Quantum Simulation Algorithms with Applications to Quantum Error Correction and Topological Data Analysis

Abhishek Rajput

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Nathan Wiebe, Chair

Martin Savage, Chair

Lukasz Fidkowski

Boris Blinov

Program Authorized to Offer Degree:
Department of Physics

University of Washington

Abstract

Quantum Simulation Algorithms with Applications to Quantum Error Correction and Topological Data Analysis

Abhishek Rajput

Co-Chairs of the Supervisory Committee:
Nathan Wiebe
Department of Physics

Martin Savage
Department of Physics

This thesis presents three results of relevance to the simulation of quantum dynamics on quantum computers. The first is the development of hybridized methods of quantum simulation with Alessandro Roggero and Nathan Wiebe [1]. Conventional methods of quantum simulation, such as qDRIFT, Trotterization, and qubitization, involve trade-offs that limit their applicability to specific contexts where their use is optimal. We develop a framework that allows different simulation methods to be hybridized via the interaction picture and thereby improve performance for interaction picture simulations over known algorithms. These approaches show asymptotic improvements over the individual methods that comprise them and further make interaction picture simulation methods practical in the near term. Physical applications of these hybridized methods yield a gate complexity scaling as $\log^2 \Lambda$ in the electric cutoff Λ for the Schwinger model and independent of the electron density for collective neutrino oscillations, outperforming the scaling for all current algorithms with these parameters. For the general problem of Hamiltonian simulation subject to dynamical constraints, these methods yield a query complexity independent of the penalty parameter λ used to impose an energy cost on time-evolution into an unphysical subspace.

Our second result concerns fault-tolerant error correction procedures [2] developed with Alessandro Roggero and Nathan Wiebe for certain lattice gauge theories (LGTs) with a view towards simulating their dynamics on quantum computers. Quantum simulations of LGTs are often formulated on an enlarged Hilbert space containing both physical and unphysical sectors in order to retain a local Hamiltonian. We provide simple fault-tolerant procedures that exploit such redundancy by combining a phase flip error correction code with the intrinsic Gauss' law gauge symmetry to correct one-qubit errors for a $\mathbb{Z}_2$ or truncated U(1) LGT in 1+1 and 2+1 dimensions with a link flux cutoff of 1. Unlike existing work on detecting violations of Gauss' law, our circuits are fault tolerant and the overall error correction scheme outperforms a naïve application of the $[5, 1, 3]$ code. The constructions outlined can be extended to LGT systems with larger cutoffs and can be used in understanding how to hybridize error correction and quantum simulation for LGTs in higher space-time dimensions and with different symmetry groups.

Lastly, we present a classical randomized algorithm developed with Nathan Wiebe for topological data analysis [3], the problem of determining the approximate Betti numbers of simplicial complexes constructed from data sets. This problem has attracted considerable interest in recent years due to claims of exponential advantage for this task on quantum computers over classical methods such as Gaussian elimination or the Lanczos algorithm that scale exponentially with the size of the input. Our algorithm demonstrates a partial dequantization of this problem via the classical simulation of the imaginary time-evolution of the combinatorial Laplacian with the path integral Monte Carlo method. We show this algorithm can extract approximate Betti numbers in polynomial time in some of the same regimes where quantum computers were claimed to provide exponential advantage over classical computers. This implies that having exponentially large dimension and Betti number are necessary but not sufficient conditions for super-polynomial advantage on quantum computers.

TABLE OF CONTENTS

Page

List of Figures . . . iii

List of Tables . . . iv

Chapter 1: Introduction . . . 1
1.1 Quantum Simulation . . . 4
1.1.1 Continuous qDRIFT . . . 7
1.1.2 Qubitization and Quantum Singular Value Transformations . . . 10
1.1.3 Trotterization . . . 18
1.1.4 Interaction Picture . . . 20
1.2 Error Correction of Abelian Lattice Gauge Theories . . . 23
1.2.1 Structure of Abelian Lattice Gauge Theories . . . 24
1.2.2 Repetition Codes . . . 27
1.3 Topological Data Analysis . . . 31
1.3.1 Simplicial Homology . . . 33
1.3.2 Computational Complexity of Topological Data Analysis . . . 41

Chapter 2: Hybridized Methods of Quantum Simulation in the Interaction Picture 44
2.1 Hybrid Trotterization and qDRIFT Protocol . . . 44
2.2 Hybrid Continuous qDRIFT and Qubitization Protocol . . . 50
2.3 Hamiltonian Simulation of Schwinger Model . . . 56
2.3.1 Schwinger Model and Query Complexity Bounds . . . 56
2.3.2 Construction of Prepare and Select Oracles . . . 62
2.3.3 Gate Complexity Analysis . . . 65
2.3.4 Comparison with Trotterization . . . 68
2.4 Collective Neutrino Oscillations . . . 70
2.4.1 Trotter Suzuki Approximations in Interaction Frame . . . 72

2.4.2 Simulating Neutrino Oscillations using Hybrid Trotter-qDRIFT . . . 74
2.5 Constrained Hamiltonian Dynamics . . . 75
2.6 Conclusions . . . 83

Chapter 3: Quantum Error Correction with Gauge Symmetries . . . 87
3.1 Integrating the repetition code with Gauss' Law . . . 87
3.2 Applications to 1D Lattice Gauge Theories . . . 92
3.2.1 Error Correction for Pure Gauge Theory . . . 93
3.2.2 Error Correction for Non-Dynamical Fermions . . . 95
3.2.3 Error Correction for Dynamical Fermions . . . 96
3.3 Extension to two dimensions . . . 100
3.4 Conclusion . . . 108

Chapter 4: Betti Number Estimation with Path Integral Monte Carlo . . . 109
4.1 Dequantization using path integral Monte Carlo . . . 114
4.1.1 Metropolis Hastings algorithm . . . 119
4.1.2 Trotter error in path integration . . . 122
4.1.3 Sample bounds . . . 124
4.2 Analysis . . . 126
4.3 Conclusion . . . 130

Chapter 5: Concluding Remarks . . . 133
5.1 Hybridized Quantum Algorithms in the Interaction Picture . . . 133
5.2 Error Correction with Gauge Symmetries . . . 134
5.3 Quantum and Classical Algorithms for Topological Data Analysis . . . 136

Appendix A: Diamond Norm . . . 156

Appendix B: Notation for qDRIFT . . . 158

Appendix C: Complexity Classes . . . 160

LIST OF FIGURES

Figure Number Page

1.1 Controlled-walk operator 14
1.2 Alternating phase modulation sequence 16
1.3 Fractional reflection gadget 17
1.4 Comparison between error-correcting schemes 25
1.5 Bit-flip logical state 28
1.6 Full bit-flip error correcting circuit 29
1.7 Equivalent circuit for bit-flip stabilizer measurement 30
1.8 Phase-flip logical state 31
1.9 Full circuit for phase-flip error correction 31
1.10 Examples of simplices 34
1.11 Example of simplicial complex 35
1.12 Example of a boundary and a cycle 37

2.1 SELECT circuit for Schwinger model hopping term 63
2.2 PREPARE circuit for Schwinger model hopping term 64

3.1 Logical-to-logical CNOT equivalence 92
3.2 Pure gauge theory error correction 94
3.3 Fault-tolerant logical-to-physical CNOT 95
3.4 Error correction for non-dynamical fermions 97
3.5 Error correction for dynamical fermions 98
3.6 Extended Gauss' law checks 100
3.7 Link convention around fermionic site in a 2D lattice gauge theory 101
3.8 Error correction schematic for 2D abelian LGT 102

LIST OF TABLES

Table Number Page

1.1 Bit-flip syndrome outcomes 29

2.1 Cost comparison for hybrid simulation protocols 86

3.1 Error localization for dynamical fermions 99

3.2 Error localization for extended Gauss' law checks 101

3.3 Syndrome measurement outcomes in a 2D abelian LGT 104

3.4 Error localization around single site for 2D abelian LGT 106

3.5 Error localization around links and sites opposite to ancillas 107

ACKNOWLEDGMENTS

I extend my sincerest gratitude to my supervisor Nathan Wiebe for his considerable support and guidance, without which this PhD would not have been possible. I am fortunate to have worked with and learned from my collaborator Alessandro Roggero on several research projects throughout the course of my degree. I would also like to thank Martin Savage for many valuable discussions and insights.

I thank the Department of Physics at the University of Washington and the InQubator for Quantum Simulation for providing a vibrant community with numerous talented people to discuss research with and learn from including Kade Cichella, Tanner Rase, Natalie Klco, John Goldak, Roland Farrell, Jesse Stryker, and Joseph Merritt. I am also grateful to the University of Toronto for the support they provided me as an international visiting student during the last two years of my PhD. I would also like to thank Dylan Airey for his insight and patience for the many mathematics questions I've asked over the years.

I acknowledge the financial support received through a research grant from Google which made the research in this thesis possible.

Chapter 1

INTRODUCTION

The field of quantum computation arose in part to answer the question of what computational and information processing tasks can be performed using quantum mechanical systems as computational devices, i.e. using "quantum computers". The first rigorous notion of a universal quantum computer, analogous to the universal Turing machine in classical computing, was developed by David Deutsch in 1985 [4]. Such an idealized quantum computer is universal in the sense that it is capable of simulating any finite dimensional and physically realizable system with arbitrarily high accuracy.

The fundamental unit of information for this model is the qubit. While a classical bit can be in one of two possible states, 0 or 1, a qubit can be thought of mathematically as a unit vector in $\mathbb{C}^2$ and can exist in a linear superposition of the $|0\rangle$ and $|1\rangle$ basis vectors. The role of "logical gates" implemented by the quantum computer acting on a qubit or collections of qubits is played by unitary operators, denoted by $\mathrm{U}(2^n)$ with n the number of qubits. An especially important set of single and two-qubit unitary operators are the Hadamard, phase, $\pi/8$, and CNOT gates. The Hadamard gate, denoted by H, can be thought of as a two-dimensional quantum Fourier transform on a single qubit and has the action $H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $H|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. The phase gate S applies a phase of i when acting on the $|1\rangle$ computational basis state and the $\pi/8$ gate T is, up to an unimportant global phase, a rotation about the $\hat{z}$-axis by $\pi/4$. The two-qubit CNOT gate has the action $\mathrm{CNOT}|a\rangle|b\rangle = |a\rangle|a \oplus b\rangle$, where the $\oplus$ denotes a bit-wise sum. In other words, the CNOT gate flips the state of the second qubit if the state of the first qubit is $|1\rangle$. Although there are an uncountably infinite number of possible unitary operators in $\mathrm{U}(2^n)$, any such operator can be approximated to a desired level of accuracy by taking finite sequences of these operators

or tensor products of them. The notion of approximation here is with respect to a distance function induced by the choice of a norm on $U(2^n)$, such as the Frobenius norm, and it is in this sense that a universal quantum computer implementing a gate set like the above can simulate any finite dimensional, physically realizable system.

This computational model would only be of theoretical interest unless computational tasks that can be performed more efficiently on quantum computers than on classical computers are identified. The first such task to be proposed in a seminal work by Richard Feynmann [5], even before Deutsch's rigorous notion of a universal quantum computer, was the simulation of the dynamics of quantum mechanical systems on quantum computers. Much of the interest in this problem was driven by the fact that simulating the dynamics of quantum systems on classical computers generically requires significant computational resources, often scaling linearly with the dimension of the Hilbert space of the system in question [4, 5, 6]. Since the dimension of the Hilbert space in turn scales exponentially in the number of particles in the system, the resulting classical simulation can often be intractable. Considerable research has therefore been undertaken on the problem of quantum simulation for a wide variety of Hamiltonians, as it is a major area where quantum computers are expected to outperform even classical supercomputers [6, 7, 8, 9, 10].

Quantum simulation algorithms approximate the unitary time-evolution under a time-dependent or time-independent Hamiltonian via a finite sequence of simpler unitary operations that can be performed on a quantum computer. They therefore encompass a broad class of problems relevant for simulating physical systems ranging from complex molecules to quantum field theories given a suitable input Hamiltonian [9, 7]. Applications extend even beyond the simulation of physical systems to other mathematical problems, and one especially important example is the Harrow-Hassidim-Lloyd (HHL) quantum algorithm for solving linear systems of equations, which are ubiquitous throughout the sciences [11]. This algorithm exploits the simulation of the unitary time-evolution of a Hermitian matrix corresponding to a linear system of equations to compute the expectation value of an operator with respect to the solution vector, and has a run time exponentially faster than the best

corresponding classical algorithm in the number of variables in the linear system.

One might nonetheless expect the simulation of unitary time-evolution operators to reflect only a portion of the space of all decision problems that can be solved, with probability greater than 2/3, on a quantum computer using a polynomially large number of operations. This class is denoted by **BQP** in the field of computational complexity theory, which concerns itself with the classification of classical and quantum computational problems. Several problems that are believed to have no efficient implementation on classical computers, such as the integer factorization and the discrete logarithm problem, have been proved to lie in **BQP** [11, 12] and it is not immediately obvious how one might frame these in terms of the time-evolution of a suitable Hamiltonian. It has been proved however that the problem of approximating the unitary time-evolution of a Hamiltonian is in fact **BQP**-complete [6], which essentially implies that any problem in **BQP** can be efficiently reduced to this one. Quantum simulation algorithms are therefore not just of considerable practical interest but of broader theoretical significance to the field of quantum computation.

This perspective on the close relationship between quantum simulation algorithms and a larger class of quantum computation problems is a major emphasis of this thesis. Chapter 2 introduces new hybridized quantum simulation algorithms developed in a joint work with Alessandro Roggero, and Nathan Wiebe [1] that exploit the interaction picture to yield improvements in simulating certain classes of Hamiltonians such as for the Schwinger model, collective neutrino oscillations, and for constrained dynamics. Rigorous error bounds are proven for algorithms that use various combinations of these methods, and regimes where some of these algorithms might outperform others are analyzed in terms of generic properties of the input Hamiltonian.

In Chapter 3, we consider the problem of how the quantum simulation of lattice gauge theories, an important stepping stone for the simulation of quantum field theories on quantum computers, can time-evolve an initial state into unphysical sectors of the Hibert space where gauge constraints are violated. In a work with Alessandro Roggero, and Nathan Wiebe [2], we address this by demonstrating how single qubit errors occurring in simple 1+1 and 2+1

dimensional lattice systems with truncated U(1) or $\mathbb{Z}_2$ gauge groups can be corrected fault tolerantly by exploiting the intrinsic Gauss' law gauge symmetry present in such systems. The resulting algorithms require less qubits than an application of the best single-qubit error correcting code, the $[5, 1, 3]$ code, to every qubit of the system and provide a pathway to integrating error correction protocols with quantum simulation circuits that require less qubits.

Lastly, as the determination of what problems can or cannot be efficiently performed on a quantum computer is critical to assessing the power of quantum over classical computation, we consider the problem of topological data analysis on quantum computers in Chapter 4. This problem is concerned with extracting the Betti numbers of simplicial complexes constructed from input data sets. A recently proposed quantum algorithm claimed to find all the Betti numbers of simplicial complexes in a run-time exponentially faster than the best classical algorithms by applying quantum algorithms such as phase estimate and Grover's search on an object called the combinatorial Laplacian [13]. We address this claim by developing in a joint work with Nathan Wiebe [3] a classical algorithm that performs the imaginary time-evolution of the combinatorial Laplacian using the path integral Monte Carlo method and show it can efficiently determine the Betti numbers of simplicial complexes, under certain conditions, in some of the same regimes where quantum algorithms were initially believed to provide an advantage over classical computers [14, 15]. This not only demonstrates the broader utility of quantum simulation methods in analyzing other types of comptuational problems but adds to an existing body of work that shows the conditions under which quantum computational problems achieve a speedup over their classical counterparts are more subtle than initially surmised.

1.1 Quantum Simulation

The problem of quantum simulation is as follows: given a $2^n \times 2^n$ Hermitian matrix H with n the number of qubits, an evolution time t, and an error tolerance ϵ, generate a sequence of unitary operations $U_i(t)$ such that $\| \prod_i U_i(t) - e^{-iHt} \| \leq \epsilon$ for an appropriate norm $\| \cdot \|$ on the

Lie group of unitary matrices $\mathrm{U}(2^n)$ and such that the computational cost of the sequence of unitary operations in terms of a universal gate set is minimal. It is assumed that we do not explicitly know the matrix elements of e^{-iHt}, so it is necessary to construct this unitary only using information about the Hermitian matrix H.

Connected to the approximation of the time-evolution operator is the mapping between the states $|\psi\rangle$ of the physical system being "simulated" and the states in the quantum computer. The states of the latter are typically expressed in terms of two-level systems, i.e. qubits, or tensor products of them. Thus given a quantum computer with n qubits, a state of the quantum computer $|\psi\rangle_c$ is a vector in $\mathbb{C}^{2^n}$. The state of the physical system $|\psi\rangle$ must be mapped onto the states of the quantum computer via some injective mapping ϕ[1]. The approximation of the ideal time-evolution e^{-iHt} under a Hermitian H then acts on $\phi(|\psi\rangle)$. This results in the state $\prod_i U_i(\phi(|\psi\rangle))$ and thus $\phi^{-1}(\prod_i U_i(\phi(|\psi\rangle)))$ is an approximation of $e^{-iHt}|\psi\rangle$ in the physical system. It is in this sense that the quantum device is said to "simulate" the original system. The map ϕ however will greatly depend on the physical system in question and the particular experimental realization of the quantum computer. We mainly focus on the problem of approximating time-evolution operators.

A variety of simulation methods have been developed over the years to approximate the ideal time-evolution operator or the corresponding quantum channel on density matrices. The oldest ones are the Trotter-Suzuki formulas and their time-ordered generalizations [6, 16, 17, 18, 19]. These involve decomposing the true time-evolution operator into products of time-evolution operators involving the summands of the full Hamiltonian $H = \sum_j H_j$. Their error scaling with respect to the simulation time t, error ϵ, and the norms of the commutators $[H_i, H_j]$ have been extensively investigated, e.g. Refs. [16, 19, 20]. Analogous recursive formulas for the decomposition of the path-ordered exponential of a time-dependent Hamiltonian also exist [17], but their scaling with respect to t, ϵ, and other properties of the

[1]Such a map will need to preserve additional algebraic structure depending on the physical system under consideration. One example is anti-commutation or commutation relations for fermionic or bosonic systems respectively.

Hamiltonian is less understood. All these formulas are space efficient in the sense that they do not require additional qubits beyond those needed to specify the system. The asymptotic scaling of such formulas however often comes with large constant factors and a dependence on the number of terms in the Hamiltonian [19]. Smaller time steps are also required for the more accurate higher order formulas, which translates to requiring larger depth quantum circuits.

Recent years have seen several additions to the repertoire of quantum simulation techniques. The method of qubitization [21, 22, 23, 24, 25, 26, 27] involves the implementation of a certain "walk" operator whose eigenvalues are an efficiently computable function of those of H and achieves linear scaling in the simulation time t, logarithmic scaling in the inverse error tolerance ϵ^{-1}, and scaling independent of the number of terms in the Hamiltonian. The function in question can be inverted via the recently developed quantum singular value transformation algorithm, which transforms the singular values of an operator by a desired polynomial, to recover the original spectrum of the Hamiltonian [23]. Many other quantum algorithms can be recast under this framework, demonstrating the wider utility of such algorithms in efficiently implementing a variety of other computational tasks [26]. The major drawbacks of qubitization are that it requires additional qubits beyond those needed to specify the physical system being simulated and does not apply to time-dependent Hamiltonians.

The quantum stochastic drift protocol [28], or qDRIFT, approximates the unitary time-evolution of a time-independent Hamiltonian by first sampling from a classical probability distribution which is biased towards summands in the input Hamiltonian with large spectral norm. The output of this classical distribution determines the summand of the Hamiltonian whose time-evolution is implemented. Repeating this procedure for each sub-interval of the full simulation time interval yields a product of operator exponentials, and this drifts towards the correct unitary time-evolution with high precision and with a gate complexity independent of the number of terms in the Hamiltonian. qDRIFT was later generalized to the continuous qDRIFT protocol for time-dependent Hamiltonians, which scales in cost as

the time-integral of the absolute value of the Hamiltonian's coefficients [29]. The principal disadvantages of this approach are that it has a larger scaling in the simulation time t compared to other algorithms and does not exploit any commutator structure between the terms of a Hamiltonian.

Given that each of these simulation protocols comes with their own advantages and disadvantages, a goal of this thesis is to show how different simulation methods can be hybridized after moving into the interaction picture (I.P.). This comes in the form of "hybrid" algorithms that combine the various conventional approaches for quantum simulation. These approaches show asymptotic improvements over the individual methods that comprise them and further make interaction picture simulation methods practical in the near term. This is significant because while the interaction picture simulation method provides the best asymptotic scaling known for many problems, the constant factors involved can make it impractical for many applications [30]. This issue is mitigated by combining algorithms such as qDRIFT and qubitization at different stages of the overall simulation procedure within the interaction picture. Since the interaction picture transformation involves conjugation of Hamiltonian summands $\sum_{k\neq j} H_k$ via e^{itH_j}, the unitary invariance of the L^1-norm scaling from qDRIFT essentially eliminates the contribution of H_j to the query complexity of the hybrid protocols.

The next few sections will provide additional details on basics of each of these standard simulation methods, the interaction picture, and how these simulation methods can be combined after an interaction picture transformation.

1.1.1 Continuous qDRIFT

In this subsection, we outline the continuous qDRIFT protocol used to simulate time-dependent Hamiltonians with a scaling depending only on the L^1-norm of the Hamiltonian. At its heart is a classical sampling protocol which randomly samples a simulation time $\tau \in [0, t]$ according to a probability distribution and evolves a given state under the time-independent Hamiltonian $H(\tau)$. The probability distribution is chosen such that it is

biased towards τ with large $||H(\tau)||_\infty$. The result is a simulation protocol that stochastically drifts towards the ideal unitary time evolution with small error in the diamond norm (see Appendix A for an overview of the diamond norm).

We present relevant results from Ref. [29] used throughout this thesis without proof. Let $H(\tau)$ be a time dependent Hamiltonian defined for $0 \leq \tau \leq t$. Unless otherwise specified, we make the following assumptions of $H(\tau)$:

1. It is non-zero and continuously differentiable on $[0, t]$
2. It is finite dimensional, i.e. $H : [0, t] \to \mathbb{C}^{M\times M}$
3. There exists an oracle $W \colon \mathbb{R}^2 \mapsto \mathbb{C}^{M\times M}$ such that for any $\tau \in [0, t]$ and $\Delta \in \mathbb{R}$, $W(\tau, \Delta) = e^{-iH(\tau)\Delta}$

The specific implementation of W depends on the simulation protocol in question. For instance, a concrete realization involves "qubitization oracles" to be discussed later in the thesis. For our present purposes, it suffices to assume the existence of such an oracle and analyze the query complexity of algorithms invoking it as a black box.

The ideal evolution of $H(\tau)$ for time t is given by $E(t, 0) = \exp_{\mathcal{T}}(-i \int_0^t d\tau H(\tau))$ and the quantum channel corresponding to this is

$$\mathcal{E}(t, 0) = E(t, 0)\rho E^\dagger(t, 0) = \exp_{\mathcal{T}}\left(-i \int_0^t d\tau H(\tau)\right)\rho \exp^\dagger_{\mathcal{T}}\left(-i \int_0^t d\tau H(\tau)\right) , \tag{1.1}$$

where the subscript $\mathcal{T}$ in $\exp_{\mathcal{T}}$ denotes the time-ordered exponential. Generalizations of these channels to non-zero initial times can be accomplished simply by changing the limits of integration.

