zhang, and Ruizhe Zhang. Quantum meets the minimum circuit size problem, 2021, arXiv:2108.03171.

[CFW22] Chris Cade, Marten Folkertsma, and Jordi Weggemans. Complexity of the Guided Local Hamiltonian Problem: Improved Parameters and Extension to Excited States. *arXiv e-prints*, page arXiv:2207.10097, July 2022, arXiv:2207.10097.

[CK11] Andrew M. Childs and Robin Kothari. Simulating sparse hamiltonians with star decompositions. In Wim van Dam, Vivien M. Kendon, and Simone Severini, editors, *Theory of Quantum Computation, Communication, and Cryptography*, pages 94–103, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[CM16] Toby Cubitt and Ashley Montanaro. Complexity classification of local hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016, arXiv:https://doi.org/10.1137/140998287. doi:10.1137/140998287.

[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.

[COS19] Andrew M. Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, September 2019. doi:10.22331/q-2019-09-02-182.

[CR92] Don Coppersmith and T. J. Rivlin. The growth of polynomials bounded at equally spaced points. *SIAM Journal on Mathematical Analysis*, 23(4):970–983, 1992, arXiv:https://doi.org/10.1137/0523054. doi:10.1137/0523054.

[CST+21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11:011020, Feb 2021. doi:10.1103/PhysRevX.11.011020.

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

[Deu85] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, July 1985. doi:10.1098/rspa.1985.0070.

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer London, London, 2013. doi:10.1007/978-1-4471-5559-1.

[DGGJ00] Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, and Mark Jerrum. On the relative complexity of approximate counting problems. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '00, page 108–119, Berlin, Heidelberg, 2000. Springer-Verlag.

[DHJBa22] Alexander M. Dalzell, Nicholas Hunter-Jones, and Fernando G. S. L. Brandão. Random quantum circuits anticoncentrate in log depth. *PRX Quantum*, 3:010333, Mar 2022. doi:10.1103/PRXQuantum.3.010333.

[DKD20] Ignacy Duleba and Iwona Karcz-Duleba. Algorithm to express lie monomials in ph. hall basis and its practical applications. In Roberto Moreno-Díaz, Franz Pichler, and Alexis Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2019*, pages 465–473, Cham, 2020. Springer International Publishing.

[DT05] David P. DiVincenzo and Barbara M. Terhal. Fermionic linear optics revisited. *Foundations of Physics*, 35(12):1967–1984, Dec 2005. doi:10.1007/s10701-005-8657-0.

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

[Fey86] Richard P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, Jun 1986. doi:10.1007/BF01886518.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer Berlin Heidelberg, 2006. doi:10.1007/3-540-29953-x.

[FL87] Michael R Fellows and Michael A Langston. Nonconstructive advances in polynomial-time complexity. *Information Processing Letters*, 26(3):157–162, 1987.

[FU16] Bill Fefferman and Christopher Umans. On the Power of Quantum Fourier Sampling. In Anne Broadbent, editor, *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, volume 61 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:19, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TQC.2016.1.

[GHLM22] Sevag Gharibian, Ryu Hayakawa, François Le Gall, and Tomoyuki Morimae. Improved Hardness Results for the Guided Local Hamiltonian Problem. *arXiv e-prints*, page arXiv:2207.10250, July 2022, arXiv:2207.10250.

[Gil19] András Gilyén. *Quantum singular value transformation & its algorithmic applications*. PhD thesis, University of Amsterdam, 2019.

[GL21] Sevag Gharibian and François Le Gall. Dequantizing the Quantum Singular Value Transformation: Hardness and Applications to Quantum Chemistry

and the Quantum PCP Conjecture. *arXiv e-prints*, page arXiv:2111.09079, November 2021, arXiv:2111.09079.

[Gol08] Oded Goldreich. Computational complexity: a conceptual perspective. *ACM Sigact News*, 39(3):35–39, 2008.

[Hal13] B.C. Hall. *Quantum Theory for Mathematicians*. Graduate Texts in Mathematics. Springer New York, 2013. URL https://books.google.com.au/books?id=bYJDAAAAQBAJ.

[HJK+19] M. Hebenstreit, R. Jozsa, B. Kraus, S. Strelchuk, and M. Yoganathan. All pure fermionic non-gaussian states are magic states for matchgate computations. *Phys. Rev. Lett.*, 123:080503, Aug 2019. doi:10.1103/PhysRevLett.123.080503.

[HJKS20] Martin Hebenstreit, Richard Jozsa, Barbara Kraus, and Sergii Strelchuk. Computational power of matchgates with supplementary resources. *arXiv e-prints*, page arXiv:2007.08231, July 2020, arXiv:2007.08231.