Since it is difficult in practice to implement the ideal channel due to the presence of time-ordered exponentials, we can instead approximate it by a mixed unitary channel defined by

$$\mathcal{U}(t, 0)(\rho) = \int_0^t d\tau \; p(\tau) e^{-i\frac{H(\tau)}{p(\tau)}} \rho e^{i\frac{H(\tau)}{p(\tau)}} , \tag{1.2}$$

where

$$p(\tau) \coloneqq \frac{||H(\tau)||_\infty}{||H||_{\infty,1}}$$

is a probability density function defined for $0 \leq \tau \leq t$ and

$$||H||_{\infty,1} \coloneqq \int_0^t d\tau \|H\|_\infty \ .$$

(i.e. the outermost subscript indicates an L^1 norm while the innermost subscript indicates a Schatten infinity norm). This channel can be implemented via a classical sampling protocol and has the following features:

(a) $p(\tau)$ is biased towards those $\tau \in [0,t]$ with large $||H(\tau)||_\infty$

(b) $p(\tau)$ decreases with the evolution time t since $||H(\tau)||_{\infty,1}$ involves an integral over $[0,t]$

(c) With a time $\tau_i \in [0,t]$ obtained from sampling $p(\tau)$, we can query the oracle W cited above by inputting $W(\tau_i, p(\tau_i)^{-1})$ to obtain an implementation of the unitary time-evolution operator $e^{-iH(\tau_i)/p(\tau_i)}$

This classical sampling protocol and the unitary channel (1.2) implemented by it is denoted by "continuous qDRIFT". We assume the spectral norm $||H||_\infty$ or an upper bound is already known and that we can efficiently sample from $p(\tau)$. We then have the following theorem when the simulation time t is assumed to be sufficiently small:

Theorem 1.1.1 (L^1-norm error bound for continuous qDRIFT, short-time version)**.** *Let $H(\tau)$ be a time-dependent Hamiltonian defined for $0 \leq \tau \leq t$ and satisfying conditions* 1 *and* 2 *above. Define $\mathcal{E}(t,0)$ and $\mathcal{U}(t,0)(\rho)$ as in equations* (1) *and* (2) *respectively. Then*

$$||\mathcal{E}(t,0) - \mathcal{U}(t,0)||_\diamond \leq 4||H||^2_{\infty,1} \ . \tag{1.3}$$

Here the diamond norm, denoted $\|\cdot\|_\diamond$, is defined to be the largest possible trace norm of the object over all possible input states and further maximized over all possible ancillary qubit registers attached to the input state. For this reason it upper bounds standard notions of

distance for quantum channels such as the trace distance and further gives information about the maximum possible distinguishability between two quantum channels. (See Appendix A for information about the diamond norm for quantum channels).

When the simulation time t is large, we will need to divide the simulation interval $[0, t]$ into sub-intervals $[t_j, t_{j+1}]$ where $0 = t_0 < t_1 < \cdots < t_r = t$ and apply the continuous qDRIFT protocol within each to control the simulation error. In these cases, we have a "long-time" version of Theorem 1.1.1:

Theorem 1.1.2. *(L^1-norm error bound for continuous qDRIFT for long simulation time) Let $H(\tau)$ be a time-dependent Hamiltonian defined for $0 \leq \tau \leq t$ and satisfying conditions 1 and 2 above. Define $\mathcal{E}(t,0)$ and $\mathcal{U}(t,0)(\rho)$ as in (1) and (2) respectively. For any positive integer r, there exists a division $0 = t_0 < t_1 < \cdots < t_r = t$*

$$\left\| \mathcal{E}(t,0) - \prod_{j=0}^{r-1} \mathcal{U}(t_{j+1}, t_j) \right\|_\diamond \leq 4 \frac{\|H\|_{\infty,1}^2}{r} \,. \tag{1.4}$$

To ensure the simulation error is at most ϵ, it suffices to choose

$$r \geq 4 \left\lceil \frac{\|H\|_{\infty,1}^2}{\epsilon} \right\rceil .$$

The value of r above can also be interpreted as the query complexity of the continuous qDRIFT protocol, i.e. the number of queries to the oracle W needed to implemented channel (1.2) and satisfy (1.4) with error less than ϵ.

1.1.2 Qubitization and Quantum Singular Value Transformations

Having considered continuous qDRIFT, we now briefly review the basics of the qubitization simulation protocol which we seek to combine with the former. We will also frame qubitization as an example of the general notion of the block-encoding of non-unitary matrices within larger unitary ones.

Qubitization is a method of Hamiltonian simulation involving the synthesis of the time-evolution operator e^{iHt}, where H is a time-independent Hamiltonian, via the implementation

of a walk operator $\mathcal{W}(H)$ whose eigenvalues are an efficiently computable function of those of H. Assuming that we have decomposed H as a linear combination of unitary matrices, the desired walk operator can implemented with the so-called SELECT and PREPARE qubitization oracles. The spectrum can then be transformed efficiently using techniques involving singular value transformations which transform the singular values of an operator by a polynomial function [21, 22].

Block-encoding refers to the embedding of a non-unitary matrix H into a larger unitary U, typically as the upper-left block of U. Once a block-encoding is achieved, a quantum circuit can be expressed in terms of U. This greatly broadens the applicability of quantum computers, particularly in the domain of the simulation of unitary quantum dynamics. We largely follow the treatments in Refs. [27, 31].

Let $H \in \mathrm{End}(\mathbb{C}^N)$, where $N = 2^n$, be a Hermitian operator. Suppose there exists an $(m+n)$-qubit unitary matrix $U_H \in \mathrm{End}(\mathbb{C}^{MN})$, where $M = 2^m$, such that

$$U_H = \begin{pmatrix} H/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix} ,$$

where $\alpha > 0$ is a known normalization constant. We may then get access to H/α by

$$H/\alpha = (\langle 0|^m \otimes I_n) U_H (|0\rangle^m \otimes I_n) .$$

To quantify how "close" the block encoded matrix is to the original one, we introduce the following general definition. This definition can also be extended to the case of block-encodings within superoperators, which we will need to consider for the proofs of some later theorems:

Definition 1.1.3 (Block Encoding)**.** Suppose that A is an n-qubit operator, $\alpha, \varepsilon \in \mathbb{R}_+$, and $m \in \mathbb{N}$. We then say that the $(m+n)$-qubit unitary U_H is a (α, m, ε)-block-encoding of A if

$$\|A - \alpha(\langle S| \otimes I_n) U_H (|S\rangle \otimes I_n)\|_\infty \leq \varepsilon .$$

where $|S\rangle$ is an m-qubit state.

Similarly, we say that a quantum channel Λ is a (α, m, ε)-block-encoding of A if

$$\max_{\rho} \|A\rho A^\dagger - \alpha(\langle T| \otimes I_n)\Lambda(|T\rangle\langle T| \otimes \rho)(|T\rangle \otimes I_n)\|_\infty \leq \epsilon,$$

where the maximization is over density matrices ρ and $|T\rangle$ is an m-qubit state.

Here $|S\rangle$ or $|T\rangle$ are referred to as the "signal state". The previous example involving H is a $(1, m, 0)$-encoding where $|S\rangle = |0\rangle^m$.

Now suppose we are given a time-independent H. H can be decomposed into a linear combination of unitary operators

$$H = \sum_{l=0}^{L-1} w_l H_l, \quad w_l \in \mathbb{R}_0^+, \quad H_l^2 = I \ . \tag{1.5}$$

Here we assume that any complex phases are absorbed into H_l. The two oracles used are a preparation oracle whose action on $|0\rangle^{\log L}$ is defined as follows:

$$\text{PREPARE } |0\rangle^{\log L} = \sum_{l=0}^{L-1} \sqrt{\frac{w_l}{\lambda}} |l\rangle = |\mathcal{L}\rangle \ , \tag{1.6}$$

where

$$\lambda = \sum_l w_l \ ,$$

and a selection oracle whose action on an ancilla register $|l\rangle$ and system register $|\Psi\rangle$ is as follows:

$$\text{SELECT} = \sum_{l=0}^{L-1} |l\rangle\langle l| \otimes H_l \ , \tag{1.7}$$

$$\text{SELECT } |l\rangle|\Psi\rangle \mapsto |l\rangle H_l |\Psi\rangle \ . \tag{1.8}$$

In other words, the SELECT oracle "selects" a unitary H_l conditioned on the state of the ancilla register $|l\rangle$. Using (1.7) and (1.8), it can be shown that SELECT squares to the identity operator and can therefore be considered as a "reflection" operator. Note that we also have the following result for the action of SELECT on $|\mathcal{L}\rangle$:

$$(\langle \mathcal{L}| \otimes I)(\text{SELECT})(|\mathcal{L}\rangle \otimes I) = \frac{1}{\lambda} \sum_l w_l H_l = \frac{H}{\lambda} \ . \tag{1.9}$$

The previous equation is a condition for qubitization and oracles that satisfy this condition are referred to as "qubitization oracles" [22]. If we define

$$U_H = (\text{PREPARE}^\dagger \otimes I)(\text{SELECT})(\text{PREPARE} \otimes I) \ ,$$

it follows from (1.9) that U_H is a $(\|w\|_1, \log L, 0)$-block encoding of H, where $\|w\|_1 = \sum_l |w_l|$.

The desired walk operator, also known as the "iterate", can now be defined as follows:

$$\mathcal{W} = \mathcal{R}_L \cdot \text{SELECT}, \qquad \mathcal{R}_L = (2|\mathcal{L}\rangle\langle\mathcal{L}| \otimes I - I) \ . \tag{1.10}$$

$\mathcal{W}$ is of the form of a Szegedy walk operator since it is the composition of two reflections. From a lemma by C. Jordan on the common invariant subspaces of two reflections [32], it follows that the Hilbert space of the system decomposes under the action of $\mathcal{W}$ into a direct sum of 1 and 2-dimensional irreducible subspaces, where the latter is spanned by $|\mathcal{L}\rangle|k\rangle$ and an orthogonal state $|\phi_k\rangle$. Here, $|k\rangle$ is an eigenstate of H with eigenvalue E_k and $|\phi_k\rangle$ is the component of $\mathcal{W}|\mathcal{L}\rangle|k\rangle$ orthogonal to $|\mathcal{L}\rangle|k\rangle$. Using (1.9), this can be expressed as

$$|\phi_k\rangle = \frac{(I - |\mathcal{L}\rangle\langle\mathcal{L}| \otimes |k\rangle\langle k|) \cdot \text{SELECT}|\mathcal{L}\rangle|k\rangle}{||(I - |\mathcal{L}\rangle\langle\mathcal{L}| \otimes |k\rangle\langle k|) \cdot \text{SELECT}|\mathcal{L}\rangle|k\rangle||} = \frac{(\text{SELECT} - \frac{E_k}{\lambda} I)|\mathcal{L}\rangle|k\rangle}{\sqrt{1 - (\frac{E_k}{\lambda})^2}} \ . \tag{1.11}$$

In the 2-dimensional subspaces, $\mathcal{W}$ acts as a rotation whereas on the 1-dimensional subspaces, it has ± 1 eigenvalues. The matrix elements of $\mathcal{W}$ within a two-dimensional subspace can be computed using the above relations. Using (1.9), the top-left entry is

$$\langle k|\langle\mathcal{L}|\mathcal{W}|\mathcal{L}\rangle|k\rangle = \frac{E_k}{\lambda} \ ,$$

and the upper-right entry using (1.11) is

$$\langle k|\langle\mathcal{L}|\mathcal{W}|\phi_k\rangle = \sqrt{1 - \left(\frac{E_k}{\lambda}\right)^2} \ .$$

The other elements can be computed in an analogous way and we obtain for the form of the 2-dimensional blocks of $\mathcal{W}$

$$\begin{bmatrix} \frac{E_k}{\lambda} & \sqrt{1-\left(\frac{E_k}{\lambda}\right)^2} \\ -\sqrt{1-\left(\frac{E_k}{\lambda}\right)^2} & \frac{E_k}{\lambda} \end{bmatrix} = e^{i \arccos(E_k/\lambda) Y} . \tag{1.12}$$

The controlled walk operator can be implemented using the circuit in Figure 1.1, see Ref. [31]. It is clear from this that $\mathcal{W}$ requires one query to SELECT and at most two queries to PREPARE to implement. The controlled-SELECT operation can be approximated as requiring the same gate complexity to implement as the SELECT operation.

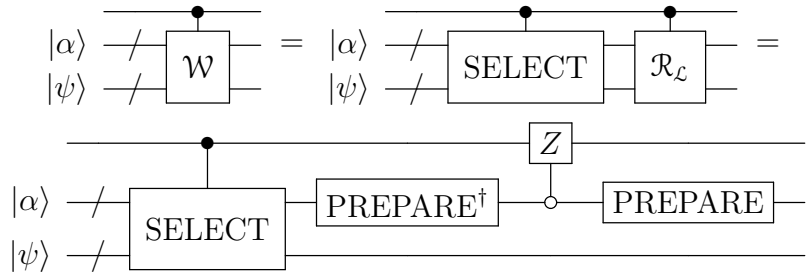


Figure 1.1: Controlled-walk operator in terms of the SELECT and PREPARE oracles

Note that if the condition that $H_l^2 = I$ in (1.5) does not hold, we no longer have the interpretation of SELECT acting like a reflection operator. It then follows that $\mathcal{W}$ cannot be interpreted as a Szegedy walk operator and we can no longer apply Jordan's lemma to it. However, we can still define $\mathcal{W}$ as in (1.10) and the subsequent computations involving the calculation of matrix elements of $\mathcal{W}$ when restricted to the two-dimensional subspace spanned by the orthogonal states $|\mathcal{L}\rangle|k\rangle$ and $|\phi_k\rangle$ remain unaffected. It can still be shown that the Hilbert space decomposes as a direct sum of such 2 dimensional irreducible subspaces as $\mathcal{W}$ does not take vectors within the subspace to those outside of it.

The arccos in (1.12) can be efficiently inverted to recover the original spectrum of H via techniques involving singular value transformations and quantum signal processing. The impetus for the development of the general formalism of singular value transformations was

the Quantum Signal Processing techniques introduced by Low et al. [33]. They considered the following problem: if one applies a gate sequence of the form

$$e^{i\phi_0\sigma_z}e^{i\theta\sigma_x}e^{i\phi_1\sigma_z}e^{i\theta\sigma_x}\cdots e^{i\theta\sigma_x}e^{i\phi_k\sigma_z}\ ,$$

for unknown θ, where $e^{i\theta\sigma_x}$ is the "signal unitary" and where we have control over the angles $\phi_0,\cdots,\phi_k$, what unitary operators can be constructed in this manner? This problem lies at the heart of "Quantum Signal Processing".

The answer to this problem is given in Theorem 3 of Ref. [23] and involves polynomial transformations of the entries of the signal unitary. This idea behind Quantum Signal Processing can be generalized to situations where we apply an arbitrary unitary U between phase operators. It can be shown that this induces polynomial transformations to the singular values of a particular block of the unitary U. In the application to qubitization we are concerned with, Quantum Signal Processing can be applied to the two-dimensional invariant subspaces of the walk operator $\mathcal{W}$.

As we saw in Section 1.1.2, qubitization exploits a lemma by C. Jordan's on the invariant subspaces of two reflection operations and the decomposition of the entire vector space into a direct sum of those subspaces. One of the reflections in the lemma can be replaced by a phase gate in the context of quantum search algorithms [33]. In Ref. [23], the other reflection is replaced by an arbitrary unitary U and the invariant subspaces in question are those arising from the singular value decomposition of a block of the unitary matrix. For our purposes, we only need the following results.

Definition 1.1.4 (Theorem 17 of Ref. [23])**.** Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and $U,\Pi,\tilde{\Pi}\in\mathrm{End}(\mathcal{H}_U)$ be linear operators on $\mathcal{H}_U$ such that U is unitary and $\Pi,\tilde{\Pi}$ are orthogonal projectors. Let $\Phi\in\mathbb{R}^n$. Then we define the phased alternating sequence U_Φ as follows

$$U_\Phi := \begin{cases} e^{i\phi_1(2\Pi-I)}U\prod_{j=1}^{(n-1)/2}(e^{i\phi_{2j}(2\Pi-I)}U^\dagger e^{i\phi_{2j+1}(2\tilde{\Pi}-I)}U) & \text{if n is odd} \\ \prod_{j=1}^{n/2}(e^{i\phi_{2j-1}(2\Pi-I)}U^\dagger e^{i\phi_{2j}(2\tilde{\Pi}-I)}U) & \text{if n is even} \end{cases}.$$

Figure 1.2 shows the circuit implementation of the alternating phase modulation sequence for even n.

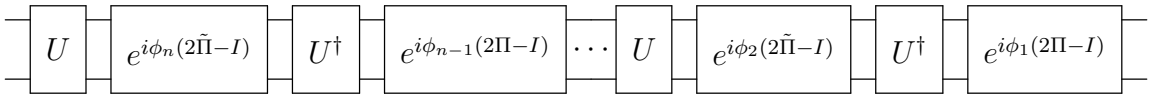


Figure 1.2: Circuit for U_Φ when n is even

Theorem 1.1.5 (Quantum Singular Value Transformation: Theorem 17 of Ref. [23])**.** *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and let $U, \Pi, \tilde{\Pi} \in \mathrm{End}(\mathcal{H}_U)$ be linear operators on $\mathcal{H}_U$ such that U is unitary, and $\Pi, \tilde{\Pi}$ are orthogonal projectors. Let $P \in \mathbb{C}[x]$ and $\Phi \in \mathbb{R}^n$. Then*

$$P^{(SV)}(\tilde{\Pi}U\Pi) = \begin{cases} \tilde{\Pi}U_\Phi\Pi & \text{if } n \text{ is odd} \\ \Pi U_\Phi\Pi & \text{if } n \text{ is even} \end{cases},$$

where $P^{(SV)}$ is a polynomial of degree at most n that performs a singular value transformation on the operator to which it is applied.

The polynomials in the above theorem are required to satisfy the conditions listed in Corollary 8 of Ref. [23]:

(a) P has parity n mod 2

(b) $\forall x \in [-1, 1]\colon |P(x)| \leq 1$

(c) $\forall x \in (-\infty, -1] \cup [1, \infty)\colon |P(x)| \geq 1$

(d) If n is even, then $\forall x \in \mathbb{R}\colon P(ix)P^*(ix) \geq 1$

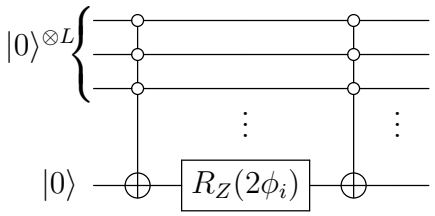


Figure 1.3: Circuit for fractional reflection gadget, $e^{i2\phi_j(2\Pi-I)}$, used in quantum singular value transformations.

Qubitization works by inverting the arccosine. While in principle this boils down to the problem of applying a cosine transformation to the input, in practice a Fourier-Chebyshev expansion is used via the Jacobi-Anger expansion that requires both the odd and even terms to closely approximate the desired function and guarantee that it is within $[-1, 1]$ for the entire domain to use the bounds provided in the work. This process is described in detail in Ref. [33] as well as in Section 5 of Ref. [23].

Applying the preceding theorems to $U = \text{CTRL}(\mathcal{W})$ and $\Pi = \tilde{\Pi} = |0\rangle^L\langle 0|^L \otimes I$, where L is the number of qubits in the register $|\alpha\rangle$ in Figure 1.1, will enable us to invert the arccos in the spectrum of the walk-operator. A circuit for the unitary operator $e^{i2\phi_j(2\Pi-I)}$ is given in Figure 1.3.

Note that we are merely concerned with the existence of a transformation of the spectrum of the walk-operator by a polynomial via the preceding theorem rather than the finding of the phase factors needed to effect a given polynomial transformation. Constructive algorithms for finding these phase factors are outlined in Refs. [26, 27, 34, 35].

The overall query complexity for qubitization and the singular value transformation is given by the following result as expressed in the language of block encodings.

Theorem 1.1.6 (Corollary 60 of Ref. [23])**.** *Let $\epsilon \in (0, \frac{1}{2})$, $t \in \mathbb{R}$ and $\alpha \in \mathbb{R}^+$. Let U be an $(\alpha, a, 0)$-block encoding of the unknown Hamiltonian H. In order to implement an*

ϵ-precise Hamiltonian simulation unitary V which is an $(1, a+2, \epsilon)$-block encoding of e^{itH}, it is necessary and sufficient to use U a total number of times

$$\Theta\left(\alpha|t| + \frac{\log(1/\varepsilon)}{\log(e + \log(1/\varepsilon)/(\alpha|t|))}\right). \tag{1.13}$$

Letting $\alpha = \lambda$ in our notation and assuming that ε is small, we can simplify (1.13) as

$$\Theta\left(\lambda t + \frac{\log(1/\varepsilon)}{\log\log(1/\varepsilon)}\right). \tag{1.14}$$

The linear term comes from the qubitization portion of the procedure while the logarithmic term stems from the transformation of the singular values via the procedure outlined above. This result can be equivalently interpreted as the query complexity for qubitization in terms of the number of queries (modulo irrelevant constants) needed to the PREPARE and SELECT oracles, since $U = \mathrm{CTRL}(\mathcal{W})$ is related to PREPARE and SELECT via Figure 1.1.

1.1.3 Trotterization

Trotterization is the oldest method of quantum simulation based on product formulas. We present the relevant results on Trotterization errors used in this thesis. Our ultimate goal is to synthesize this method of simulation with the interaction picture and continuous qDRIFT, and compare it with a hybrid continuous qDRIFT and qubitization protocol. This will be followed by an application of both methods to several physical models.

Let $H = \sum_{i=1}^{\Gamma} H_i$ be a time-*independent* Hamiltonian expressed as a sum of Γ terms. The unitary time-evolution operator generated by H is then $e^{it\sum_{i=1}^{\Gamma} H_i}$. There are a variety of product formulas that can be used to decompose the time-evolution operator into a product of exponentials involving the individual terms H_i. The most basic is the first-order Lie-Trotter formula

$$\mathscr{S}_1(t) \coloneqq e^{itH_\Gamma} \cdots e^{itH_1}.$$

Higher order generalizations are the Suzuki formulas defined recursively as

$$\mathscr{S}_2(t) \coloneqq e^{i\frac{t}{2}H_1} \cdots e^{i\frac{t}{2}H_{\Gamma-1}} e^{itH_\Gamma} e^{i\frac{t}{2}H_{\Gamma-1}} \cdots e^{i\frac{t}{2}H_1},$$

$$\mathscr{S}_{2k}(t) \coloneqq \mathscr{S}_{2k-2}(u_k t)^2 \mathscr{S}_{2k-2}((1-4u_k)t)\mathscr{S}_{2k-2}(u_k t)^2 \ ,$$

where $u_k = (4 - 4^{-(2k-1)})^{-1}$. There is an extensive literature devoted to investigating the utility and performance of various product-formulas for a variety of physical systems and applications [8, 9, 19]. While there are multiple strategies for addressing the time-ordering of the operators for the time-ordered operator exponentials that emerge when simulating time-dependent Hamiltonians, we broadly follow the analysis outlined in Ref. [18].

Let $H(t) = \sum_S H_S(t)$ be a time-*dependent* Hamiltonian acting on N particles, where $S \subset \{1, \ldots, N\}$, and each term has bounded norm and acts on at most k particles with k a constant independent of N. The time-evolution operator $E(t, 0)$ governing the evolution of the system from time 0 to t is determined by the Schrodinger equation

$$\frac{d}{dt}E(t,0) = -iH(t)E(t,0) \ ,$$

which admits a solution in terms of a time-ordered exponential

$$E(t,0) = \exp_{\mathcal{T}}\left\{ -i\int_0^t H(s)ds \right\} .$$

It turns out that the Trotter-Suzuki formulas given above can be generalized to time-dependent scenarios, even in situations where the Hamiltonian experiences fluctuations on time-scales shorter than the time step Δt [18]. Suppose we wish to simulate the time-evolution of our system up to time $t_r + \Delta t = T$ from $t_0 = 0$. The exact time-evolution operator can be broken up into shorter segments of the form

$$E(T,0) = \prod_{i=0}^{r} E(t_i + \Delta t, t_i) \ ,$$

where

$$E(t_i + \Delta t, t_i) = \exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} ds \sum_S H_S(s) \right) . \tag{1.15}$$

In the case where the sum over S involves only two terms, H_1 and H_2, the generalized Trotter-Suzuki expansion is of the form

$$\begin{aligned} E^{\mathrm{TS}}(t_j + \Delta t, t_j) &= \exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} ds H_1(s) \right) \exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} ds H_2(s) \right) \\ &= E_1^{\mathrm{TS}}(t_j + \Delta t, t_j) E_2^{\mathrm{TS}}(t_j + \Delta t, t_j) \ , \end{aligned} \tag{1.16}$$

and gives rise to a simulation error of

$$\|E(t_j + \Delta t, t_j) - E^{\mathrm{TS}}(t_j + \Delta t, t_j)\|_\infty \leq c_{12}(\Delta t^2) \ , \tag{1.17}$$

where c_{12} is given by

$$c_{12} = \frac{1}{(\Delta t)^2} \int_{t_j}^{t_j+\Delta t} dv \int_{t_j}^{v} du \|[H_1(u), H_2(v)]\|_\infty \leq \frac{1}{2} \max_{u,v} \left(\|[H_1(u), H_2(v)]\|_\infty \right) \ . \tag{1.18}$$

1.1.4 Interaction Picture

The interaction picture or Dirac picture of quantum mechanics is one of the three representations of operators and states in quantum mechanics [36, 37]. It is intermediate to the Schrodinger and Heisenberg pictures of quantum mechanics, where the former is characterized by state vectors that evolve in time but with operators constant in time, and vice versa for the latter. Within the interaction picture however, both operators and states have time dependence but the latter evolves according to the interaction Hamiltonian consisting of the left-over terms in the original Hamiltonian. This picture is particularly useful with dealing with terms in a Hamiltonian that can be treated as small perturbations to a main term such as in time-dependent perturbation theory, where it is used in deriving transition rates via Fermi's golden rule and the Dyson series perturbative expansion of the time-evolution operator [38]. It also finds widespread application in interacting quantum field theories, e.g. Refs. [39, 40, 41].

We follow the derivation in Ref. [38]. Consider a time-independent Hamiltonian $H = \sum_i H_i$. Suppose the energy eigenvalues and eigenstates of H_j for some j are known.

At $t = t_0$, let the state of the physical system be given by $|\alpha\rangle$. At a later time t, we denote the state in the Schrodinger picture by $|\alpha, t_0; t\rangle_S$. Now define

$$|\alpha, t_0; t\rangle_I \coloneqq e^{iH_j t} |\alpha, t_0; t\rangle_S \ , \tag{1.19}$$

where we have implicitly set $\hbar = 1$ and where the subscript I indicates the same situation as represented in the so-called "interaction picture" (I.P.).

We also define observables in the interaction picture as

$$A_I(t) \coloneqq e^{iH_j t} A_S e^{-iH_j t} \ . \tag{1.20}$$

The physical implication of this definition is that we pick any term in the Hamiltonian and move into its "interaction frame" via conjugation by $e^{iH_j t}$. The major difference between this definition and the analogous one in the Heisenberg picture is the appearance of H_j in the former as opposed to the full H in the latter.

We now take the time derivative of equation (1.19):

$$\begin{aligned} i\frac{\partial}{\partial t}|\alpha, t_0; t\rangle_I &= i\frac{\partial}{\partial t}\left(e^{iH_j t}|\alpha, t_0; t\rangle_S\right) \\ &= -H_j e^{iH_j t}|\alpha, t_0; t\rangle_S + e^{iH_j t}(H_j + \sum_{i\neq j} H_i)|\alpha, t_0; t\rangle_S \\ &= e^{iH_j t}\sum_{i\neq j} H_i e^{-iH_j t} e^{iH_j t}|\alpha, t_0; t\rangle_S = H_I(t)|\alpha, t_0; t\rangle_I \ , \end{aligned} \tag{1.21}$$

where we used the Schrodinger equation in the second equality. Thus we have

$$i\hbar\frac{\partial}{\partial t}|\alpha, t_0; t\rangle_I = H_I(t)|\alpha, t_0; t\rangle_I \ , \tag{1.22}$$

with

$$H_I(t) = e^{iH_j t}\left(\sum_{i\neq j} H_i\right)e^{-iH_j t} = \sum_{i\neq j} H_i^I, \tag{1.23}$$

where

$$H_i^I = e^{iH_j t} H_i e^{-iH_j t} \ .$$

This is a Schrodinger-like equation for the time-evolution of the interaction picture state but with the Hamiltonian H replaced by H_I.

It is important to note the distinction between how observables in the interaction picture are represented in (1.20) versus the interaction Hamiltonian above. Naively, we would expect the full Hamiltonian to be what is conjugated within the big parentheses in (1.23) by analogy with (1.20). This "discrepancy" merely arises from the fact that we needed to define H_I as above to obtain the Schrodinger-like equation (1.22).

We can apply the interaction picture to the continuous qDRIFT protocol outlined before and obtain the following simple lemma.

Lemma 1.1.7. *(L^1-norm error bound for IP continuous qDRIFT for long simulation time) Let $H_I(\tau)$ be an interaction picture Hamiltonian as in (1.23). Suppose it is defined for $0 \leq \tau \leq t$ and satisfies conditions 1 and 2 in Section 1.1.1. Define $\mathcal{E}(t,0)$ and $\mathcal{U}(t,0)$ as in (1.1) and (1.2) respectively but with $H_I(\tau)$. Then for any positive integer r, there exists a division $0 = t_0 < t_1 < \cdots < t_r = t$ such that*

$$\left\| \mathcal{E}(t,0) - \prod_{j=0}^{r-1} \mathcal{U}(t_{j+1}, t_j) \right\|_\diamond \leq 4t^2 \frac{\| \sum_{i \neq j} H_i \|_\infty^2}{r} \,. \tag{1.24}$$

To ensure the simulation error is at most ϵ, it therefore suffices to choose

$$r \geq 4 \left\lceil \frac{t^2 \| \sum_{i \neq j} H_i \|_\infty^2}{\epsilon} \right\rceil .$$

Proof. We can substitute (1.23) directly into equation (1.4) and the expression for r. Note however the spectral norm of an operator (and the Schatten norms more generally) is invariant under unitary transformations of that operator. We then obtain the simplification

$$||H_I||_{\infty,1} = \int_0^t d\tau \|H_I(\tau)\|_\infty = \left\| \sum_{i \neq j} H_i \right\|_\infty t \,,$$

so that

$$\left\| \mathcal{E}_I(t,0) - \mathcal{U}_I(t,0) \right\|_\diamond \leq 4 \frac{(\| \sum_{i \neq j} H_i \|_\infty)^2 t^2}{r} \,, \tag{1.25}$$

and

$$r \geq 4 \left\lceil \frac{(\| \sum_{i \neq j} H_i \|_\infty)^2 t^2}{\epsilon} \right\rceil ,$$

to ensure our simulation error is less than some desired ϵ. □

As before, r can also be interpreted as the number of queries to the oracle W defined in Section 1.1.1. Each resulting time-independent piece will need to be simulated using techniques like Trotterization or Qubitization and the main goal of the thesis is to quantify

the overall query and gate complexity of "hybrid" protocols combining these with the IP continuous qDRIFT technique outlined here.

Comparing this result to Theorem 1.1.2, we see that moving into the interaction frame of a fixed term H_j of the overall Hamiltonian effectively "eliminates" its contribution to the error. Moreover, due to the properties of the spectral norm and the interaction Hamiltonian, L^1-norm dependence of the results in Theorem 1.1.2 reduce to those reminiscent of the time-independent case. This behavior recurs in subsequent results and is particularly useful when dealing with terms with unbounded behavior or large ∞-norm, such as the electric term in the Schwinger Model considered later in this thesis.

1.2 Error Correction of Abelian Lattice Gauge Theories

As gauge theories lie at the heart of the framework governing the interactions and forces described by the Standard Model, considerable effort has been expended in numerical simulations of these theories to extract physical quantities. Lattice gauge theories (LGTs) have been among the most fruitful formulations of non-perturbative approaches amenable for implementation on classical computers [42, 43, 44, 45, 46]. In regimes where classical simulations are plagued by an exponential scaling of the computational cost, digital quantum computers have emerged as a promising platform for the efficient simulation of LGTs. Notable examples are real-time dynamics or the study of systems at finite density [47, 48, 49, 50].

Despite considerable advances on this front, digital quantum simulation of LGTs can time-evolve a given initial state into unphysical sectors where gauge symmetries are violated due to noise and the approximation error associated with the simulation protocol used. A popular approach to mitigate this problem, especially useful for ground-state calculations but shown to be useful also out-of-equilibrium [51], is to enforce Gauss' Law by adding an energy penalty term to the Hamiltonian (see e.g. Refs. [48, 52, 47, 53, 51] and also Ref. [1] for techniques to reduce the gate cost of adding penalty terms). Another proposal for error mitigation in gauge theories uses random gauge transformations to suppress the component of the quantum state in the unphysical Hilbert space [54, 55]. The recent approach proposed

in Ref. [56], and its generalization to non-Abelian theories [57], uses instead a quantum oracle to detect the presence of gauge violating errors by performing explicit Gauss' Law checks and flagging an ancilla qubit. These techniques are suitable for error detection but in general do not possess error correction capabilities and are not fault-tolerant.

With the long-time goal of performing quantum simulation of LGT on fault-tolerant protocols, an intriguing possibility is to tailor general purpose error correction schemes to best exploit the structural properties of these theories in order to reduce the resource requirements for early explorations (see e.g. Ref. [58] for a recent attempt in this direction using the surface code). The physical intuition behind the approach followed here is that error correcting codes can be seen as artificial gauge theories where the logical Hilbert space is determined by states that satisfy a suitable local symmetry. When simulating LGTs which themselves need to satisfy a physical local symmetry, it might then be advantageous to exploit this natural redundancy to reduce the cost of the full error correction encoding.

We now review the basics of abelian LGTs and repetition codes that will be used for the constructions presented in Chapter 3.

1.2.1 Structure of Abelian Lattice Gauge Theories

We follow the basic outline given in Ref. [56] and review the structure of abelian lattice gauge theories, specifically for the general gauge groups $G = \mathbb{Z}_n$ and $G = \mathrm{U}(1)$ which contain those considered throughout this thesis. There are several physical models for which the gauge symmetries discussed here are important such as the Schwinger Model, i.e. QED in 1+1 dimensions on a lattice [59, 60]. It is the one of the simplest concrete examples of an Abelian LGT and serves as a convenient setting for the analysis and application of Gauss' Law symmetries to error correction. This model has been extensively used as an important stepping stone in simulations of lattice field theories using both tensor networks and quantum devices [61, 62, 52, 63, 64].

We discretize space on a cubic lattice L with sites labeled by s and links labeled by l. We assume that the lattice consists of N sites for even $N \geq 0$ and that a staggered fermion

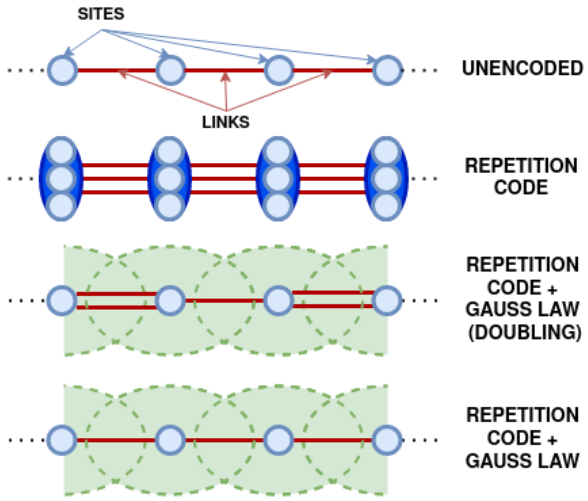


Figure 1.4: Schematic illustration of the differences between two error correcting schemes for a simple one-dimensional LGT: a traditional bit-flip encoding scheme and the two schemes proposed here exploiting the Gauss' Law gauge symmetry.

representation is used wherein every second site is positronic. Each link l is associated an independent separable Hilbert space $\mathcal{H}_l$ with the same orthonormal basis:

$$\langle \epsilon' | \epsilon \rangle = \delta_{m',m}, \qquad \hat{I} = \sum_{\epsilon} |\epsilon\rangle\langle\epsilon| \tag{1.26}$$

with

$$\epsilon', \epsilon \in \begin{cases} \mathbb{Z}_n, & \text{if } G = \mathbb{Z}_n \\ \mathbb{Z}, & \text{if } G = \mathrm{U}(1). \end{cases}$$

The Hamiltonian for this LGT is a function of the link operators $\hat{U}_l$ and their conjugate electric fields $\hat{E}_l$ defined explicitly in this basis by

$$\hat{U}_l = \sum_{\epsilon_l} |\epsilon_l + 1\rangle\langle\epsilon_l|, \qquad \hat{E}_l = \sum_{\epsilon_l} \epsilon_l |\epsilon_l\rangle\langle\epsilon_l|. \tag{1.27}$$

From this expression we can see that $\hat{U}_l$ acts as a raising operator and its adjoint as a lowering operator on the Hilbert space $\mathcal{H}_l$ of the link. Operators defined on different link

Hilbert spaces commute while the same-link commutation relations are given by

$$[\hat{E}_l, \hat{U}_l] = \hat{U}_l, \qquad G = \mathrm{U}(1) \tag{1.28}$$

$$\hat{Q}_l \hat{U}_l \hat{Q}_l^\dagger = \hat{U}_l e^{2\pi i/n} \qquad G = \mathbb{Z}_n \tag{1.29}$$

where

$$\hat{Q}_l \coloneqq e^{2\pi i \hat{E}_l/n} = \sum_{\epsilon_l=0}^{N-1} e^{2\pi i \epsilon_l/n} |\epsilon_l\rangle\langle\epsilon_l|.$$

The form of the commutation relation for $\mathbb{Z}_n$ is due to the fact that the electric field values are periodic, so the Hamiltonian depends on $\hat{Q}_l$.

When considering fermionic matter fields on the sites, we work in the occupation number basis where number operators n_σ are diagonal with eigenvalues $\{0, 1\}$. Here, σ is a collective index denoting the relevant species or indices (like flavor or spinor) involved. The global state of the entire lattice is spanned by a basis given by a specification of electric fields on the links of the lattice and occupation numbers on the sites. Due to the locality of the symmetry, we will typically consider a particular site on a lattice and those links attached to it and denote the corresponding basis states by

$$|\mathbf{E}, \rho\rangle \to \otimes_{i=1}^{D} |E_i(s)\rangle \otimes_{i=1}^{D} |E_i(s - \hat{e}_i)\rangle \otimes_\sigma |n_\sigma\rangle,$$

where D is the spatial dimension of the lattice and ρ is a discretized charge density defined by

$$\hat{\rho}(s) = \sum_\sigma e_\sigma \hat{n}_\sigma(s),$$

and $e_\sigma = \pm 1$. Due to gauge invariance, states in the physical Hilbert space satisfy a local Gauss' Law which relates the state of a site with the state of the links emanating from it. Gauge-invariant states are in the kernel of the operator

$$\begin{aligned} \hat{G}_s &\coloneqq (\nabla \cdot \hat{E})(s) - \hat{\rho}(s) \\ &\coloneqq \sum_{i=1}^{D} (\hat{E}_i(s) - \hat{E}_i(s - \hat{e}_i)) - \sum_\sigma e_\sigma \hat{n}_\sigma(s) \ , \end{aligned} \tag{1.30}$$

where the second line is obtained by discretization of the gradient operator on the lattice.

When dealing with a U(1) gauge group, it is necessary to truncate the link electric field values to enable digital quantum simulation with a finite number of qubits. This can be accomplished by "wrapping" the electric field at a cutoff Λ:

$$\hat{E}_l = \sum_{\epsilon_l=-\Lambda}^{\Lambda-1} \epsilon_l |\epsilon_l\rangle\langle\epsilon_l| \tag{1.31}$$

$$\hat{U}_l|\Lambda-1\rangle = |-\Lambda\rangle \tag{1.32}$$

$$\hat{U}_l^\dagger|-\Lambda\rangle = |\Lambda-1\rangle. \tag{1.33}$$

This choice of discretization results in a modification of the commutation relations as follows:

$$[\hat{E}_l, \hat{U}_l] = \hat{U}_l - 2\Lambda|-\Lambda\rangle\langle\Lambda-1| \tag{1.34}$$

$$[\hat{E}_l, \hat{U}_l^\dagger] = -\hat{U}_l^\dagger + 2\Lambda|\Lambda-1\rangle\langle-\Lambda|. \tag{1.35}$$

Note that with our choice of the lower and upper bound, the link Hilbert spaces are even-dimensional and can therefore be mapped onto a $\lceil\log(2\Lambda)\rceil$-qubit Hilbert space. In this thesis, we will restrict the discussion to $\Lambda = 1$ and comment on the prospects of generalizing our constructions to arbitrary cutoffs in Section 3.4.

1.2.2 Repetition Codes

We now review the basics of the bit and phase flip error correction codes following the treatment given in Ref. [65]. First consider a noisy classical communications channel through which we wish to send a bit between two locations and suppose its behavior is such that it flips the bit with probability p. To protect the bit against the effects of noise, we can employ what is known as a "repetition code". This involves replacing the bit with three copies of itself, i.e. $0 \to 000$ and $1 \to 111$. These new bit strings are denoted as the "logical 0" and "logical 1" and we send these through the channel. The receiver then attempts to decode what the original bit was. If the output is 010 for instance, then provided the probability p of error is not high and the noise acts independently on each bit, it is likely the second

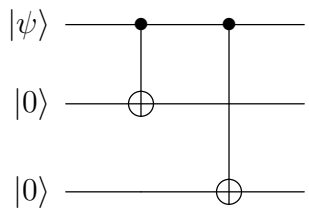


Figure 1.5: Circuit for creating the encoded logical state $|\psi\rangle_L = a|000\rangle + b|111\rangle$ for the bit-flip code.

bit was flipped and that the original bit was 0. This is known as majority voting, since the intended original message is determined by whatever bit value appears more in the output. This can obviously fail if more than one bit was flipped. It can be easily determined that with this encoding scheme, the transmission becomes more reliable if $p < 1/2$.

Now consider a noisy quantum channel that applies a bit-flip, or X gate, to a state $|\psi\rangle$ sent through it with probability p. We write $|\psi\rangle$ in terms of the computational basis as $|\psi\rangle = a|0\rangle + b|1\rangle$ and encode it in three qubits as $a|000\rangle + b|111\rangle$. In other words, we have a mapping between the "physical" qubits $|0\rangle$ and $|1\rangle$ to the logical qubits $|0\rangle_L = |000\rangle$ and $|1\rangle_L = |111\rangle$ respectively, where the subscript L denotes a logical state. Such an encoding can be accomplished by the circuit in Figure 1.5.

Each qubit in the encoded state is passed through a separate bit-flip channel. If a bit-flip occurs on at most one qubit, we can measure the parities of the qubits by performing projective measurements of the operators Z_1Z_2 and Z_2Z_3, where the tensor product is implied. This process is known as making "syndrome measurements" for the error syndromes Z_1Z_2 and Z_2Z_3. These operators have eigenvalues of ± 1. Z_1Z_2 measures the parities of the first two qubits and yields the eigenvalue -1 if they differ and $+1$ if they do not. Z_2Z_3 acts in the same way for the second and third qubits. The measurement outcomes of either operator allows us to determine which qubit was flipped. For instance, if eigenvalues of -1 are obtained from the measurement of both operators, we know that with high probability the second qubit was flipped. We can then perform error correction by applying an X gate on

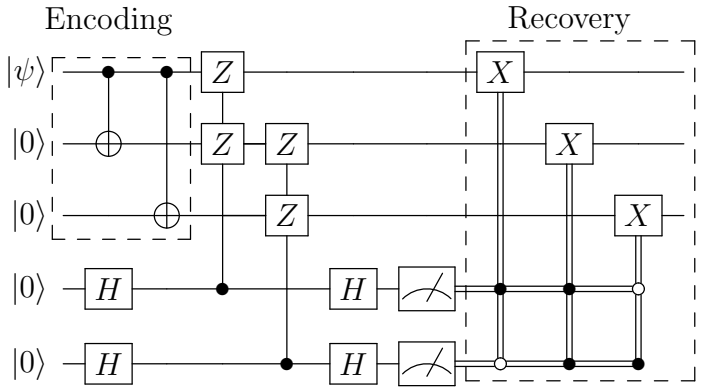


Figure 1.6: Full circuit for correcting a bit flip error on a general state $|\psi\rangle$.

Z_1Z_2	Z_2Z_3	Correction
1	1	III
1	-1	IIX
-1	1	XII
-1	-1	IXI

Table 1.1: Syndrome measurements outcomes, and correction operations for bit flip error correction code.

the 2^{nd} qubit to flip it back to its original state. Note that the measurement of these operators gives no information about the amplitudes a and b of the encoded state and therefore do not destroy the state we wish to perform error detection and correction on. Table 1.1 outlines the possible measurement outcomes for the syndromes Z_1Z_2 and Z_2Z_3 and the error correction operations to perform. Figure 1.6 gives the full circuit to correct bit-flip errors. An equivalent circuit used for the projective measurement of the stabilizers Z_1Z_2 and Z_2Z_3 is presented in Figure 1.7. We will use this decomposition in the rest of this work.

Now suppose we have a noisy quantum channel that applies a phase flip (i.e a Z gate) with probability p to a qubit in the state $|\psi\rangle = a|0\rangle + b|1\rangle$. Unlike the bit-flip encoding,

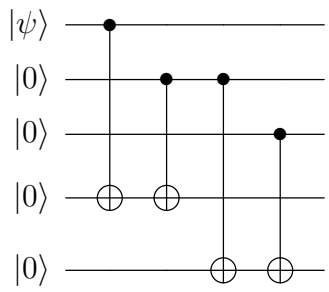


Figure 1.7: Equivalent circuit for the measurement of the stabilizers Z_1Z_2 and Z_2Z_3 for the bit-flip error correction code. This can be obtained from the identity $HZH = X$ and the fact that controlled-Z gates are equivalent to a controlled-Z gate with the control and target flipped.

there is no classical analogue of applying a "phase" to a bit. However, we can convert this channel to a bit flip channel by working in the $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ basis. With respect to this basis, the Z operator takes $|+\rangle$ to $|-\rangle$ and therefore acts as a bit flip. We can then apply the same logic for error correction in the bit-flip case to the present case by switching from the computational basis to the $|+\rangle$ and $|-\rangle$ basis via the Hadamard gate. In our present work, we will find it convenient to use the logical codeword basis $|0\rangle_L = (|+++\rangle + |---\rangle)/\sqrt{2}$ and $|1\rangle_L = (|+++\rangle - |---\rangle)/\sqrt{2}$. Then to detect errors, we can perform projective measurements of the stabilizers X_1X_2 and X_2X_3 to determine the parity of the bits. Based on the measurement outcomes, we can apply Z gates to correct the errors accordingly. The encoding circuit and the phase-flip error correction procedure are depicted in Figure 1.8 and Figure 1.9 respectively.