[HS05] Peter Hoyer and Robert Spalek. Quantum fan-out is powerful. *Theory of Computing*, 1(1):81–103, 2005. doi:10.4086/toc.2005.v001a005.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:https://doi.org/10.1006/jcss.2000.1727.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:https://doi.org/10.1006/jcss.2001.1774.

[Iva17] Dmitri A. Ivanov. Computational complexity of exterior products and multiparticle amplitudes of noninteracting fermions in entangled states. *Phys. Rev. A*, 96:012322, Jul 2017. doi:10.1103/PhysRevA.96.012322.

[JM08] Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464(2100):3089–3106, July 2008. doi:10.1098/rspa.2008.0189.

[KKR05] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, pages 372–383, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[KL97] William Kahan and Ren-Cang Li. Composition constants for raising the orders of unconventional schemes for ordinary differential equations. *Math. Comput.*, 66(219):1089–1099, jul 1997. doi:10.1090/S0025-5718-97-00873-9.

[KMM22] Y. Kondo, R. Mori, and R. Movassagh. Quantum supremacy and hardness of estimating output probabilities of quantum circuits. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1296–1307, Los Alamitos, CA, USA, feb 2022. IEEE Computer Society. doi:10.1109/FOCS52979.2021.00126.

[KR03] Julia Kempe and Oded Regev. 3-local hamitonian is qma-complete. *Quantum Info. Comput.*, 3(3):258–264, may 2003.

[Kro22] Hari Krovi. Average-case hardness of estimating probabilities of random quantum circuits with a linear scaling in the error exponent. *arXiv e-prints*, page arXiv:2206.05642, June 2022, arXiv:2206.05642. doi:10.48550/arXiv.2206.05642.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and Quantum Computation*. Number 47 in Graduate Studies in Mathematics. American Mathematical Society, 2002. doi:10.1090/gsm/047.

[Kul97] Wladyslaw Kulpa. The poincaré-miranda theorem. *The American Mathematical Monthly*, 104(6):545–550, 1997. URL http://www.jstor.org/stable/2975081.

[Lau83] Clemens Lautemann. Bpp and the polynomial hierarchy. *Information Processing Letters*, 17(4):215 – 217, 1983. doi:https://doi.org/10.1016/0020-0190(83)90044-3.

[LC17a] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017. doi:10.1103/PhysRevLett.118.010501.

[LC17b] Guang Hao Low and Isaac L Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118(1):010501, 2017. doi:10.1103/PhysRevLett.118.010501.

[LCV07] Yi-Kai Liu, Matthias Christandl, and F. Verstraete. Quantum computational complexity of the *n*-representability problem: Qma complete. *Phys. Rev. Lett.*, 98:110503, Mar 2007. doi:10.1103/PhysRevLett.98.110503.

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

[Low19] Guang Hao Low. Hamiltonian simulation with nearly optimal dependence on spectral norm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 491–502, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316386.

[LSTT22] Guang Hao Low, Yuan Su, Yu Tong, and Minh C. Tran. On the complexity of implementing trotter steps. *arXiv:2211.09133*, 2022. URL https://arxiv.org/abs/2211.09133.

[MBZ20] M. E. S. Morales, J. D. Biamonte, and Z. Zimborás. On the universality of the quantum approximate optimization algorithm. *Quantum Information Processing*, 19(9):291, Aug 2020. doi:10.1007/s11128-020-02748-9.

[MCB+22] Mauro E. S. Morales, Pedro C. S. Costa, Daniel K. Burgarth, Yuval R. Sanders, and Dominic W. Berry. Greatly improved higher-order product formulae for quantum simulation, 2022. doi:10.48550/ARXIV.2210.15817.

[Mor17] Tomoyuki Morimae. Hardness of classically sampling the one-clean-qubit model with constant total variation distance error. *Phys. Rev. A*, 96:040302, Oct 2017. doi:10.1103/PhysRevA.96.040302.

[Mov19] Ramis Movassagh. Quantum supremacy and random circuits. *arXiv e-prints*, page arXiv:1909.06210, Sep 2019, arXiv:1909.06210.

[MS83] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Pub. Co., 1983.

[MTB18] Mauro E. S. Morales, Timur Tlyachev, and Jacob Biamonte. Variational learning of grover's quantum search algorithm. *Phys. Rev. A*, 98:062333, Dec 2018. doi:10.1103/PhysRevA.98.062333.

[MW05] Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005, arXiv:cs/0506068. doi:10.1007/s00037-005-0194-x.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA. arXiv:0904.1549, 2009.