The constructions given in Ref. [66] allow us to ensure the fault tolerance of the encoding, error detection, and recovery operations in either code. The underlying technique involves introducing an extra "flag" qubit prepared in the $|+\rangle$ state and performing CNOT operations from it on the syndrome ancilla qubit at key points in the circuit (see Figure 3(b) in Ref. [66]). This flag qubit is then measured in the X basis and a result of $|-\rangle$ indicates an error of weight two or more on the data qubits. Additional flag qubits can be added between each gate in

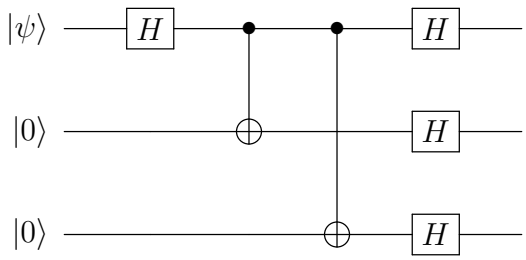


Figure 1.8: Circuit for creating the encoded logical state $|\psi\rangle_L = a|0\rangle_L + b|1\rangle_L$ for the phase-flip code, where $|0\rangle_L = (|++\!+\rangle + |--\!-\rangle)/\sqrt{2}$ and $|1\rangle_L = (|++\!+\rangle - |--\!-\rangle)/\sqrt{2}$.

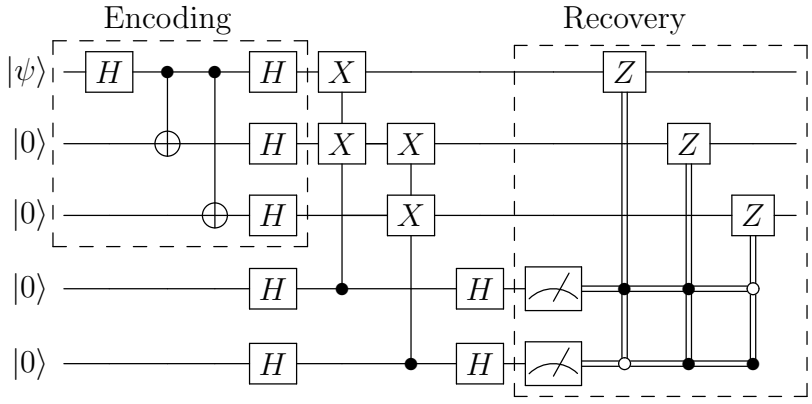


Figure 1.9: Full circuit for correcting phase flip errors on a general state $|\psi\rangle$.

the stabilizer measurement to ensure a localization of errors. Similar constructions apply in creating fault-tolerant versions of other important subroutines like logical state-preparation and stabilizer measurements (see Appendix A in Ref. [66]).

1.3 Topological Data Analysis

The subject of topology in mathematics is concerned with determining the properties of topological spaces that are invariant under continuous deformations, or more accurately under homeomorphisms or homotopy equivalences. Invariant properties help us classify topological spaces as follows: if two spaces are topologically equivalent, then they have the

same invariants. Equivalently, if two spaces do not have the same invariants, then they are not topologically equivalent to each other.

One such invariant property is the number of holes of different dimensions in the topological space, which are called the Betti numbers. For example, it is intuitive that the sphere and the torus cannot be continuously deformed into each other since the sphere has no 1-dimensional hole whereas the torus has two, a hole in the center of the "doughnut" and one in the tube. The tool of singular homology, and its special case of simplicial homology we will mainly deal with, rigorously captures these notions and its fundamental objects are simplicial complexes. Simplicial complexes are topological spaces obtained from attaching objects called simplices together, which are points, lines, triangles, tetrahedra, and their higher-dimensional generalizations. Simplicial complexes can serve as approximations to topological spaces that nevertheless reflect their topological properties, and their discrete nature makes the determination of properties like the Betti numbers more tractable.

Simplicial homology has antecedents as far back as the 1800s, but its application to the problem of determining on computers the topological properties of data sets is more recent [67, 68, 69]. An important goal in data analysis is to extract features of a data set and use them to cluster or classify the data. This data set can be represented as a set of points in some metric space, such as $\mathbb{R}^n$ with the Euclidean distance function. One approach for the analysis is to convert the point cloud into a graph where the vertices are the given data points and the edges are determined by whether or not pairs of points lie within a chosen distance ϵ. This approach can capture features such as connectivity but ignores potential higher dimensional features, especially if the data points are sampled from some underlying high-dimensional manifold [70]. Topological data analysis (TDA) attempts to extract such higher dimensional global topological features, namely the Betti numbers or estimates thereof, of simplicial complexes constructed out of an underlying data set via simplicial homology.

Classical deterministic algorithms for estimating Betti numbers of simplicial complexes have a run time typically exponential in the dimension of the simplicial complex and are thus intractable for even a moderate amount of data [71, 72]. Lloyd *et al.* developed a quantum

algorithm for extracting all the Betti numbers of a simplicial complex with a runtime they claimed was exponentially faster than the best classical algorithms [13]. This algorithm was not immediately "dequantizable" with the same techniques used to dequantize a wide classes of quantum algorithms with claimed exponential speedups over their classical counterparts at the time [73, 74], thereby generating considerable interest in TDA as a potential example of a genuine quantum advantage [15, 75].

One main problem of interest in this thesis is to therefore develop a classical randomized algorithm that finds the Betti numbers of a simplicial complex associated to an input graph with runtime polynomial in the number of points. We briefly develop in the next few sections the necessary background material needed to understand simplicial homology for topological data analysis, loosely following the treatment in Ref. [76].

1.3.1 Simplicial Homology

Let $v_0, \ldots, v_k$ be $k+1$ distinct points in $\mathbb{R}^n$. The set $\{v_0, \ldots, v_k\}$ is said to be **affinely independent** if the set $\{v_1 - v_0, \ldots, v_k - v_0\}$ is linearly independent. In other words, we consider the given set of points to be affinely independent if when we take one of the points to be the "origin" (say v_0 WLOG) and draw vectors from this point to the others, the collection of the resulting vectors is linearly independent.

If $\{v_0, \ldots, v_k\}$ is affinely independent, the **k-simplex** spanned by them is the set

$$[v_0, \ldots, v_k] := \left\{ \sum_{i=0}^{k} t_i v_i \colon t_i \geq 0 \text{ and } \sum_{i=0}^{k} t_i = 1 \right\} \tag{1.36}$$

Equivalently, a simplex is just the convex hull of its affinely independent set of vertices. The points v_i are the **vertices** of the simplex and the integer k is the **dimension** of the simplex. Figure 1.10 shows some examples of simplices. Note that it follows from the definitions given that $k \leq n$ since any set of $n+2$ points or more cannot be affinely independent. This in turn is because no collection of $n+1$ vectors or more in an n-dimensional vector space can be linearly independent. Such a collection of vectors therefore cannot determine any simplicies of dimension $n+1$ or higher.

Let σ be a k-simplex. A simplex spanned by a non-empty subset of the vertices of σ is a **face** of σ. For example, the 0-dimensional faces of σ are its vertices and its 1-dimensional faces are the edges, which are spanned by two vertices. Faces of σ that are not equal to σ are called **proper faces**. The $(k-1)$-dimensional faces of σ are called its **boundary faces** and their union is its **boundary**.

Figure 1.10: A 0-simplex (point), 1-simplex (edge), 2-simplex (triangle), and 3-simplex (tetrahedron) from left to right.

Definition 1.3.1 (**Simplicial Complex**)**.** A simplicial complex S is a finite collection of simplices satisfying the following conditions:

1. If $\sigma \in S$, every face of σ is in S

2. If $\sigma_1, \sigma_2 \in S$, then $\sigma_1 \cap \sigma_2 = \varnothing$ or $\sigma_1 \cap \sigma_2$ is a face of σ_1 and σ_2

The first condition says that a simplicial complex should also contain all the faces of a given simplex in the complex. The second condition says that any two simplices in a simplicial complex either do not intersect or intersect at a common face of both. We define the **dimension of a simplicial complex** to be the maximum of the dimensions of all simplices in the complex.

It can be shown that a simplicial complex is completely determined by its vertices and information about which sets of vertices span which simplices. This provides the motivation for the following definition.

Definition 1.3.2 (**Abstract Simplicial Complex**)**.** An abstract simplicial complex is a collection C of non-empty finite sets such that if $s \in C$, then every non-empty subset of s is also in C.

Figure 1.11: (Left) A 2D simplicial complex in $\mathbb{R}^2$. (Right) A set that is not a simplicial complex in $\mathbb{R}^2$. It violates condition 2 of Definition 1.3.1.

This general notion of a simplicial complex is particularly useful when we wish to construct one without reference to a particular embedding into Euclidean space.

We will mainly be concerned with computing the Betti numbers of certain simplicial complexes called clique complexes associated to graphs. This problem is known as "clique homology". The graphs serving as inputs into our classical algorithm are not necessarily induced by any finite-dimensional data and are instead abstract graphs, i.e. abstract simpicial complexes.

Definition 1.3.3 (**Graphs**)**.** A **graph** G is a pair of objects $G = (V, E)$, where V is a set of elements referred to as the "vertices" of G and E is a set consisting of pairs of vertices thought of as "edges" connecting the pairs of vertices.

Equivalently, we may also define a graph as a 0 or 1 dimensional simplicial complex. Given an undirected graph, i.e. one in which the edges are not assumed to have direction, a **clique** C of a graph is a subset of V such that every pair of distinct vertices in C is connected by an edge. C is called a **k-clique** if $|C| = k$.

We now define the notion of a clique complex alluded to earlier:

Definition 1.3.4 (**Clique Complex**)**.** The clique complex of a graph G is the abstract simplical complex formed by associating a k-simplex to every $(k+1)$-clique in G.

The topological data we are most interested in are the Betti numbers of a simplicial

complex, which give the number of holes of a given dimension in that simplicial complex. We now show how to determine the Betti numbers of a simplicial complex.

Let K be a simplicial complex and consider the set of k-simplices in K. In what follows, we could like to make sense of taking "linear combinations" of the k-simplices in K with coefficients in some field R (we will only need $R = \mathbb{R}$ or $R = \mathbb{C}$ in this thesis). Since the set of k-simplices is merely a set and carries no algebraic structure, this does not make sense literally. However, given any set S, we can construct a corresponding R-vector space with the elements of S serving as a basis in the following way.

First consider the set of functions $f\colon S \to R$ which take non-zero values on only *finitely* many elements of S. This is an R-vector space under pointwise addition and scalar multiplication by elements of R and is called the "free R-vector space" on S, denoted by $F^R(S)$. Note that for each element $s \in S$, there is a unique function $f_s \in F^R(S)$ that takes the value 1 on s and 0 on all other elements of S. Thus we can identify each $s \in S$ with the corresponding function $f_s \in F^R(S)$ and therefore think of S as a subset of $F^R(S)$. Every element of $f \in F^R(S)$ can be written as a linear combination $f = \sum_{s\in S} r_s f_s$ where $r_s \in R$. Thus S is a basis for $F^R(S)$ and $F^R(S)$ is finite-dimensional iff S is finite, which is always the case for our data analysis applications. In this case, we may simply identify the elements of S viewed as a subset of $F^R(S)$ with the standard basis vectors $e_i = (0, \ldots, 1, \ldots, 0)^T$, where the 1 is in the i-th spot.[2]

The free R-vector space on the set of k-simplices in K is denoted by $C_k(K, R)$, $C_k(K)$, or C_k depending on if the simplicial complex or field are understood, and is called the **k-th chain group**. Its elements are called **k-chains** and can be written as linear combinations $\sum_i r_i\sigma_i$, where σ_i are the k-simplices of K (viewed as basis elements in $C_k(K, R)$) and $r_i \in R$. It follows from the considerations in the preceding paragraph that as a vector space,

[2]The preceding constructions can be generalized to create free R-modules on a set S, where R can be any commutative ring. This is what is often needed in algebraic topology to make sense of taking linear combinations of simplices with coefficients in such rings. Linear combinations of simplices with coefficients in arbitrary abelian groups are also necessary in algebraic topology, but we will not need such general constructions for our applications.

$\dim C_k(K, R)$ is simply the number of k-simplices in K.

Definition 1.3.5 (**Boundary Map**)**.** Let $\sigma = [u_0, \ldots, u_k]$ be a k-simplex. The boundary map on k-simplices is a map

$$\partial_k \colon C_k(K) \to C_{k-1}(K) \tag{1.37}$$

that acts as

$$\partial_k \sigma = \sum_{i=0}^{k} (-1)^i [u_0, u_1, \ldots, \hat{u}_i, \ldots, u_k] \tag{1.38}$$

where $\hat{u}_i$ denotes that the vertex i has been removed.

The boundary map acts on k-simplices $\sigma \in C_k(K)$ and gives a $(k-1)$-simplex $\partial_k \sigma$ that can be interpreted as the boundary of σ. The boundary of a 0-simplex, i.e. a point, is simply 0.

An **k-cycle** is an k-chain $c \in C_k(K)$ such that $\partial_k c = 0$. Therefore, k-cycles are precisely the kernel of the boundary map and are a subspace of $C_k(K)$ denoted by $Z_k = \ker \partial_k$. An k-chain c is an **k-boundary** if there exists an $(k+1)$-chain $\sigma \in C_{k+1}(K)$ such that $c = \partial_{k+1}(\sigma)$. Equivalently, k-boundaries are precisely the image of the boundary map and form a subspace denoted by $B_k(K) = \text{Im } \partial_{k+1}$. Figure 1.12 shows an example of a 1-boundary and 1-cycle.

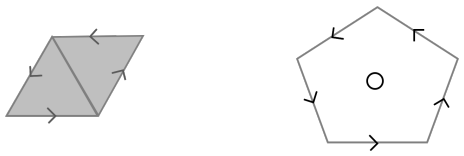


Figure 1.12: (Left) The 1-boundary of a 2-chain as indicated by the arrows. (Right) A 1-cycle that is not the boundary of any 2-chain.

It turns out that there is a relationship between the two subspaces as implied by the following fundamental result in homology theory:

Proposition 1.3.6. *$\partial_k \circ \partial_{k+1}(\sigma) = 0$ for all $k+1$ chains σ in K and all $0 \leq k < \dim K$.*

Proof.

$$\begin{aligned}\partial_k(\partial_{k+1}(\sigma)) &= \sum_{i=0}^{k}(-1)^i\partial_{k+1}([u_0,\ldots,\hat{u}_i,\ldots,u_{k+1}] \\ &= \sum_{i=0}^{k}(-1)^i\left(\sum_{j<i}(-1)^j[u_0,\ldots,\hat{u}_j,\ldots,\hat{u}_i,\ldots,u_{k+1}] + \sum_{j>i}(-1)^{j-1}[u_0,\ldots,\hat{u}_i,\ldots,\hat{u}_j,\ldots,u_{k+1}]\right)\end{aligned}$$

Note the factor of $(-1)^{j-1}$ in the second sum. This occurs because if $j = i+1$ for instance, this really corresponds to the i-th iteration in the sum since the i-th vertex was removed. Hence we must shift the original factor of $(-1)^j$ down. Then

$$= \sum_{0\le j<i\le k+1}(-1)^{i+j}[u_0,\ldots,\hat{u}_j,\ldots,\hat{u}_i,\ldots,u_{k+1}] - \sum_{0\le i<j\le k+1}(-1)^{i+j}[u_0,\ldots,\hat{u}_i,\ldots,\hat{u}_j,\ldots,u_{k+1}]$$

Exchanging the roles of i and j in the second sum gives the result. □

The above proposition essentially says that the boundary of a boundary is 0. This immediately implies that the image of ∂_{k+1}, B_k, is contained in the kernel of ∂_k, Z_k. This allows us to define the homology groups as follows:

Definition 1.3.7 (**Homology Groups**)**.** The **k-th homology group** H_k of a simplicial complex is the quotient vector space

$$H_k(K,R) = Z_k(K,R)/B_k(K,R) = \text{Ker } \partial_k/\text{Im } \partial_{k+1}\,. \tag{1.39}$$

Definition 1.3.8 (**Betti Numbers**)**.** The **k-th Betti number** β_k is the dimension of the k-th homology group.

The k-th homology groups range from $k = 0$ to $k = \dim K$, the dimension of the simplicial complex. They are generated by k-cycles that are *not* the k-boundaries of any simplex. In other words, these are simplices that enclose a "void" or "hole" (see the right image of Figure 1.12 above). It is worth noting that β_0, the 0^{th} Betti number, represents the number of connected components the simplicial complex has.

The problem of computing the Betti numbers of a simplicial complex therefore reduces to the problem of computing the rank of the boundary map. A common and simple classical approach to doing this at the computational level is as follows.

Let K be a simplical complex and assume for simplicity that we are working over the field $\mathbb{Z}_2$. Label the p-simplices in $C_p(K)$ by $x_1, \ldots, x_{n_p}$ and the $(p-1)$-simplices in $C_{p-1}(K)$ by $y_1, \ldots, y_{n_{p-1}}$. These simplices form bases for $C_p(K)$ and $C_{p-1}(K)$ as mentioned previously. We can then represent the action of the boundary map ∂_p on $C_p(K)$ as follows

$$\partial_p(x_j) = \sum_{i=1}^{n_{p-1}} a_j^i y_i \text{ where } a_j^i = \begin{cases} 1 & \text{if } y_i \text{ is a face of } x_j \\ 0 & \text{otherwise} \end{cases} \tag{1.40}$$

Then for any p-chain $c = \sum_{j=1}^{n_p} a_j x_j$, we can write the above in matrix form

$$\partial_p c = \begin{bmatrix} a_1^1 & a_1^2 & \ldots & a_1^{n_p} \\ a_2^1 & a_2^2 & \ldots & a_2^{n_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \ldots & a_{n_{p-1}}^{n_p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_p} \end{bmatrix} \tag{1.41}$$

Thus the boundary map on $C_p(K)$ can be represented as an $n_{p-1} \times n_p$ sparse matrix with entries in $\mathbb{Z}_2$. The columns of this matrix span Im $\partial_p = B_{p-1}$, so rank $\partial_p = \dim B_{p-1} = b_{p-1}$. The boundary matrix can brought into the Smith Normal Form via a generalization of Gaussian elimination which applies to any principal ideal domain (this includes fields). This results in a matrix with a number of 1's on the diagonal equal to the rank of the boundary matrix. Gaussian elimination for ∂_p takes $O(n_{p-1} n_p \min(n_p, n_{p-1}))$ time and requires $O((n_{p-1} + n_p)^2)$ memory [69, 77]. When working over arbitrary fields, the same procedure holds with the exception that the matrix elements of the boundary operator can be ± 1 or 0.

For our purposes though, we will find it more convenient to work with a different but simpler to analyze object to determine the Betti numbers of K.

Definition 1.3.9 (**Combinatorial Laplacian**)**.** The **k-th combinatorial Laplacian**[3] is

[3]This operator is sometimes defined in terms of the coboundary operator acting on cochains, in which

the operator $\Delta_k \colon C_k(K) \to C_k(K)$ defined by

$$\Delta_k = \partial_k^\dagger \partial_k + \partial_{k+1}\partial_{k+1}^\dagger.$$

$\partial_k^\dagger \colon C_{k-1}(K) \to C_k(K)$ is the adjoint of ∂_k when viewed as a matrix with respect to the standard basis of $C_k(K)$ and $C_{k-1}(K)$. It is immediate from the definition that Δ_k is Hermitian and even positive semi-definite when $C_k(K)$ has the standard Hermitian inner product $\langle v, w\rangle \coloneqq v^\dagger w$, with $v, w \in C_k(K)$. The most important for our immediate purposes however is the following theorem.

Theorem 1.3.10. $\dim \ker(\Delta_k) = \beta_k$.

Proof. We endow $C_k(K)$ with the standard Hermitian inner product $\langle v, w\rangle \coloneqq v^\dagger w$ where $v, w \in C_k(K)$. We first claim that $\ker \Delta_k = \ker \partial_k^\dagger \partial_k \cap \ker \partial_{k+1}\partial_{k+1}^\dagger$. Clearly if $v \in C_k(K)$ is in $\ker \partial_k^\dagger \partial_k \cap \ker \partial_{k+1}\partial_{k+1}^\dagger$, then $\Delta_k v = (\partial_k^\dagger \partial_k)v + (\partial_{k+1}\partial_{k+1}^\dagger)v = 0$, so $v \in \ker \Delta_k$. Conversely, let $v \in \ker \Delta_k$. Then $(\partial_k^\dagger \partial_k)v + (\partial_{k+1}\partial_{k+1}^\dagger)v = 0$. Multiplying on the left by $v^\dagger$, we have

$$v^\dagger(\partial_k^\dagger \partial_k)v + v^\dagger(\partial_{k+1}\partial_{k+1}^\dagger)v = (\partial_k v)^\dagger(\partial_k v) + (\partial_{k+1}^\dagger v)^\dagger(\partial_{k+1}^\dagger v) = \langle \partial_k v, \partial_k v\rangle + \langle \partial_{k+1}^\dagger v, \partial_{k+1}^\dagger v\rangle = 0$$

Since the inner products are each greater than or equal to 0, they must vanish independently for the last equality to hold. By the definition of an inner product, we therefore have $\partial_k v = 0$ and $\partial_{k+1}^\dagger v = 0$, which implies $\partial_k^\dagger \partial_k v = 0$ and $\partial_{k+1}\partial_{k+1}^\dagger v = 0$ and thus $v \in \ker \partial_k^\dagger \partial_k \cap \ker \partial_{k+1}\partial_{k+1}^\dagger$.

Next, note that for any operator $A \colon V \to W$ between finite-dimensional vector spaces V, W endowed with Hermitian inner products, $\ker(AA^\dagger) = \ker(A^\dagger)$. To show this, let $x \in \ker(A^\dagger)$. Then $A^\dagger x = 0 \implies AA^\dagger x = 0$, so $x \in \ker AA^\dagger$. If $x \in \ker(AA^\dagger)$, then

$$AA^\dagger x = 0 \implies \langle AA^\dagger x, x\rangle = \langle A^\dagger x, A^\dagger x\rangle = 0 \implies A^\dagger x = 0,$$

---

case the dimension of its kernel is actually the dimension of the singular cohomology group $H^k(K, R)$. However, all our homology groups are finite dimensional vector spaces and are therefore finitely generated. The universal coefficient theorem from algebraic topology then implies $H_k(K, R) \cong H^k(K, R)$, so it is immaterial which version of the combinatorial Laplacian we study.

so $x \in \ker A^\dagger$. This result shows $\ker(\partial_k^\dagger \partial_k) = \ker \partial_k$ and $\ker(\partial_{k+1}\partial_{k+1}^\dagger) = \ker \partial_{k+1}^\dagger$, so $\ker(\Delta_k) = \ker \partial_k \cap \ker \partial_{k+1}^\dagger$.