[ODMZ22] Michał Oszmaniec, Ninnat Dangniam, Mauro E.S. Morales, and Zoltán Zimborás. Fermion sampling: A robust quantum computational advantage scheme using fermionic linear optics and magic input states. *PRX Quantum*, 3:020328, May 2022. doi:10.1103/PRXQuantum.3.020328.

[OIWF22] Bryan O'Gorman, Sandy Irani, James Whitfield, and Bill Fefferman. Intractability of electronic structure in a fixed basis. *PRX Quantum*, 3:020322, May 2022. doi:10.1103/PRXQuantum.3.020322.

[OT08] Roberto Oliveira and Barbara M. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *arXiv:quant-ph/0504050*, 2008.

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 468–474, New York, NY, USA, 1992. Association for Computing Machinery. doi:10.1145/129712.129758.

[PBG20] Hakop Pashayan, Stephen D. Bartlett, and David Gross. From estimation of quantum probabilities to simulation of quantum circuits. *Quantum*, 4:223, January 2020. doi:10.22331/q-2020-01-13-223.

[PM17] Stephen Piddock and Ashley Montanaro. The complexity of antiferromagnetic interactions and 2d lattices. *Quantum Info. Comput.*, 17(7–8):636–672, jun 2017.

[Ser12] Jean-Pierre Serre. *Linear representations of finite groups*. Graduate Texts in Mathematics. Springer, New York, NY, 1977 edition, December 2012.

[SHC21] Yuan Su, Hsin-Yuan Huang, and Earl T. Campbell. Nearly tight Trotterization of interacting electrons. *Quantum*, 5:495, July 2021. doi:10.22331/q-2021-07-05-495.

[SS05] Mark Sofroniou and Giulia Spaletta. Derivation of symmetric composition constants for symmetric integrators. *Optimization Methods and Software*, 20(4-5):597–613, 2005, arXiv:https://doi.org/10.1080/10556780500140664. doi:10.1080/10556780500140664.

[Sto85] L. Stockmeyer. On approximation algorithms for #P. *SIAM J. Comput.*, 14:849–861, 1985. doi:https://doi.org/10.1137/0214060.

[Suz90] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(6):319–323, 1990. doi:10.1016/0375-9601(90)90962-N.

[Suz91] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991. doi:10.1063/1.529425.

[TD02] Barbara M. Terhal and David P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A*, 65:032325, Mar 2002. doi:10.1103/PhysRevA.65.032325.

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, October 1991. doi:10.1137/0220053.

[Val01] Leslie G. Valiant. Quantum computers that can be simulated classically in polynomial time. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 114–123, New York, NY, USA, 2001. Association for Computing Machinery. doi:10.1145/380752.380785.

[VV16] Alexander Van-Brunt and Matt Visser. Simplifying the Reinsch algorithm for the Baker-Campbell-Hausdorff series. *Journal of Mathematical Physics*, 57(2):023507, February 2016, arXiv:1501.05034. doi:10.1063/1.4939929.

[Wat08] John Watrous. Quantum Computational Complexity. *arXiv e-prints*, page arXiv:0804.3401, April 2008, arXiv:0804.3401. doi:10.48550/arXiv.0804.3401.

[WLAG13] James Daniel Whitfield, Peter John Love, and Alán Aspuru-Guzik. Computational complexity in electronic structure. *Phys. Chem. Chem. Phys.*, 15:397–411, 2013. doi:10.1039/C2CP42695A.

[WMN10] Tzu-Chieh Wei, Michele Mosca, and Ashwin Nayak. Interacting boson problems can be qma hard. *Phys. Rev. Lett.*, 104:040501, Jan 2010. doi:10.1103/PhysRevLett.104.040501.

[YJS19] Mithuna Yoganathan, Richard Jozsa, and Sergii Strelchuk. Quantum advantage of unitary clifford circuits with magic state inputs. *Proceedings of the Royal Society A*, 475(2225):20180427, 2019.

[Yos90] Haruo Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5):262–268, 1990. doi:10.1016/0375-9601(90)90092-3.

[Zha12] Chi Zhang. Randomized algorithms for hamiltonian simulation. In Leszek Plaskota and Henryk Woźniakowski, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 709–719, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[ZRB23] Sergiy Zhuk, Niall Robertson, and Sergey Bravyi. Trotter error bounds and dynamic multi-product formulas for Hamiltonian simulation. *arXiv e-prints*, page arXiv:2306.12569, June 2023, arXiv:2306.12569. doi:10.48550/arXiv.2306.12569.