Lastly, using the same notation as above, we show $(\mathrm{Im}(A))^\perp = \ker A^\dagger$. Let $y \in (\mathrm{Im}(A))^\perp$. Then for all $x \in V$, $\langle Ax, y\rangle = 0$. Let $x = A^\dagger y$. Then

$$\langle AA^\dagger y, y\rangle = \langle A^\dagger y, A^\dagger y\rangle = 0 \implies A^\dagger y = 0,$$

so $y \in \ker A^\dagger$. Conversely if $y \in \ker A^\dagger$, then for any $x \in V$

$$\langle Ax, y\rangle = \langle x, A^\dagger y\rangle = 0 \implies y \in (\mathrm{Im}A)^\perp.$$

This result shows $\ker \partial_{k+1}^\dagger = (\mathrm{Im}(\partial_{k+1}))^\perp$.

Putting all these results together:

$$\begin{aligned}
\ker \Delta_k &= \ker \partial_k^\dagger \partial_k \cap \ker \partial_{k+1}\partial_{k+1}^\dagger \\
&= \ker \partial_k \cap \ker \partial_{k+1}^\dagger \\
&= \ker \partial_k \cap (\mathrm{Im}(\partial_{k+1}))^\perp \\
&\cong H_k(K)
\end{aligned}$$

Thus $\dim(\ker \Delta_k) = \dim H_k(K) = \beta_k$. □

From an algorithmic standpoint, we can therefore determine the Betti numbers of a simplicial complex, or approximations of them, by studying eigenstates corresponding to sufficiently small eigenvalues. The computational complexity of determining the low-lying eigenstates of general Hermitian operators is well-studied in the complexity theoretic literature. In the next section, we give a brief summary of the complexity theoretic results that have been established for this problem and for topological data analysis in general.

1.3.2 Computational Complexity of Topological Data Analysis

There are several variations on the problem of determining the Betti numbers of simplicial complexes on classical or quantum computers. The determination of *exact* Betti numbers of

simplicial complexes is known to be NP-hard for classical computers [78] and even PSPACE-hard for complex (projective) algebraic varieties [79], which are zero sets of (homogeneous) polynomials with coefficients in $\mathbb{C}$ (see Appendix C for a brief overview of some classes from quantum and classical complexity theory). In line with these classical results is that of Ref. [80], which shows that deciding if a combinatorial Laplacian has a trivial or non-trivial kernel on quantum computers is QMA_1-hard. Thus this problem is likely not efficient even for quantum computers in the worst case scenarios.

Given these results, one may instead attempt to broaden the problem in various ways. Rather than determining the exact number of zero eigenvalues of the combinatorial Laplacian, one can instead attempt to count to number of eigenvalues below a certain sufficiently small threshold for instance. This can still be meaningful since results like Cheeger's inequality show that the magnitudes of the small non-zero eigenvalues of the graph Laplacian (a special case of the combinatorial Laplacian for graphs) characterises the connectedness of the graph [81], with analogous results holding for combinatorial Laplacians on their corresponding simplicial complexes [82]. Such a quantity may be more computationally tractable to determine as well. For example, it has been conjectured that the magnitude of the smallest non-zero eigenvalue of combinatorial Laplacians typically scales inverse polynomially in even difficult instances of the problem [83]. If this is true, the number of "small" eigenvalues would then coincide with the number of zero eigenvalues under a suitable choice of threshold. Unfortunately, if the combinatorial Laplacians have a sufficiently large spectral gap, determining the number of zero eigenvalues could still be QMA_1-hard if the threshold chosen happens to be below the spectral gap.

Another variation on the problem is to determine *normalized* Betti numbers with inverse polynomial *additive* error, where the normalized Betti number is the ratio of the number of zero eigenvalues of Δ_k to the total number of eigenvalues of Δ_k (i.e. the number of k simplices in the simplicial complex). This is more natural from a quantum computational perspective since a quantum computer estimates probabilities or normalized quantities. It has been shown that the problem of estimating the low-lying spectral density for arbitrary

Hermitian operators, i.e. the ratio of the number of small eigenvalues within a certain range to the total number of eigenvalues, is DQC1-hard and thus likely to be immune to "dequantization", i.e. be efficiently solvable on classical computers (see Appendix C) [15]. Subsequent work [84], using tools supersymmetric many-body physics and homological algebra, showed the problem remains DQC1-hard when restricting the input to combinatorial Laplacians of general chain complexes, though whether DQC1-hardness persists when considering combinatorial Laplacians on clique complexes, the case primarily considered here, remains an open problem. More specifically, it may be possible to come up with an efficient classical algorithm for this problem and this is what we address in this thesis.

Chapter 2

HYBRIDIZED METHODS OF QUANTUM SIMULATION IN THE INTERACTION PICTURE

We summarize the scaling of the newly introduced hybrid schemes and compare them to standard approaches in Table 2.1. These are expressed in terms of the oracle complexity for approximating the time-evolution under a Hamiltonian $H = \sum_{i=1}^{L} H_i$. For the Trotter/qDRIFT based I.P. methods, we show the asymptotic scaling in terms of queries to oracles $\{W_k\}_{k=1}^{L}$ implementing $W_k(t) = e^{-iH_k t}$ for any choice of summand H_k. For the hybrid qubitization I.P. based methods, the queries are instead to the SELECT/PREPARE oracles (see Section 1.1.2 for details) and the oracle $W_l(t) = e^{-itH_l}$. The latter is specifically used to implement the time evolution of the term H_l to enter the interaction picture, while the oracles $\{W_k\}_{k=1}^{L}$ above are used to implement all the time-evolutions. The constants λ and λ_α are obtained by first writing H as a linear combination of unitaries $H = \sum_k \omega_k U_k$ with real $\omega_l > 0$. Then we have $\lambda = \sum_k \omega_k$ and $\lambda_\alpha = \sum_{k\neq l} \omega_k = \lambda - \omega_l$. As anticipated above, the hybrid I.P. schemes introduced here can become advantageous when $\lambda_\alpha \ll \lambda$ or $\|H - H_l\|_\infty \ll \|H\|_\infty$, that is, when the Hamiltonian term H_l has a large norm (here and in the rest of the thesis, $\|H\|_p$ denotes the Schatten p-norm of a matrix. See Appendix B for further details).

2.1 Hybrid Trotterization and qDRIFT Protocol

We now present an analysis of our first hybrid simulation protocol where a generalization of the time-dependent Trotter-Suzuki formula given in (1.16) proved below is combined with continuous qDRIFT. Let $H(t) = \sum_{k=1}^{L} H_k(t)$. The procedure is as follows:

1. Use the Trotterization technique below to approximate the time-ordered exponential

of $H(t)$ as a product of L time-ordered exponentials.

2. Use continuous qDRIFT to approximate each time-ordered exponential by the channel (1.2). Implementing this channel involves sampling from a probability distribution and yields a product of r time-*independent* terms of the form $\exp\left(-iH_I(\tau_k)/p(\tau_k)\right)$, where r is the number of sub-intervals of the whole simulation interval.

Before proving the error bounds for these processes, we first show the following simple lemma with time arguments suppressed for notational convenience:

Lemma 2.1.1. *Let $\mathcal{E}^{TS}$ denote the superoperator representing the Trotter-Suzuki decomposition of the time-ordered exponential in (1.16) and let $\mathcal{E}$ be as in (1.1). If D_{2^n} is the set of density operators in the domain of $\mathcal{E}$, then*

$$\|\mathcal{E} - \mathcal{E}^{TS}\|_\infty := \sup_{\rho \in D_{2^n}} \|\mathcal{E}(\rho) - \mathcal{E}^{TS}(\rho)\|_\infty \leq 2\|E - E^{TS}\|_\infty \,. \tag{2.1}$$

Proof. From the triangle inequality we have that

$$\begin{aligned}
\|\mathcal{E} - \mathcal{E}^{\mathrm{TS}}\|_\infty &\leq \sup_{\rho \in D_{2^n}} \|E\rho E^\dagger - E^{\mathrm{TS}}\rho E^\dagger\|_\infty + \sup_{\rho \in D_{2^n}} \|E^{\mathrm{TS}}\rho E^\dagger - E^{\mathrm{TS}}\rho (E^{\mathrm{TS}})^\dagger\|_\infty \\
&= \sup_{\rho \in D_{2^n}} \|(E\rho - E^{\mathrm{TS}}\rho)E^\dagger\|_\infty + \sup_{\rho \in D_{2^n}} \|E^{\mathrm{TS}}(\rho E^\dagger - \rho (E^{\mathrm{TS}})^\dagger)\|_\infty \\
&= \sup_{\rho \in D_{2^n}} \|(E - E^{\mathrm{TS}})\rho\|_\infty + \sup_{\rho \in D_{2^n}} \|(E - E^{\mathrm{TS}})\rho\|_\infty \\
&\leq 2\|E - E^{\mathrm{TS}}\|_\infty \,.
\end{aligned}$$

In the third line, we used the unitary invariance of the infinity norm and that $\|A\|_\infty = \|A^\dagger\|_\infty$ for any bounded square operator A. The latter follows from the fact that the Schatten infinity norm is the spectral norm, which is the largest eigenvalue of $\sqrt{AA^\dagger}$ and coincides with the largest eigenvalue of $A^\dagger A$. In the fourth line, we used the sub-multiplicativity of the infinity norm and the fact that $\|\rho\|_\infty \leq 1$ for all density operators. □

We now have the following results for quantum simulation with this hybrid protocol:

Theorem 2.1.2 (Hybrid Trotterization and qDRIFT Simulation)**.** *Let* $\{H_k(t) : k = 1, \ldots, L\}$ *be a set of time-dependent Hermitian operators satisfying conditions 1 and 2 in Section 1.1.1. Let* $\mathcal{U}_k$ *denote the superoperator representing the continuous qDRIFT channel for the time-dependent summand* $H_k(t)$ *as in* (1.2)*. Then given a decomposition of* $[0, t]$ *into* r *sub-intervals of length* $\Delta t = t/r$*,*

$$\left\| \mathcal{E}(t,0) - \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}_k(t_j + \Delta t, t_j) \right\|_\infty \leq \frac{L^2 c_{\max}}{r} t^2 + 4r \sum_{k=1}^{L} \|H_k\|_{\infty,1}^2 \,. \tag{2.2}$$

Here $c_{\max}$ *is defined as* $c_{\max} = \frac{1}{L^2} \max_{u,v} \sum_p^L \|[H_p(u), \sum_{q>p}^L H_q(v)]\|_\infty$ *and the* 1*-norm in* $\|H_k\|_{\infty,1}^2$ *denotes an integral over an interval of size* Δt*.*

Proof. We first generalize (1.16) to the case where $H(t)$ is the sum of L time-dependent terms. Suppose we break up $H(t)$ as $H(t) = H_1(t) + \sum_{k>1}^{L} H_k(t)$. Treating the sum as our "second" term and considering a specific time-step $[t_l, t_l + \Delta t]$, we can substitute these into the expression for c_{12} above. Our proof of the error bound from recursively applying the bound in (1.16) is inductive. Let us consider the base case. Using (1.18) we have that

$$\begin{aligned} &\left\| \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} H_1(t) + \sum_{k=2}^{L} H_k(t) \mathrm{d}t \right) \right. \\ &\qquad \left. - \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} H_1(t) \mathrm{d}t \right) \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} \sum_{k=2}^{L} H_k(t) \mathrm{d}t \right) \right\|_\infty \\ &\qquad \leq \frac{1}{2} \max_{u,v} \left\| \left[H_1(u), \sum_{q>1}^{L} H_q(v) \right] \right\|_\infty \Delta t^2 \end{aligned} \tag{2.3}$$

Next assume that for some $p \geq 1$ we have that

$$\begin{aligned} &\left\| \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} H_1(t) + \sum_{k=2}^{L} H_k(t) \mathrm{d}t \right) \right. \\ &\qquad \left. - \prod_{q=1}^{p} \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} H_q(t) \mathrm{d}t \right) \exp_{\mathcal{T}} \left(-i \int_{t_j}^{t_j+\Delta t} \sum_{k=p+1}^{L} H_k(t) \mathrm{d}t \right) \right\|_\infty \\ &\qquad \leq \frac{1}{2} \sum_{\ell=1}^{p} \max_{u,v} \left\| \left[H_\ell(u), \sum_{q>\ell}^{L} H_q(v) \right] \right\|_\infty \Delta t^2 \,. \end{aligned} \tag{2.4}$$

We then have from the triangle inequality and the unitary invariance of Schatten norms that for $p+1$

$$\begin{aligned}
&\left\| \exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} H_1(t)+\sum_{k=2}^{L} H_k(t)\mathrm{d}t\right)\right. \\
&\qquad \left. -\prod_{q=1}^{p+1}\exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} H_q(t)\mathrm{d}t\right)\exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t}\sum_{k=p+2}^{L} H_k(t)\mathrm{d}t\right)\right\|_{\infty} \\
&\leq \frac{1}{2}\sum_{\ell=1}^{p}\max_{u,v}\left\|\left[H_\ell(u),\sum_{q>\ell}^{L} H_q(v)\right]\right\|\Delta t^2+\left\|\exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t}\sum_{k=p+1}^{L} H_k(t)\mathrm{d}t\right)\right. \\
&\qquad \left. -\exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t} H_{p+1}(t)\mathrm{d}t\right)\exp_{\mathcal{T}}\left(-i\int_{t_j}^{t_j+\Delta t}\sum_{k=p+2}^{L} H_k(t)\mathrm{d}t\right)\right\|_{\infty} \\
&\leq \frac{1}{2}\sum_{\ell=1}^{p}\max_{u,v}\left\|\left[H_\ell(u),\sum_{q>\ell}^{L} H_q(v)\right]\right\|_{\infty}\Delta t^2+\frac{1}{2}\max_{u,v}\left\|\left[H_{p+1}(u),\sum_{q>p+1}^{L} H_q(v)\right]\right\|_{\infty}\Delta t^2 \\
&= \frac{1}{2}\sum_{\ell=1}^{p+1}\max_{u,v}\left\|\left[H_\ell(u),\sum_{q>\ell}^{L} H_q(v)\right]\right\|_{\infty}\Delta t^2\ . \qquad (2.5)
\end{aligned}$$

This demonstrates the induction step and combined with the base case in (2.3) shows the error bound we need inductively.

Since this analysis was for the time interval $[t_j+\Delta t, t_j]$ and since there are r such intervals sub-dividing our simulation interval, we can multiply our previous result by r using Box 4.1 in [65]. Since $\Delta t = t/r$, we then have

$$\|E(t,0)-E^{\mathrm{TS}}(t,0)\|_{\infty}\leq\frac{L^2 c_{\max}}{2r}t^2\ . \qquad (2.6)$$

Now note that from (1.3) that if we denote the time evolution under H_k to be given by the unitary superoperator $\mathcal{E}_k(t_j+\Delta t,t_j)$, then

$$\|\mathcal{E}_k(t_j+\Delta t,t_j)-\mathcal{U}_k(t_j+\Delta t,t_j)\|_{\infty}\leq\|\mathcal{E}_k(t_j+\Delta t,t_j)-\mathcal{U}_k(t_j+\Delta t,t_j)\|_{\diamond}\leq 4\|H_k\|_{\infty,1}^2\ . \qquad (2.7)$$

Using the sub-multiplicativity and triangle inequality for the induced infinity norm for superoperators, we get

$$\left\|\prod_{k=1}^{L}\mathcal{E}_k(t_j+\Delta t,t_j)-\prod_{k=1}^{L}\mathcal{U}_k(t_j+\Delta t,t_j)\right\|_{\infty}\leq 4\sum_{k=1}^{L}\|H_k\|_{\infty,1}^2\ , \qquad (2.8)$$

where the 1-norm in the subscript on the RHS denotes an integral over an interval of size Δt from t_j to $t_j + \Delta t$. Note that this notation causes the duration of the integral over time to be implicitly rather than explicitly defined. Despite this drawback, we use this notation in places throughout the thesis for brevity.

A straightforward generalization of the argument in Box 4.1 in [65] using the sub-multiplicativity and triangle inequality for the diamond norm, and the fact quantum channels have diamond norm at most 1 yields

$$\left\| \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{E}_k(t_j + \Delta t, t_j) - \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}_k(t_j + \Delta t, t_j) \right\|_{\infty} \leq 4r \sum_{k=1}^{L} \|H_k\|_{\infty,1}^2 \,. \tag{2.9}$$

From the above inequality, Lemma 1.1.7, and Lemma 2.1.1 we obtain that the bound of the induced ∞-norm of the difference between the super-operator and the hybridized channel is

$$\begin{aligned} &\left\| \mathcal{E}(t,0) - \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}_k(t_j + \Delta t, t_j) \right\|_{\infty} \leq \\ &\left\| \mathcal{E}(t,0) - \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{E}_k(t_j + \Delta t, t_j) \right\|_{\infty} + \left\| \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{E}_k(t_j + \Delta t, t_j) - \prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}_k(t_j + \Delta t, t_j) \right\|_{\infty} \\ &\leq \frac{L^2 c_{\max}}{r} t^2 + 4r \sum_{k=1}^{L} \|H_k\|_{\infty,1}^2 \,. \end{aligned} \tag{2.10}$$

□

Note that since the 1-norm in $\|H_k\|_{\infty,1}^2$ denotes an integral over a time-interval of size Δt, this term scales with t^2/r. If we implement each qDRIFT channel $\mathcal{U}_k$ with some error ϵ, an easy application of the triangle inequality will add an additional subdominant term of $rL\epsilon$ to (2.2).

It should also be noted that the result of Theorem 2.1.2 applies for both the case of time-dependent as well as time-independent Hamiltonian evolution. This is relevant because it shows that the lowest-order Trotter-Suzuki formula can be combined with qDRIFT profitably wherein small terms in the Hamiltonian can be reallocated between the Trotter and the qDRIFT portions of the Hamiltonian to reduce the simulation cost. This can be seen as an extension of the coalescing strategy of Ref. [85].

If we compare this result with that given in Theorem 7' of Ref.[29], we find that the error in the latter approach using solely continuous qDRIFT scales with $\|H_k\|^2_{\infty,1,1}$, where the last 1 in the subscript denotes a sum over k, and we square *after* performing the integral and sum. While the result in (2.2) adds a term which scales at worst quadratically in the number of terms L in the Hamiltonian, we will find that for systems like those considered later in this thesis, we can exploit the commutation relations between the terms in the Hamiltonian to give bounds that scale linearly with L.

Corollary 2.1.3 (Hybrid Trotterization and qDRIFT Simulation in Interaction Picture)**.** *Let $H = \sum_{k=1}^{L} H_k$ be a time-independent Hamiltonian where each summand satisfies conditions 1 and 2 in Section 1.1.1. Then given a decomposition of $[0,t]$ into r sub-intervals of size Δt, we can perform the Hamiltonian simulation of H in the interaction frame of H_l as in (1.23) such that*

$$\left\| \mathcal{E}(t,0) - \prod_{j=1}^{r} \prod_{k\neq l}^{L} \mathcal{U}_k(t_j + \Delta t, t_j) \right\|_\infty \leq \frac{t^2}{r}\left(c_I + 4 \sum_{k\neq l}^{L} \|H_k\|^2_\infty \right) , \tag{2.11}$$

where $c_I = \sum_{p\neq l}^{L} \|[H_p, \sum_{q>p}^{L} H_q]\|_\infty$. To ensure the simulation error in the infinity-norm is less than ϵ, it therefore suffices to choose

$$r \geq \frac{t^2}{\epsilon}\left(c_I + 4 \sum_{k\neq l}^{L} \|H_k\|^2_\infty \right) . \tag{2.12}$$

Proof. When moving into the interaction frame of a particular term H_l in H as in (1.23), we have

$$[H_p^I, H_q^I] = H_p^I H_q^I - H_q^I H_p^I = e^{iH_l t} H_p H_q e^{-iH_l t} - e^{iH_l t} H_q H_p e^{-iH_l t} = [H_p, H_q]^I .$$

Since the infinity norm is unitarily invariant, we then have that

$$\|[H_p, H_q]^I\|_\infty = \|[H_p, H_q]\|_\infty .$$

As the time-dependence came only from the $e^{iH_l t}$ terms, we can drop the maximization over times in $c_{\max}$. The sums in $c_{\max}$ will be over those indices $p, q \neq l$ and we define this simplified quantity as c_I as above.

The 1-norm in $\|H_k\|^2_{\infty,1}$ denotes an integral over an interval of measure Δt, so it again follows from the unitary invariance of the infinity-norm that $\|H_k\|^2_{\infty,1} = (\Delta t)^2 \|H_k\|^2_\infty$. Substituting $\Delta t = t/r$ into (2.2) then yields the desired expression. □

We can frame the complexity of the preceding process in terms of oracles defined as follows:

Definition 2.1.4. Let $H = \sum_k H_k$ be a *time-independent* Hamiltonian in $\mathbb{C}^{M\times M}$. We define oracles $\{W_k\}_{k=1}^L$ such that for each k, $W_k : \mathbb{R} \mapsto \mathbb{C}^{M\times M}$ with the action $W_k(\Delta) = e^{-iH_k\Delta}$.

These oracles can be used to implement the interaction frame transformation and the time evolution under specific summands of H at various fixed times. Equation (2.12) then gives an upper bound on the number of queries to the oracles W_k needed to ensure the simulation protocol is within error ϵ.

2.2 Hybrid Continuous qDRIFT and Qubitization Protocol

We would also like to consider the scenario where we simulate a time-*independent* Hamiltonian H with the following procedure:

1. Move into the interaction frame of a term H_j in H to turn the simulation problem into one involving a time-*dependent* interaction Hamiltonian $H_I(\tau)$ as in (1.23).

2. Use continuous qDRIFT to approximate the ideal time-evolution by the channel (1.2). Implementing this channel involves sampling from a probability distribution and yields a product of r time-*independent* terms of the form $\exp\left(-iH_I(\tau_k)/p(\tau_k)\right)$, where r is the number of sub-intervals of the whole simulation interval.

3. Use qubitization to simulate each time-independent term above and perform a singular value transformation to transform the spectrum in (1.12) and recover the original spectrum of H.

We first make the following definition:

Definition 2.2.1. Let $H = \sum_k w_k H_k$ be a time-independent Hamiltonian in $\mathbb{C}^{M\times M}$. We define an oracle W_j such that $W_j : \mathbb{R} \mapsto \mathbb{C}^{M\times M}$ with the action $W_j(\Delta) = e^{-iH_j\Delta}$

We use this oracle to transform to the interaction frame of a particular summand H_j in the Hamiltonian H in the following theorem:

Theorem 2.2.2 (Hybrid qDRIFT and Qubitization I.P. Simulation)**.** *Let $H = H_j + H_\alpha \in \mathbb{C}^{2^n\times 2^n}$ be a time-independent Hamiltonian such that H_α has an LCU decomposition $H_\alpha = \sum_{l\neq j}^{L} w_l H_l$, where $w_l \in \mathbb{R}^+$, and each w_l and H_l are obtained by oracles* PREPARE *and* SELECT *in* (1.6) *and* (1.7) *respectively.*

There exists a quantum algorithm such that for any $\epsilon, t > 0$, it implements a quantum channel Λ that is a $(1, O(\log L), \epsilon)$ block-encoding of e^{-iHt} using a number of queries to PREPARE*,* SELECT*, and $W_j(t)$ in*

$$O\left(\lambda_\alpha t + \left(\frac{\|H_\alpha\|_\infty^2 t^2}{\epsilon}\right) \frac{\log(\|H_\alpha\|_\infty t/\epsilon)}{\log\log(\|H_\alpha\|_\infty t/\epsilon)}\right), \tag{2.13}$$

where $\lambda_\alpha = \sum_{l\neq j} |w_l|$.

Proof. From Theorem 2.2, we have that for any positive integer r, there exists a division of $[0, t]$ where $0 = t_0 < t_1 < \cdots < t_k < \cdots < t_r = t$ such that (1.4) holds, where $\mathcal{E}(t, 0)$ and each $\mathcal{U}(t_k, t_{k+1})$ are understood as involving the interaction Hamiltonian $H_I(\tau)$ of (1.23).

By equation (1.4)

$$\left\| \mathcal{E}(t,0) - \prod_{j=0}^{r-1} \mathcal{U}(t_{j+1}, t_j) \right\|_\diamond \leq 4\frac{\|H_I\|_{\infty,1}^2}{r} .$$

From the relationship of the trace norm to the diamond norm in (A.3) and the mono-

tonicity of the Schatten p-norm, we get after choosing $r \geq 8\frac{\|H_I\|^2_{\infty,1}}{\epsilon}$ that

$$\begin{aligned}
&\left\| \mathcal{E}(t,0) - \prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right\|_\infty := \max_{\rho \in D_{2^n}} \left\| \mathcal{E}(t,0)\circ\rho - \left(\prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right) \circ \rho \right\|_\infty \\
&\leq \max_{\rho \in D_{2^n}} \left\| \mathcal{E}(t,0)\circ\rho - \left(\prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right) \circ \rho \right\|_1 = \left\| \mathcal{E}(t,0) - \prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right\|_\diamond \\
&\leq 4\frac{\|H_I\|^2_{\infty,1}}{r} \leq \frac{\epsilon}{2} \, .
\end{aligned} \tag{2.14}$$

Next, let $Q(t_{k+1},t_k)$ denote a channel which implements the three-step procedure outlined in the beginning of the section and let

$$\Lambda = \prod_{j=0}^{r-1} Q(t_{j+1},t_j) \, ,$$

We claim Λ is the desired channel. To show this, note that from Definition 1.1.3, we have upon fixing a signal state $|T\rangle = |0\rangle^m$ and setting $\alpha = 1$ (which can be done since we're implementing qubitization) that

$$\begin{aligned}
&\max_{\rho \in D_{2^n}} \|\mathcal{E}(t,0)(\rho) - (\langle 0|^m \otimes I_n)(\Lambda(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n)\|_\infty \\
&\leq \max_{\rho \in D_{2^n}} \left\| \mathcal{E}(t,0)(\rho) - \left(\prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right)(\rho) \right\|_\infty \\
&+ \max_{\rho \in D_{2^n}} \left\| \left(\prod_{j=0}^{r-1} \mathcal{U}(t_{j+1},t_j) \right)(\rho) - (\langle 0|^m \otimes I_n)(\Lambda(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n) \right\|_\infty \\
&\leq \frac{\epsilon}{2} + r \max_j \max_{\rho \in D_{2^n}} \|\mathcal{U}(t_{j+1},t_j)(\rho) - (\langle 0|^m \otimes I_n)(Q(t_{j+1},t_j)(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n)\|_\infty \, .
\end{aligned} \tag{2.15}$$

Recall that sampling from $\mathcal{U}(t_k,t_{k+1})$ yields a time-*independent* term $\exp(-iH_I(\tau_k)/p(\tau_k))$ where $\tau_k \in [t_k,t_{k+1}] \subset [0,t]$ is a specific time in some sub-interval $[t_k,t_{k+1}]$ at which $H_I(\tau)$ is being evaluated. The latter term above can thus be interpreted as the maximum spectral norm of the difference between an ideal implementation of the time-evolution operator for $t \in [t_k,t_{k+1}]$ and an implementation involving qubitization, maximized over all sub-intervals.

This can be made as small as desired via singular value transformation techniques discussed previously. Choosing

$$\max_j \max_{\rho \in D_{2^n}} \|\mathcal{U}(t_{j+1}, t_j)(\rho) - (\langle 0|^m \otimes I_n)(Q(t_{j+1}, t_j)(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n)\|_\infty \leq \frac{\epsilon}{2r} ,$$

we then have

$$\max_{\rho \in D_{2^n}} \|\mathcal{E}(t, 0)(\rho) - (\langle 0|^m \otimes I_n)(\Lambda(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n)\|_\infty \leq \frac{\epsilon}{2} + r\frac{\epsilon}{2r} = \epsilon .$$

We now define

$$\tilde{H}_i(\tau) = H_i(\tau)/p(\tau) ,$$

for $i \neq j$. Using the following identity which holds for all invertible matrices U

$$Ue^A U^\dagger = \exp(UAU^\dagger) , \tag{2.16}$$

we have

$$\begin{aligned} \exp(-iH_I(\tau)/p(\tau)) &= \exp\left(e^{iH_j\tau}\left(-i\sum_{i\neq j}\tilde{H}_i\right)e^{-iH_j\tau}\right) \\ &= e^{iH_j\tau}\left(\exp\left(\sum_{i\neq j} -i\tilde{H}_i\right)\right)e^{-iH_j\tau} . \end{aligned} \tag{2.17}$$

Each $\exp\left(-iH_I(\tau_k)/p(\tau_k)\right)$ term obtained from sampling $\mathcal{U}(t_{k+1}, t_k)$ can be expanded as in (2.17). Using the unitary invariance of the spectral norm, we have the simplification

$$\begin{aligned} p(\tau_k) = \frac{\|H_I(\tau_k)\|_\infty}{\|H_I(\tau)\|_{\infty,1}} &= \frac{\|e^{iH_j\tau_k}(\sum_{i\neq j} H_i)e^{-iH_j\tau_k}\|_\infty}{\int_{t_k}^{t_{k+1}} dt \|e^{iH_j\tau_k}(\sum_{i\neq j} H_i)e^{-iH_j\tau_k}\|_\infty} \\ &= \frac{\|\sum_{i\neq j} H_i\|_\infty}{\|\sum_{i\neq j} H_i\|_\infty \int_{t_k}^{t_{k+1}} dt} = \frac{1}{t_{k+1} - t_k} . \end{aligned}$$

Thus, we obtain a product of terms of the form

$$\exp(-iH_I(\tau_k)(t_{k+1} - t_k)) = e^{iH_j\tau_k}\exp\left(-i(t_{k+1} - t_k)\sum_{i\neq j} H_i\right)e^{-iH_j\tau_k} .$$

We then obtain the overall query complexity by summing (1.14) as applied to each subinterval $[t_k, t_{k+1}]$ from 0 to $r-1$ with error in the QSP transformation at most δ:

$$O\left(\sum_{k=0}^{r-1}\left(\lambda_\alpha(t_{k+1} - t_k) + \frac{\log(1/\delta)}{\log\log(1/\delta)}\right)\right) = O\left(\lambda_\alpha t + r\frac{\log(1/\delta)}{\log\log(1/\delta)}\right) . \tag{2.18}$$

Letting $\delta = O(\epsilon/r)$ for our choice of r in the above completes the proof. □

Lastly, we consider a hybrid Trotter, qDRIFT, and qubitization I.P. protocol which extends the results of Theorem 2.1.3 to include a qubitization step at the end to simulate all the resulting time-independent exponentials. This procedure is largely similar to that outlined in the beginning of the section but with an additional Trotter step:

1. Move into the interaction frame of a term H_j in H to turn the simulation problem into one involving a time-*dependent* interaction Hamiltonian $H_I(\tau)$ as in (1.23).

2. Use the Trotterization technique outlined in Section 1.1.3 to split the resulting time-ordered exponential into a product of L time-ordered exponentials, one for each summand in the Hamiltonian.

3. Use continuous qDRIFT to approximate each of the L time-ordered exponentials by the channel (1.2). Implementing this channel involves sampling from a probability distribution that yields a product of r time-*independent* terms of the form $\exp\left(-iH_I(\tau_k)/p(\tau_k)\right)$ for each of the L time-ordered exponentials.

4. Use qubitization to simulate the rL time-independent pieces and perform a singular value transformation to transform the spectrum in (1.12) and recover the original spectrum of H.

This yields the following theorem:

Theorem 2.2.3 (Hybrid Trotter, qDRIFT, and Qubitization I.P. Simulation)**.** *Let the assumptions of the previous theorem hold. There exists a quantum algorithm such that for any $\epsilon, t > 0$, it implements a quantum channel Γ that is a $(1, O(\log L), \epsilon)$ block-encoding of e^{-iHt} using a number of queries to* PREPARE*,* SELECT*, and $W_j(t)$ in*

$$O\left(\lambda_\alpha t + rL\frac{\log(rL/\epsilon)}{\log\log(rL/\epsilon)}\right), \tag{2.19}$$

where $\lambda_\alpha = \sum_{l\neq j}|w_l|$ and r is as in (2.12).

Proof. Let Γ denote a channel which implements the four-step procedure outlined above. Using the notation from the proof of the preceding theorem, we can express Γ as

$$\Gamma = \prod_{j=1}^{r} \prod_{k=1}^{L} Q_k(t_{j+1}, t_j)$$

where the subscript k denotes the quantum channel performing steps 3-4 above for a specific Hamiltonian term H_k.

Replicating the arguments of the preceding theorem, we can pick

$$\max_k \max_j \max_{\rho \in D_{2^n}} \left\| \mathcal{U}_k(t_{j+1}, t_j)(\rho) - (\langle 0|^m \otimes I_n)(Q_k(t_{j+1}, t_j)(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n) \right\|_\infty \leq \frac{\epsilon}{2rL} \,. \tag{2.20}$$

From Corollary 2.1.3, we can pick $r \geq \frac{2t^2}{\epsilon}\left(c_I + 4\sum_{k\neq l}^{L} \|H_k\|_\infty^2\right)$. Then from the triangle inequality, we have

$$\begin{aligned}
&\max_{\rho \in D_{2^n}} \left\| \mathcal{E}(t,0)(\rho) - (\langle 0|^m \otimes I_n)(\Gamma(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n) \right\|_\infty \\
&\leq \max_{\rho \in D_{2^n}} \left\| \mathcal{E}(t,0)(\rho) - \left(\prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}(t_{j+1}, t_j) \right)(\rho) \right\|_\infty \\
&+ \max_{\rho \in D_{2^n}} \left\| \left(\prod_{j=1}^{r} \prod_{k=1}^{L} \mathcal{U}(t_{j+1}, t_j) \right)(\rho) - (\langle 0|^m \otimes I_n)(\Gamma(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n) \right\|_\infty \\
&\leq \frac{\epsilon}{2} + rL \max_k \max_j \max_{\rho \in D_{2^n}} \left\| \mathcal{U}(t_{j+1}, t_j)(\rho) - (\langle 0|^m \otimes I_n)(Q(t_{j+1}, t_j)(|0\rangle\langle 0|^m \otimes \rho))(|0\rangle^m \otimes I_n) \right\|_\infty \\
&\leq \frac{\epsilon}{2} + rL\frac{\epsilon}{2rL} = \epsilon \,.
\end{aligned} \tag{2.21}$$

The overall query complexity is obtained by summing (1.14) as applied to each of the $\delta = t/r$ sized sub-intervals and summing over the magnitude of the coefficients in the interaction Hamiltonian. We then have, after choosing $\delta = \frac{\epsilon}{2rL}$ that

$$O\left(r \sum_{k=1}^{L} \left(\lambda_i \Delta t + \frac{\log(1/\delta)}{\log\log(1/\delta)} \right) \right) = O\left(\lambda_\alpha t + rL \frac{\log(rL/\epsilon)}{\log\log(rL/\epsilon)} \right) . \tag{2.22}$$

□

Note that the above methods can also be used to hybridize these simulation methods in the time-independent case. Unlike the Trotter-methods, the scaling of the query complexity is not substantially improved. Instead, any potential cost improvements to the simulation come from simplifications to PREPARE and SELECT.

Finally, we note that one can choose other combinations than an outer qDRIFT or Trotter loop and an inner qubitization loop. The first step in each of these hybrid procedures is to exploit the L^1-norm invariance of continuous qDRIFT by begining with a time-independent Hamiltonian and transforming into the interaction frame of a particular summand. This results in a time-dependent Hamiltonian, which cannot be simulated via qubitization and constrains us to use either Trotter or qDRIFT first. This still leaves open the possibility of whether trading an inner qDRIFT loop for another Trotterization procedure that decomposes the time-ordered exponentials to ordinary exponentials or randomly interleaving qDRIFT or Trotter procedures can result in additional speedups, and we leave such investigations for future work.

2.3 Hamiltonian Simulation of Schwinger Model

2.3.1 Schwinger Model and Query Complexity Bounds

We apply these ideas in simulating the Schwinger Model, quantum electrodynamics in 1+1 dimensions on a lattice [59, 60]. This model has been extensively used as an important stepping stone in simulations of lattice field theories using both tensor networks (see e.g. [61, 62]) and quantum devices (see e.g. Refs. [52, 63, 64]).

Using the Hamiltonian formulation of lattice gauge theory in the U(1) compact case [42, 86], the Hamiltonian of the model with $N-1$ links and $N/2$ spatial sites (half of which are electronic and half are positronic), is given by

$$H = H_E + H_h + H_M \tag{2.23}$$

with

$$H_E = \frac{g^2 a}{2} \sum_r E_r^2 \tag{2.24}$$

$$H_h = \frac{1}{2a} \sum_r U_r \psi_r^\dagger \psi_{r+1} - U_r^\dagger \psi_r \psi_{r+1}^\dagger \tag{2.25}$$

$$H_M = m \sum_r (-1)^r \psi_r^\dagger \psi_r, \tag{2.26}$$

where a is the lattice spacing, m the fermion mass, and g is the coupling constant. H_E can be interpreted as the electric energy given in terms of E_r, the integer-valued electric fields residing on the links. The remaining terms are expressed in terms of the fermionic operators ψ_r and $\psi_r^\dagger$ living on each site r, and the unitary link operators $U_r = e^{iaA_r}$ are expressed in terms of the gauge-potential $A_\mu = (0, A_1)$ in the temporal-gauge. H_h is a lattice analog of the minimal coupling of the Dirac fermionic field to the gauge field and H_M is the mass energy of the Dirac fermions, which are staggered based on the $(-1)^r$ factor.

We also have the following commutation relations between the link operators E_r and U_r

$$[E_r, U_s] = U_r \delta_{rs} \Rightarrow [E_r, U_s^\dagger] = -U_r^\dagger \delta_{rs}, \tag{2.27}$$

and between the fermionic creation and annihilation operators

$$\{\psi_r, \psi_s\} = \{\psi_r^\dagger, \psi_s^\dagger\} = 0 \tag{2.28}$$

$$\{\psi_r, \psi_s^\dagger\} = \delta_{rs}. \tag{2.29}$$

We can map the fermionic creation and annihilation operators in equations (2.25) and (2.26) onto a corresponding set of operators acting on spin degrees of freedom via the Jordan-Wigner transformation

$$\psi_r^\dagger = \frac{(X_r - iY_r)}{2} \prod_{j=1}^{r-1} Z_j. \tag{2.30}$$

Substituting the above into (2.25) and (2.26) and simplifying yields

$$H_h = \frac{1}{2a} \sum_{r=1}^{N-1} [U_r \sigma_r^- \sigma_{r+1}^+ + U_r^\dagger \sigma_r^+ \sigma_{r+1}^-]$$

$$= \frac{1}{8a}\sum_{r=1}^{N-1}[(U_r + U_r^\dagger)(X_rX_{r+1} + Y_rY_{r+1}) + i(U_r - U_r^\dagger)(X_rX_{r+1} - Y_rY_{r+1})] \tag{2.31}$$

and

$$H_M = \frac{m}{2}\sum_{r=1}^{N}(-1)^{r+1}Z_r. \tag{2.32}$$

Note that a factor of $I/2$ was dropped in the above equation since terms proportional to the identity in a Hamiltonian merely shift the spectrum by a constant. The derivation above also assumes open boundary conditions, but generalizations to periodic boundary conditions are straightforward. In that case the total number of links becomes N instead of $N-1$ and the asymptotic results we derive below for simulating the Schwinger Model remain unchanged.

It is customary to use the electric eigenbasis $|\epsilon\rangle_r$ for the infinite-dimensional Hilbert space of each link. In this basis, the E_r operator takes the diagonal form

$$E_r = \sum_\epsilon \epsilon|\epsilon\rangle_r\langle\epsilon|_r$$

and U_r takes the form

$$U_r = \sum_\epsilon |\epsilon+1\rangle\langle\epsilon|,$$

i.e. of a raising operator. Note that in order to map these degrees onto a quantum computer, it is customary to truncate the link Hilbert space by wrapping the electric field at a chosen cutoff Λ. This requires modifying the commutation relations in (2.27) but this issue is not directly relevant for our present work.

Since H_h and H_M are manifestly a sum of unitary operators, we can use the PREPARE and SELECT oracles from the qubitization simulation technique outlined previously. The overarching strategy is to move into the interaction frame of the H_E term, employ our hybrid protocols as outlined in the previous sections, and determine the query complexity in terms of the qubitization query model. The physical reasons for selecting the H_E term for the interaction picture is that the spectral norm of E_r is either large for a large cutoff Λ or unbounded in the strong coupling regime where $g \to \infty$. Choosing this term "removes" it from consideration in the interaction Hamiltonian as per equation (1.23). Additionally, since

the E_r operators are diagonal in its eigenbasis and the matrix elements are computable in polynomial time, the cost of simulating H_E in isolation is in $O(\text{poly}(n \log(1/\epsilon))$ [16]. This efficiency justifies the choice to consider such simulations as oracles in the prior discussion. As H_M commutes with the H_E term and is also 1-sparse, we may also opt to move into the combined interaction frame of the H_E and H_M terms. In this case, it will suffice to simulate only the H_h term via qubitization, and the simulation of this term will be the biggest asymptotic driver of the query complexity.

Recall that the PREPARE oracle acts on an empty ancilla register of $O(\log L)$ qubits, if L is the number of terms in the decomposition of the Hamiltonian into unitary operators, and prepares the superposition state

$$\text{PREPARE} \equiv \sum_{l=1}^{L} \sqrt{\frac{w_l}{\lambda}} |l\rangle\langle 0|,$$

where w_l denotes the coefficients of the terms in the decomposition of H_h and $\lambda = \sum_l |w_l|$ is the sum of the absolute value of the coefficients in the H_h term. Note that this oracle does not get altered when moving into the interaction frame since the coefficients w_l remain the same. Since there are $8(N-1)$ terms in the LCU decomposition of H_h, we may set $L = 8(N-1)$. There is only one type of coefficient in H_h in terms of magnitude, so $w_l/\lambda = 1/L$ and we obtain for our situation

$$\text{PREPARE} \equiv \frac{1}{\sqrt{L}} \sum_{l=1}^{L} |l\rangle\langle 0|. \tag{2.33}$$

As a result, we can scale every term in our Hamiltonian by a factor of $8a$ and scale the simulation time by a factor of $1/(8a)$.

On the other hand, a modification of the traditional select oracle is used to incorporate the interaction picture:

$$\begin{aligned} \text{SELECT}' &\equiv \sum_l |l\rangle\langle l| \otimes e^{i(H_E+H_M)t} H_l' e^{-i(H_E+H_M)t} \\ &= (I \otimes e^{i(H_E+H_M)t})(\sum_l |l\rangle\langle l| \otimes H_l')(I \otimes e^{-i(H_E+H_M)t}). \end{aligned} \tag{2.34}$$

This is merely the customary SELECT oracle but conjugated by the unitary operator $e^{iH_E t}e^{iH_M t}$ on the data qubits since H_E and H_M commute. It thus suffices to give circuit implementations of the usual SELECT oracle.

To summarize, since the conditions of Theorem 2.2.2 are satisfied, we have the following corollary:

Corollary 2.3.1 (Hybrid qDRIFT and Qubitization I.P Simulation for the Schwinger Model)**.** *Let $H = H_E + H_M + H_h$ be the Schwinger model Hamiltonian as given in (2.24), (2.25), and (2.26). Then we can perform the Hamiltonian simulation of H with the method of Theorem 2.2.2 using a total number of queries to PREPARE, SELECT, and $W_{H_M+H_E}(t)$ in*

$$O\left(\frac{N^2t^2}{a^2\epsilon}\frac{\log(Nt/a\epsilon)}{\log\log(Nt/a\epsilon)}\right). \tag{2.35}$$

Here, N is the number of sites in the system, a is the lattice spacing, $t \geq 0$ is the simulation time, and ϵ is the error quantifying the distance in 1-norm from the ideal time-evolution channel.

Proof. Note that H_h is manifestly a linear combination of unitary operators and that H_M and H_E are diagonal, and therefore 1-sparse, and commute with each other. Thus, the conditions of Theorem 2.2.2 are satisfied and we may move into the interaction frame of both the H_M and H_E terms.

To compute the explicit form of λ', note that H_h consists of a sum over $8(N-1)$ unitary terms each with a coefficient of absolute value $1/(8a)$. We therefore have $(N-1)/a$ for the overall sum. Thus

$$\lambda' = \frac{N-1}{a}. \tag{2.36}$$

Now note that $\|U\|_\infty = 1$ for any unitary operator U. Then we have $\|H_h\|_\infty \leq \lambda'$ by using the triangle-inequality and the sub-multiplicativity of the Schatten infinity norm. Substituting these relationships into (2.13) and retaining the dominant terms gives the claimed query complexity. □

Corollary 2.3.2 (Hybrid Trotter, qDRIFT, and qubitization I.P Simulation for Schwinger Model)**.** *Let $H = H_E + H_M + H_h$ be the Schwinger model Hamiltonian as given in (2.24), (2.25), and (2.26). Then we can perform the Hamiltonian simulation of the Schwinger model with the method of Theorem 2.2.3 using a number of queries to PREPARE, SELECT, and $W_{H_E+H_M}(t)$ in*

$$O\left(\frac{Nt^2}{a^2\epsilon}\frac{\log(Nt^2/(a^2\epsilon^2))}{\log\log(Nt^2/(a^2\epsilon^2))}\right). \tag{2.37}$$

Proof. We directly apply Corollary 2.1.3 to the situation where we move into the interaction frame of H_E and H_M, leaving only the H_h term of the Schwinger model remaining. For the terms in H_h, we use the notation $U_r\sigma_r^i\sigma_{r+1}^j$, where $i = 1$ or 2 so that $\sigma_r^1 = X_r$ and $\sigma_r^2 = Y_r$. Since $[U_r, U_s] = 0$ for all r, s and $[U_r, U_r^\dagger] = 0$ since U_r is unitary (and therefore normal), we need only focus on the commutators between the Pauli matrices in computing c_I. But since Pauli operators acting on different sites commute, we can further specialize to considering those terms that yield $[X_r, Y_r] = iZ_r$. Therefore, given a term $U_r\sigma_r^i\sigma_{r+1}^j$, it fails to commute with only $U_{r-1}\sigma_{r-1}^k\sigma_r^l$ and $U_{r+1}\sigma_{r+1}^m\sigma_{r+2}^n$, with analogous statements holding for $U_r^\dagger\sigma_r^i\sigma_{r+1}^j$. In other words, the terms involving a particular site r fail to commute with only those involving adjacent sites. Terms such as $[U_rX_rX_{r+1}, U_rY_rY_{r+1}]$ do not contribute since

$$[U_rX_rX_{r+1}, U_rY_rY_{r+1}] = U_r^2X_rY_rX_{r+1}Y_{r+1} - U_r^2Y_rX_rY_{r+1}X_{r+1} = 0,$$

where we've used in the anti-commutation relation twice $\{X_k, Y_k\} = 0$ to obtain the last equality.

Therefore, we can decompose H_h into "even" and "odd" pieces as follows:

$$H_h^{\text{even}} = \frac{1}{8a}\sum_{r=1}^{(N-1)/2}[(U_{2r} + U_{2r}^\dagger)(X_{2r}X_{2r+1} + Y_{2r}Y_{2r+1}) + i(U_{2r} - U_{2r}^\dagger)(X_{2r}X_{2r+1} - Y_{2r}Y_{2r+1})]$$

$$H_h^{\text{odd}} = \frac{1}{8a}\sum_{r=1}^{(N-1)/2}[(U_{2r-1} + U_{2r-1}^\dagger)(X_{2r-1}X_{2r} + Y_{2r-1}Y_{2r}) + i(U_{2r-1} - U_{2r-1}^\dagger)(X_{2r-1}X_{2r} - Y_{2r-1}Y_{2r})].$$

From the preceding discussion, given a particular term in the "odd" sum, there are exactly two terms in the even sum that fail to commute with it. In particular, there is exactly one

term in H_I^{even} with higher site index that fails to commute with it. Then by the definition of c_I,

$$c_I = \|[H_p^{\text{even}}, H_q^{\text{odd}}]\| = \frac{(N-1)}{2}\frac{1}{64a^2} = \frac{N-1}{128a^2}.$$

Similarly, since each term in H_h has norm 1, we have

$$4\sum_{r=1}^{N-1} \|H_h\|_\infty^2 \leq \frac{32(N-1)}{64a^2.}$$

Substituting these into (2.12) gives

$$r \geq \frac{65(N-1)t^2}{128a^2\epsilon}. \tag{2.38}$$

Substituting this and $L = 2$ into (2.19) and retaining the dominant terms gives

$$O\left(\frac{Nt^2}{a^2\epsilon}\frac{\log(Nt^2/(a^2\epsilon^2))}{\log\log(Nt^2/(a^2\epsilon^2))}\right) \tag{2.39}$$

as claimed. Note that these choices for the errors ensure that the total error for approximation of the ideal time-evolution channel via this entire procedure is $\epsilon/2+(2r)(\epsilon/4r) = \epsilon/2+\epsilon/2 = \epsilon$. □

Comparing the results of the preceeding corollaries, we see that Trotterizing first before applying qDRIFT can result in improvements in the query complexity in situations where the Hamiltonian has additional commutator structure that can be exploited. For unstructured problems, the additional Trotterization step is not generally useful.

2.3.2 Construction of Prepare and Select Oracles

We now give high-level circuit implementations of the aforementioned SELECT and PREPARE oracles for the H_h term of the Schwinger model.

We opt to employ a unary encoding of the control qubits $|c_i\rangle$ needed for the prepare and select oracles, i.e. $|i\rangle = |0...1...0\rangle$, where the 1 occurs on the i-th spot in the ket. Though this encoding requires a number of qubits linear in the number of terms in the LCU decomposition

rather than logarithmic for the implementation of the oracles, it greatly simplifies the control structures required within the circuits.

To implement the select oracle for H_h, we exploit certain patterns within the coefficients of the terms in H_h. Note that there are terms with $U_r^\dagger$ and U_r, phases of $\pm i$, and X_r and Y_r. We can switch between X and Y via the identity $SXS^\dagger = Y$. From techniques involving two's complement numbers, there exists a unitary operator Q that can flip U_r to $U_r^\dagger$ [87]. Lastly, the factors $\pm i$ can be inserted via suitable insertions of controlled-Z gate and controlled-S gate operations. The circuit that accomplishes this is given in Figure 2.1.

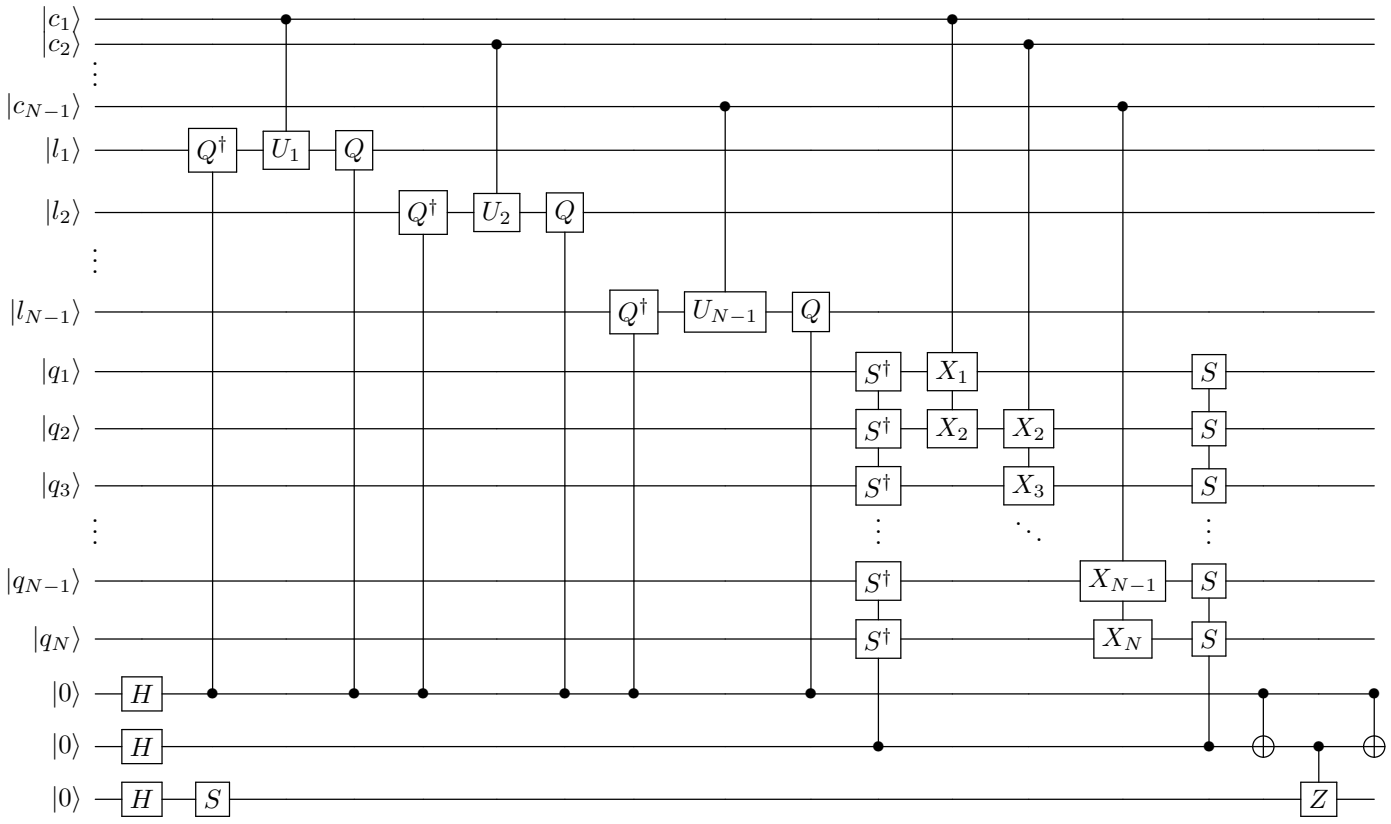


Figure 2.1: SELECT circuit for the H_h term in Schwinger Model

Here, $|c_r\rangle$ represent the control qubits used to implement the controlled operations, $|l_r\rangle$

the qubits corresponding to the links of the system, and $|q_r\rangle$ the qubits corresponding to the sites. Note how the control gate structure in this unary encoding is much more simple than what would have been required with a binary encoding. Though the latter encoding would have required only $\log(N-1)$ control qubits instead of $N-1$ as above, the advantage there is mitigated by the fact that numerous multi-controlled gates would have been required.

Since the terms in H_h for a given r only differ by coefficients of ± 1 or $\pm i$, these can be implemented via the insertion of a Z or S gate through the above constructions that exploit the aforementioned patterns in H_h. Note that the number of qubits needed to specify the state of link $|l_r\rangle$ will depend on the cut off for the electric energy term chosen. If our cutoff is Λ, then $|l_r\rangle$ will be a $\log \Lambda$-qubit state and U_r a $\log \Lambda$-qubit operator.

The circuit implementation of the PREPARE oracle reduces to preparing a uniform superposition state in binary, as per (2.33), and then converting the encoding to a unary one. The overall circuit with the general pattern is depicted in Figure 2.2 with $k = \log(8(N-1))$ ancilla control qubits. Note that since in the unary encoding an integer k is expressed as a state with a 1 in the k-th spot and 0's elsewhere, the X gate on $|0\rangle_1$ and the subsequent swaps have the effect of permuting the 1 to the appropriate position.

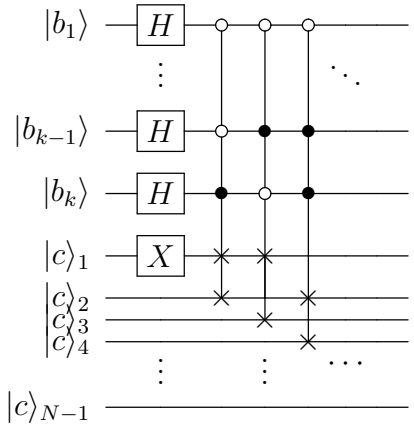


Figure 2.2: PREPARE circuit for the H_h term in Schwinger Model

The control structure on the bits b_j encoding the binary integers for the controlled-swap

gates is given precisely by the binary representation of that index. For example, since the binary integer $|00\ldots1\rangle$ gets mapped to $|01\ldots0\rangle$ in our unary encoding, we have to do a swap on the $|0\rangle_1$ and $|0\rangle_2$ qubits controlled on the first $k-1$ b qubits being 0 and b_k being 1.

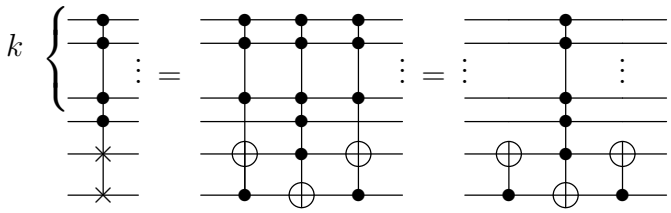
2.3.3 Gate Complexity Analysis

We now analyze the gate complexity per query to CTRL($\mathcal{W}$) of our simulation protocol and do so by analyzing the circuits given in Figure 2.1 and Figure 2.2. It suffices to express the complexity in terms of Toffoli gates since they dominate the computational complexity compared to Clifford operations.

Our analysis proceeds as follows:

1. First consider the PREPARE† and PREPARE operations in Figure 1.1. We make the approximation that they have roughly the same gate complexity and that it therefore suffices to determine the gate complexity of just the PREPARE circuit.

   From Figure 2.2, note that we have $N-1$ multiply-controlled swap gates since we needed to perform a binary-to-unary conversion to the $N-1$ qubits we have. Each can be converted to standard C^k(SWAP) by inserting X gates on either side of the 0 controls. Since we are assuming Pauli operations are approximately cost-free, it suffices to determine the gate complexity of these $N-1$ C^k(SWAP) gates. Standard circuit arguments show that the following identities hold:



   Thus, each C^k(SWAP) gate can be decomposed into a C^{k+1}(NOT) gate and 2 CNOT gates. From Corollary 1 in Ref. [88], we get that C^{k+1}(NOT) gate can be decomposed into $8(k+2)-24 = 8k-8$ Toffoli gates using only a single auxiliary qubit that can be

reused. This gives roughly $2(N-1)(8k-8)$ Toffoli gates that are needed for both the PREPARE and PREPARE† parts of Figure 1.1. Overall we have a gate complexity of

$$O(N \log N)$$

with 1 ancilla qubit needed.

The methods in Ref. [89] can be used to perform PREPARE circuit with $N \log N$ controlled swap gates, each of which can be decomposed into at most four non-Clifford operations using the techniques in Ref. [90]. This results in a smaller gate cost but ultimately does not affect the asymptotic gate complexity shown above.

2. The multiply-controlled Z gate in Figure 1.1 is controlled only on the $N-1$ control qubits that make up the unary encoding portion of Figure 2.2. Applying the above corollary again, we get a gate complexity of

$$O(N)$$

with 1 ancilla qubit needed.

3. To analyze the gate complexity of the controlled-SELECT operation in Figure 1.1, we first examine the operations that don't involve U_r and Q. The external control distributes among the operations involving the $2(N-1)$ controlled-S and controlled-$S^\dagger$ gates, $2(N-2)+2$ controlled-X_r gates, and the final 3 CNOT operations. Adding these together yields a total of $4N-1$ Toffoli gates, giving a gate complexity of

$$O(N)$$

with no ancilla qubits needed.

4. For the U_r gates, note that its action on states is that of an incrementer. This can be implemented by utilizing the quantum ripple-carry adder circuit given by Cuccaro in

Ref. [91]. Since we increment $\log\Lambda$-qubit numbers, the gate complexity is given in the thesis to be $2\log\Lambda - 1$ Toffoli gates, $5\log\Lambda - 3$ CNOTs, and $2\log\Lambda - 4$ negations.

Since we must perform a controlled-U_r operation for the walk operator, we get $(5\log\Lambda - 3)$ Toffoli gates, and $(2\log\Lambda - 1)$ C^3(NOT) gates. Again from Corollary 1 in Ref. [88], the latter is equivalent to $8(2\log\Lambda - 1)$ Toffoli gates with 1 extra qubit needed. Since we must perform $N-1$ of these U_r operations, we have altogether $(N-1)(21\log\Lambda - 11)$ Toffoli gates, giving a gate complexity of

$$O(N\log\Lambda)$$

with 1 ancilla qubit needed.

5. The Q and $Q^\dagger$ operations can be implemented with $O(\log(\Lambda))$ Toffoli gates each (see Ref. [87]), with the extra controls giving only constant pre-factors to the cost. Since there are $2(N-1)$ of these operations performed, we have a gate complexity of

$$O(N\log\Lambda)$$

6. Finally, the cost of performing the diagonal Hamiltonian simulation is $O(N\log^2(\Lambda))$ as the computation of the diagonal elements of the Hamiltonian involves squaring the input value, which can be performed in time $O(\log^2(\Lambda))$ (see Lemma 2 of Ref. [92]).

These considerations give us the following corollaries:

Corollary 2.3.3 (Gate Complexity for Hybrid qDRIFT and Qubitization I.P. Simulation of Schwinger Model)**.** *Let $H = H_h + H_M + H_E$ be the Schwinger model Hamiltonian as given in (2.24), (2.25), and (2.26). Then the Hamiltonian simulation of H can be performed using the method of Theorem 2.2.2 with a gate complexity in*

$$O\left(\frac{N^3t^2}{a^2\epsilon}\frac{\log(Nt/a\epsilon)}{\log\log(Nt/a\epsilon)}\log^2(N\Lambda)\right), \tag{2.40}$$

with an ancilla qubit overhead of O(1). In $\tilde{O}$ notation, the gate complexity is

$$\tilde{O}\left(\frac{N^3 t^2}{a^2\epsilon}\log^2\Lambda\right). \tag{2.41}$$

Proof. Summing up the scaling results from steps 1-6 as outlined above, we get a *per-query* gate complexity of

$$O(N\log^2(\Lambda)+N\log(N))\subseteq O(N\log^2(N\Lambda)). \tag{2.42}$$

Multiplying these by (2.35) and retaining the dominant terms yields the stated results. □

Corollary 2.3.4 (Gate Complexity for Hybrid Trotter, qDRIFT, Qubitization I.P. Simulation of Schwinger Model)**.** *Let $H = H_h + H_M + H_E$ be the Schwinger model Hamiltonian as given in* (2.24)*,* (2.25)*, and* (2.26)*. Then the Hamiltonian simulation of H can be performed using the method of Theorem 2.2.3 with a gate complexity in*

$$O\left(\frac{N^2t^2}{a^2\epsilon}\frac{\log(Nt^2/(a^2\epsilon^2))}{\log\log(Nt^2/(a^2\epsilon^2))}\log^2(\Lambda)\right), \tag{2.43}$$

with an ancilla qubit overhead of O(1). In $\tilde{O}$ notation, the gate complexity is

$$\tilde{O}\left(\frac{N^2t^2}{a^2\epsilon}\log^2(\Lambda)\right). \tag{2.44}$$

Proof. Multiplying (2.37) by $N\log(N\Lambda)$ gives the big-O cost. Dropping all sub-dominant logarithmic factors gives the $\tilde{O}$ scaling. □

2.3.4 Comparison with Trotterization

The results of (2.41) and (2.44) can be directly compared with the result given in Corollary 11 of Ref. [92] for using a second-order Trotter-Suzuki formula to perform the quantum simulation of the Schwinger model. We will first consider the regime in which simulations are carried at constant $1/(ga) = O(1)$ and for fixed $m/g = O(1)$. For this condition, we can use the result in Corollary 9 of Ref. [92] which after rescaling the time variable $T \to ag^2t/2$

to align with our normalization conventions for the Schwinger Model Hamiltonian, gives a total T-gate cost of

$$\tilde{O}\left(\frac{N^{3/2}t^{3/2}\Lambda a^{1/2}g^2}{\epsilon^{1/2}}\right) = \tilde{O}\left(\frac{N^{3/2}t^{3/2}\Lambda g^{3/2}}{\epsilon^{1/2}}\right) .$$

In the same regime, the result of (2.44) gives a gate complexity in

$$\tilde{O}\left(\frac{N^2t^2}{a^2\epsilon}\log^2(\Lambda)\right) = \tilde{O}\left(\frac{N^2t^2g^2}{\epsilon}\log^2(\Lambda)\right) .$$

We then see that the hybrid I.P. scheme provides a quasi-exponential speedup with respect to the electric cutoff Λ over the second-order Trotter-Suzuki approach, at the expense of a slightly worse scaling in all the other parameters (N, t, g, ϵ).

In order to extract physical observables however, it is important to consider that the number of sites N and the the lattice spacing cannot be chosen independently as the product $L = Na$ gives the physical size of the system. In past numerical simulations it was found that choosing $Nga = O(10)$ is appropriate for a large number of configurations (see e.g. Ref. [61]). In order to keep our derivation general, we will then introduce the dimensionless parameter $l = Nga$. In addition to the thermodynamic limit $Na \to \infty$, one also has to work with $ga \to 0$ in order to recover the continuum limit of the theory. The resulting gate complexity of the hybrid I.P. algorithm from Corollary 2.3.4 is found to be

$$\tilde{O}\left(\frac{l^2t^2}{g^2a^4\epsilon}\log^2(\Lambda)\right) ,$$

while for the second order Trotter-Suzuki scheme we have to consider two distinct regimes

- large cutoff limit $\Lambda ga > 1$, in which case the T-gate count is bounded by
$$\tilde{O}\left(\frac{l^{3/2}t^{3/2}\Lambda}{a^{3/2}\epsilon^{1/2}}\right) ,$$
- small lattice spacing limit $\Lambda ga < 1$, in which case the result of Corollary 9 of Ref. [92] does not hold anymore. The T-gate count can be found instead by using the more general result from Corollary 11 there, resulting in the Λ-independent scaling
$$\tilde{O}\left(\frac{l^{3/2}t^{3/2}}{g^{3/2}a^3\epsilon^{1/2}}\right) .$$

These results show that for fixed error ϵ and lattice extent l, the hybrid I.P. approach can be especially beneficial in the first regime thanks to the poly-logarithmic dependence on the cutoff Λ. In the small lattice spacing regime relevant for the continuum limit, the second order Trotter-Suzuki scheme developed in Ref. [92] has instead a better scaling with respect to all the parameters.

Finally, for the continuum limit it might be possible to improve the the gate complexity in some regimes by choosing to perform the I.P. simulation in the rotating frame given by the the interaction Hamiltonian H_h instead. Using the hybrid Trotter and qDRIFT scheme from Corollary 2.1.3, together with the implementation via qubitization of the time evolution operator for H_h derived in Section 2.3.2, this scheme has gate cost in

$$\tilde{O}\left(\frac{N^2 t}{a} + \frac{N^2 t^2}{\epsilon}\left(m^2 + g^4 a^2 \Lambda^4\right)\right).$$

For $m/g = O(1)$ and introducing the dimensional lattice size $l = Nga$, this becomes

$$\tilde{O}\left(\frac{l^2 t}{g^2 a^3} + \frac{l^2 t^2}{a^2 \epsilon} + \frac{l^2 t^2}{\epsilon} g^2 \Lambda^4\right).$$

This is clearly worse than either the Hybrid approaches discussed above or the second order Trotter-Suzuki scheme from Ref. [92] in the large cutoff limit $\Lambda g a > 1$, but can become competitive in the small lattice spacing limit $\Lambda g a < 1$ for some choices of (ϵ, l, Λ).

A detailed comparison of these different schemes to extract continuum quantities of physical interest in the Schwinger model with some target precision δ would require a more careful analysis of the scaling of the lattice size l and the electric field cutoff Λ, as well as a more careful consideration of the logarithmic factors hidden by the $\tilde{O}$ notation. We leave this interesting extension of the present work to future studies.

2.4 Collective Neutrino Oscillations

In extreme astrophysical environments, such as supernova explosions, neutrinos are present in such large densities that neutrino-neutrino interactions can become important to describe flavor evolution [93, 94]. These interactions are responsible for the appearance of collective

modes in flavor oscillations and have traditionally been studied with the help of a mean-field approximation (see e.g. Refs. [95, 96] for reviews). Due to the presence of interactions, many-body effects and quantum correlations could be important in understanding these phenomena and a number of studies is underway with a variety of techniques: from exact diagonalization [97] to Bethe-ansatz solutions [98], from tensor networks [99, 100] to digital quantum simulations [101, 102]. Quantum computing might offer a promising route to study these phenomena in situations where the entanglement entropy grows too fast with system size for tensor network simulations to remain feasible.

An important obstacle towards describing collective oscillations in realistic regimes is the fact that besides interactions with other neutrinos, scattering with external leptons (especially the abundant electrons) is an important effect *external* to the proto-neutron star. The matter interaction terms can become the dominant contributions in this regime, requiring very small time-steps for an accurate simulation of the flavor dynamics. On the quantum computing side, this requirement translates into a large number of gates required for the simulation and it is therefore important to design simulation algorithms with a gentle computational scaling with the external matter density.

The Hamiltonian of interest in the regime external to the star can be written as follows (see e.g. Ref. [103] for a derivation)

$$H = \sum_{i=1}^{N} \frac{\omega_i}{2} \vec{B} \cdot \vec{\sigma}_i + \frac{\lambda}{2} \sum_{i=1}^{N} Z_i + \frac{\mu}{2N} \sum_{i<j}^{N} J_{ij} \vec{\sigma}_i \cdot \vec{\sigma}_j \ . \tag{2.45}$$

Here $\vec{\sigma}_i$ is the vector of Pauli matrices acting on the i-th qubit and the single particle energies ω_i are positive for neutrinos and negative for anti-neutrinos. The coupling matrix J_{ij} takes values in $[0, 2]$ and the normalized vector $\vec{B}$ contains the vacuum mixing angle as $\vec{B} = (\sin(2\theta), 0, -\cos(2\theta))$. The constants are given by $\lambda = \sqrt{2} G_F n_e$ and $\mu = \sqrt{2} G_F n_\nu$, with G_F Fermi's constant and n_e and n_ν the electron and neutrino densities respectively. In typical situations the electron contribution λ is the dominant term. A standard approach to deal with this problem is to move to the rotating frame defined by the unitary $U_e(t) =$

$\exp(-i\frac{t}{2}\lambda\sum_{i=1}^{N} Z_i)$ and define the Hamiltonian in the interaction picture as

$$\begin{aligned} H(t) &= U_e^\dagger(t) H U_e(t) - i U_e^\dagger(t)\frac{\partial}{\partial t} U_e(t) \\ &= \sin(2\theta)\sum_{i=1}^{N}\frac{\omega_i}{2}\left(\cos(\lambda t)X_i - \sin(\lambda t)Y_i\right) - \cos(2\theta)\sum_{i=1}^{N}\frac{\omega_i}{2}Z_i + \frac{\mu}{2N}\sum_{i<j}^{N} J_{ij}\vec{\sigma}_i\cdot\vec{\sigma}_j \\ &= e^{i\lambda\sum_i Z_i t} H_\nu e^{-i\lambda\sum_i Z_i t}, \end{aligned} \tag{2.46}$$

where

$$H_\nu = \sum_{i=1}^{N}\frac{\omega_i}{2}\vec{B}\cdot\vec{\sigma}_i + \frac{\mu}{2N}\sum_{i<j}^{N} J_{ij}\vec{\sigma}_i\cdot\vec{\sigma}_j\ . \tag{2.47}$$

Typically only the leading order contribution in the Magnus expansion is retained, giving the time-independent Hamiltonian

$$H_0 = -\cos(2\theta)\sum_{i=1}^{N}\frac{\omega_i}{2}Z_i + \frac{\mu}{2N}\sum_{i<j}^{N} J_{ij}\vec{\sigma}_i\cdot\vec{\sigma}_j\ . \tag{2.48}$$

In this limit, flavor states will not experience oscillations and typically this is solved by defining the flavor axis to be rotated by a small phenomenological amount away from the Z axis. It would be desirable however to be able to exercise more control in this approximation. Expansions to high orders in the Magnus expansion quickly produce higher order interactions which will complicate the implementation of the corresponding time-independent evolution. Here we use the time-dependent algorithm described above to work directly in the interaction picture without introducing uncontrollable errors.

2.4.1 Trotter Suzuki Approximations in Interaction Frame

As a first step, let us consider simulating the Hamiltonian in the interaction frame using a k^{th}-order Trotter-Suzuki formula such as those in Ref. [17]. To do this, we need to introduce a notion of the typical energy scale of the time-dependent Hamiltonian with respect to the Trotter decomposition of the interaction frame Hamiltonian. If we use a conventional Trotter decomposition, as opposed to (1.18), we find that the error incurred from using a first-order Trotter formula for an ordered operator exponential $U_1(t)$ formed by evaluating

the Hamiltonian at $t = 0$ and then Trotterizing the resultant ordinary operator exponential is

$$\left\| \exp_\tau \left(-i \int H(t) \mathrm{d}t \right) - U_1(t) \right\|_\infty \in O((\max_t \|H'(t)\|_\infty + \sum_{p,q} \max_t \|[H_p(t), H_q(t)]\|_\infty) t^2), \tag{2.49}$$

where the specific constants can be found using the techniques in Ref. [85]. The derivative of the Hamiltonian in the interaction frame is in

$$\|H'(t)\|_\infty \in O(\theta N \lambda). \tag{2.50}$$

The commutator sum similarly obeys

$$\begin{aligned} \| \sum_{p,q} \max_{t,t'} \|[H_p(t'), H_q(t)]\|_\infty &\in O\left(N\omega\theta + \mu N\theta + \omega N\mu + N\mu^2 \right) \\ &= O\left(N(\theta(\omega + \mu) + \mu(\omega + \mu)) \right) \\ &\subseteq O\left(N\mu^2 \right), \end{aligned} \tag{2.51}$$

where $\omega = \max_i |\omega_i|$ and in the last term we take $\mu \gg \omega$.

The overall error in the simulation is therefore

$$O\left((N\mu^2 + \theta N \lambda) t^2 \right). \tag{2.52}$$

If we break the overall evolution into r time slices, then it follows that the error in the simulation can be made at most ϵ by choosing

$$r \in O\left(\frac{N(\mu^2 + \theta\lambda) t^2}{\epsilon} \right). \tag{2.53}$$

As there are $O(N^2)$ operator exponentials per time step, the total number of operator exponentials needed to perform the simulation is

$$N_{\mathrm{exp}} \in O\left(\frac{N^3(\mu^2 + \theta\lambda) t^2}{\epsilon} \right). \tag{2.54}$$

Since each operator exponential requires $O(1)$ gates from the H, R_z, CNOT gate library, the gate complexity is also proportional to this [16]. Interestingly, using the swap-network

protocol from Ref. [101] (and inspired from their fermionic variant [104]), this cost is not affected by limited connectivity in the device despite the interaction being all-to-all. This cost also coincides with the optimal scaling with λ permitted by the no-fast forwarding theorem [33], despite being a low-order formula that has inferior scaling with respect to the other parameters relative to alternative simulation methods.

Higher-order *time-dependent* Trotter formulas for the simulation can be used, but the advantage gleaned by using them with respect to the λ scaling is less clear. Such algorithms scale with the parameter [17]

$$\Lambda_k^{k+1}/r^k = \max_{j\leq k}(\|\partial_t^j H(t)\|_\infty^{1/j+1})^{k+1}/r^k \in O(\lambda^k/r^k), \tag{2.55}$$

where k represents the order of the Trotter formula. It then follows that these formulas ultimately lead to the same linear scaling in the gate complexity with λ (assuming that $\theta \in \Theta(1)$). By contrast, if we did not use the interaction picture algorithm, the cost of simulation using the k^{th}-order Trotter *time-independent* formula would scale as $O(\lambda^{1+1/2k})$ [16, 17]. This illustrates that for problems with an imbalance in the scales of the operators, switching to an interaction frame can be beneficial at virtually no cost overhead.

2.4.2 Simulating Neutrino Oscillations using Hybrid Trotter-qDRIFT

Now we will apply Corollary 2.1.3 to compare this cost to that required by the hybrid Trotter and continuous qDRIFT algorithm. Specifically, the error in an r-segment simulation is of the form (under the assumption that $\mu \gg \omega$)

$$\frac{t^2}{r}\left(c_I + 4\sum_{k\neq l}^{L}\|H_k\|_\infty^2\right) \in O\left(\frac{N\mu^2 t^2}{r}\right). \tag{2.56}$$

As each segment of qubitization requires application of a first order Trotter formula, the cost per segment in terms of operator exponentials scales as $O(N^2)$. Thus if we demand that the error is at most ϵ, the cost is

$$N_{\text{exp}} \in O(N^2 r) \subseteq O\left(\frac{N^3\mu^2 t^2}{\epsilon}\right), \tag{2.57}$$

wherein each operator exponential requires $O(1)$ applications of H, R_z and CNOT. This shows that the above asymptotic scaling applies in the gate complexity as well as the number of exponentials.

Interestingly, in the limit where $\lambda \gg 1$, this result provides better scaling than even the gate complexity of the truncated Dyson series [105, 106] which scales in the interaction frame as $\log(\lambda)$. In our case, the quantum computational complexity is completely independent of λ. Of course, poly-logarithmic costs with these algorithms need to be incurred at the classical side to compute the rotation angles that go into the single qubit rotations but such costs are assumed to be negligible in our cost model. This implies that for such cases where the cost of the simulation is gated by the cost of preparing and controlling from the time-register, switching to a method that only requires classical controls can allow us to outperform such methods and make the gate count (rather than just the query complexity [106]) independent of the magnitude of the norm of the interaction Hamiltonian.

As a final note, similar scaling can also be attained by using the approach of Ref. [18] to time-order the operator exponentials that we use in the interaction frame. The performance of this method is summarized in (1.18) and gives an alternative to the hybrid approach considered here and yields comparable scaling with λ.

2.5 Constrained Hamiltonian Dynamics

As a final application of these techniques, let us consider the application of quantum simulation to dynamics subject to dynamical constraints. Specifically, we will consider a Hamiltonian of the form

$$H = H_f + \lambda P_c, \tag{2.58}$$

where $H_f \in \mathbb{C}^{2^n \times 2^n}$ is the free Hamiltonian and $P_c \in \mathbb{C}^{2^n \times 2^n}$ is a projector onto an infeasible region. The idea behind our approach to simulating constrained quantum dynamics is that if we choose $\lambda \gg \|H_f\|_\infty$ and an initial state $|\psi\rangle = (I - P_c)|\psi\rangle$, then the dynamics of the quantum system will, up to small errors, be confined to within the dynamically feasible region specified by the null-space of P_c. Note that this result is reminiscent of others in the

literature such as Refs. [107, 108]; however, this result is specialized to time evolution and is simpler to employ in this context.

Lemma 2.5.1. *Let $H_f \in \mathbb{C}^{2^n \times 2^n}$ be a free Hamiltonian for a system and let λ be a variable describing the strength of the constraint such that $\|H_f\|_\infty \ll \lambda$. We then have that for any $|\psi\rangle$ in the null-space of P_c,*

$$\|e^{-i(H_f-\lambda P_c)t}|\psi\rangle - \lim_{\lambda\to\infty} e^{-i(H_f-\lambda P_c)t}|\psi\rangle\|_2 \in O\left(\frac{\|H_f\|_\infty^2 t}{\lambda}\right)$$

where $\|\cdot\|_2$ refers to the vector 2-norm.

Proof. In order to show the deviation in each eigenvector of the Hamiltonian that arises from adding the small Hamiltonian H_f to the constraint term, we will introduce

$$H_f(x) := H_f x + \lambda P_c, \tag{2.59}$$

where $x \in [0,1]$. For $x = 0$, $H_f(0) = \lambda P_c$ has a degenerate null-space denoted $\mathcal{P}^0$. Let $|v_j(x)\rangle$ denote the eigenvectors of $H_f(x)$ with corresponding eigenvalues $E_j(x)$. We then have from perturbation theory that the derivative of $|v_j(x)\rangle$ is

$$\frac{\partial|v_j(x)\rangle}{\partial x} = \sum_{k\neq j} |v_k(x)\rangle \frac{\langle v_k(x)|H_f|v_j(x)\rangle}{E_j(x) - E_k(x)} \tag{2.60}$$

Assuming that the eigenvalue gaps are non-zero, we further have that the second derivative is finite. From the definition of a Riemann integral, we get

$$|v_j(1)\rangle = \int_0^1 \frac{\partial|v_j(x)\rangle}{\partial x}\mathrm{d}x = |v_j(0)\rangle + \lim_{r\to\infty}\sum_{p=2}^{r}\sum_{k\neq j} |v_k((p-1)/r)\rangle \frac{\langle v_k((p-1)/r)|H_f|v_j((p-1)/r)\rangle}{E_j((p-1)/r) - E_k((p-1)/r)}\frac{1}{r}. \tag{2.61}$$

This expression allows us to relate the shift in the eigenvectors recursively. First let us consider the initial time step. As $H_f(x)$ is degenerate at $x = 0$, we can choose the eigenvectors such that the matrix with components $\langle v_j(0)|H_f|v_k(0)\rangle$ is a diagonal matrix for all $|v_j(0)\rangle, |v_k(0)\rangle \in \mathcal{P}^0$ or $|v_j(0)\rangle, |v_k(0)\rangle \in \mathcal{P}^{0\perp}$. In the former case we have that $(I-P_c)|v_j(0)\rangle = |v_j(0)\rangle$, so $\langle v_j(0)|H_f|v_k(0)\rangle = \langle v_j(0)|(I-P_c)H_f(I-P_c)|v_k(0)\rangle$. Thus we can achieve the diagonal criteria by choosing each $|v_j(0)\rangle$ to be an eigenvector of $(I-P_c)H_f(I-P_c)$. Similarly,

we can achieve the diagonal criteria for each vector in $\mathcal{P}^{0\perp}$ by choosing each $|v_j\rangle$ to be an eigenvector of $P_c H_f P_c$. We then have that for any $|v_j(0)\rangle$ in $\mathcal{P}^0$,

$$\begin{aligned}|v_j(1/r)\rangle &= |v_j(0)\rangle + \frac{1}{r}\sum_{k\neq j}|v_k(0)\rangle\frac{\langle v_k(0)|H_f|v_j(0)\rangle}{E_j(0)-E_k(0)} \\ &= |v_j(0)\rangle - \sum_{k:|v_k(0)\rangle\neq\mathcal{P}^0}|v_k(0)\rangle\frac{\langle v_k(0)|H_f|v_j(0)\rangle}{\lambda r}\ . \end{aligned} \tag{2.62}$$

In turn

$$\||v_j(0)\rangle - |v_j(1/r)\rangle\|_2 \in O(\|H_f\|_\infty/\lambda r). \tag{2.63}$$

Furthermore from Ref. [109], we have that for all k, $|E_k(x)-E_k(0)| \leq x\|H_f\|_\infty$. Now let us assume that for some integer $q \geq 0$

$$\||v_j(0)\rangle - |v_j(q/r)\rangle\|_2 \in O(q\|H_f\|_\infty/\lambda r)\ . \tag{2.64}$$

We then have

$$|v_j((q+1)/r)\rangle = |v_j(q/r)\rangle + \frac{1}{r}\sum_{k\neq j}|v_k(q/r)\rangle\frac{\langle v_k(q/r)|H_f|v_j(q/r)\rangle}{E_j(q/r)-E_k(q/r)} \tag{2.65}$$

and therefore

$$\||v_j((q+1)/r)\rangle - |v_j(q/r)\rangle\|_2 \in O\left(\frac{\|H_f\|_\infty}{(\lambda-2\|H_f\|_\infty)r}\right) = O\left(\frac{\|H_f\|_\infty}{\lambda r}\right). \tag{2.66}$$

Thus we have

$$\begin{aligned}\||v_j((q+1)/r)\rangle - |v_j(0)\rangle\|_2 &\leq \||v_j((q+1)/r)\rangle - |v_j(q/r)\rangle\|_2 + \||v_j(0)\rangle - |v_j(q/r)\rangle\|_2 \\ &\in O\left(\frac{(q+1)\|H_f\|_\infty}{\lambda r}\right)\ . \end{aligned} \tag{2.67}$$

This in turn shows us that

$$\||v_j(1)\rangle - |v_j(0)\rangle\|_2 \in O\left(\frac{\|H_f\|_\infty}{\lambda}\right). \tag{2.68}$$

Next by examining the differential equation for the eigenvalues, we have that the corresponding eigenvalue $E_j(1)$ obeys

$$\begin{aligned}E_j(1) &= E_j(0) + \int_0^1 \frac{\partial E_j(x)}{\partial x}\mathrm{d}x = \int_0^1 \frac{\partial E_j(x)}{\partial x}\mathrm{d}x \\ &= \int_0^1 \langle v_j(x)|H_f|v_j(x)\rangle \mathrm{d}x = \langle v_j(0)|H_f|v_j(0)\rangle + O\left(\frac{\|H_f\|_\infty^2}{\lambda}\right)\ . \end{aligned} \tag{2.69}$$

Similarly for any $|v_j(0)\rangle \in \mathcal{P}^{0\perp}$, $E_j(1) = \lambda + \langle v_j(0)|H_f|v_j(0)\rangle + O\left(\frac{\|H_f\|_\infty^2}{\lambda}\right)$. We therefore have from the triangle inequality that

$$\|H(1) - \sum_k (\lambda \delta_{|v_k\rangle \in \mathcal{P}^{0\perp}} + \langle v_k(0)|H_f|v_k(0)\rangle |v_k(0)\rangle\langle v_k(0)|)\|_\infty$$
$$\in O\left(\frac{\|H_f\|_\infty^2}{\lambda}\right) . \tag{2.70}$$

We therefore have from the fact that $\|e^{-iHt} - e^{-iH't}\|_\infty \leq \|H - H'\|_\infty t$ for all Hermitian matrices H and H' of equal dimension that

$$\|e^{-iH(1)t} - e^{-i\sum_k \left(\lambda \delta_{|v_k\rangle \in \mathcal{P}^{0\perp}} + \langle v_k(0)|H_f|v_k(0)\rangle |v_k(0)\rangle\langle v_k(0)|\right)t}\|_\infty \in O\left(\frac{\|H_f\|_\infty^2 t}{\lambda}\right) . \tag{2.71}$$

Therefore for any $|\psi\rangle \in \mathcal{P}^0$ we have that

$$\|e^{-iH(1)t}|\psi\rangle - \lim_{\lambda\to\infty} e^{-iH(1)t}|\psi\rangle\|_2 \in O\left(\frac{\|H_f\|_\infty^2 t}{\lambda}\right) . \tag{2.72}$$

□

This shows that we can simulate constrained dynamics for time t within error ϵ by choosing $\lambda \geq \|H_f\|_\infty^2 t/\epsilon$. This in turn leads to a substantial degradation of the scaling of most simulation algorithms if $[H_f, P_c] \neq 0$, because the Hamiltonian's norm scales with both the evolution time and the uncertainty desired in the simulation. This makes such constrained dynamics impractical for many applications.

This drawback can, however, be mitigated through the use of an interaction frame transformation. By transforming to the interaction frame of the constraint, we can perform the simulation at cost that is (in some cases) independent of the choice of λ. The cost of such simulations using a hybrid qubitization and qDRIFT algorithm is given below. We cite the complexity of this algorithm rather than truncated Dyson methods because such methods explicitly have a cost that scales logarithmically with λ; whereas in some cases the quantum gate complexity will be independent of λ.

Theorem 2.5.2. *Let the assumptions of Theorem 2.2.2 hold. Then there exists a quantum algorithm that implements, for any $t > 0$ and $\epsilon > 0$, a quantum channel that is a*

$(1, O(\log(L)), \epsilon)$ block encoding of $e^{-i(H_f+\lambda P_c)t}$. Further this implementation requires a total number of queries to PREPARE, SELECT and W_{P_c} in

$$O\left(\beta t + \left(\frac{\|H_f\|_\infty^2 t^2}{\epsilon}\right) \frac{\log(\|H_f\|_\infty t/\epsilon)}{\log\log(\|H_f\|_\infty t/\epsilon)}\right).$$

Proof. The proof follows directly from previous results. Specifically we have that

$$\begin{aligned} &\|V|\psi\rangle - \lim_{\lambda\to\infty} e^{-i(H_f+\lambda P_c)t}\|_2 \\ &\leq \|V|\psi\rangle - e^{-i(H_f+\lambda P_c)t}|\psi\rangle\|_2 + \|e^{-i(H_f+\lambda P_c)t}|\psi\rangle - \lim_{\lambda\to\infty} e^{-i(H_f+\lambda P_c)t}|\psi\rangle\|_2 \,. \end{aligned} \tag{2.73}$$

From Theorem 2.2.2, we have that the number of queries to PREPARE, SELECT and W_{P_c} needed to implement a $(1, O(\log(L)), \epsilon/2)$ block encoding is in

$$O\left(\beta t + \left(\frac{\|H_f\|_\infty^2 t^2}{\epsilon}\right) \frac{\log(\|H_f\|_\infty t/\epsilon)}{\log\log(\|H_f\|_\infty t/\epsilon)}\right).$$

Next, from Theorem 2.5.1 we have that there exists a value of $\lambda \in O(\|H_f\|_\infty^2 t/\epsilon)$ such that $\|e^{-i(H_f+\lambda P_c)t}|\psi\rangle - \lim_{\lambda\to\infty} e^{-i(H_f+\lambda P_c)t}|\psi\rangle\|_2 \leq \epsilon/2$. The result then follows from the triangle inequality. □

These results show that query efficient methods exist for simulating Hamiltonian dynamics; however, the existence of a query efficient algorithm for simulating dynamics subject to a particular constraint does not imply the existence of a gate efficient algorithm. For example, let us consider the case where $P_c|x\rangle = |x\rangle$ if and only if $E(x) \leq \delta$ for some $\delta > 0$ and $E(x)$ is the energy function for an arbitrary Ising model. Since this problem is NP-hard [110], a gate efficient version of this constraint is only possible if NP $\subseteq$ BQP, which is strongly believed to be false. For this reason, we provide below a sufficient, but not a necessary, condition for the W_{P_c} to be simulatable in $O(\mathsf{polylog}(2^n \lambda t/\epsilon))$ gate operations.

Proposition 2.5.3. *Let $P_c \in \mathbb{C}^{2^n \times 2^n}$ be a projector matrix. Suppose there exist functions such that for any $x, y \in \mathbb{Z}_{2^n}$, $g(x, y) = \langle x|P_c|y\rangle$ and $f(x, i)$ yields the column index of the i^{th} non-zero matrix element of P_c as represented in the computational basis. If*

1. *P_c has at most 1 non-zero matrix elements in each row when expressed in the computational basis.*

2. *f is computable using a number of quantum gates that are in $O(\mathsf{poly}(n))$ and g within error 2^{-m} using $O(\mathsf{poly}(nm))$ quantum operations*

then for any $\lambda \geq 0$ and $t \geq 0$, a unitary $\tilde{U}$ can be constructed such that $\|\tilde{U} - U_I(\lambda; t)\|_\infty \leq \epsilon$ using $O(\mathsf{polylog}(2^n \lambda t/\epsilon))$ quantum operations.

Proof. The proof follows straight forwardly. If we assume that $g(x, y)$ can be implemented within zero error with m bits of precision, we have from Ref. [111] that $e^{-i\lambda P_c t}$ can be implemented with zero error using $O(1)$ applications of f and g as well as $O(\text{poly}(nm))$ auxillary quantum operations.

Now let us assume that $g(x, y)$ cannot be computed within zero error using $m < \infty$ bits of precision. If we denote $\tilde{g}(x, y)$ to be the approximate version of g, we have from the fact that P_c is one-sparse that

$$\begin{aligned}
&\|e^{-i\lambda t \sum_x g(x,f(x,1))|x\rangle\langle f(x,1)|} - e^{-i\lambda t \sum_x \tilde{g}(x,f(x,1))|x\rangle\langle f(x,1)|}\|_\infty \\
&= \max\Bigg(\max_{x:x\neq f(x,1)} \|e^{-i\lambda t g(x,f(x,1))(|x\rangle\langle f(x,1)| + |f(x,1)\rangle\langle x|)} - e^{-i\lambda t \tilde{g}(x,f(x,1))(|x\rangle\langle f(x,1)| + |f(x,1)\rangle\langle x|)}\|_\infty \\
&\qquad\qquad , \max_{x:x=f(x,1)} \|e^{-i\lambda t g(x,f(x,1))|x\rangle\langle x|} - e^{-i\lambda t \tilde{g}(x,f(x,1))|x\rangle\langle x|)}\|_\infty \Bigg) \\
&\leq \lambda t \max_x |g(x, f(x, 1)) - \tilde{g}(x, f(x, 1))| \leq \lambda t 2^{-m} \, .
\end{aligned} \tag{2.74}$$

Thus to achieve an error of ϵ, we need to take $m \in O(\log(\lambda t/\epsilon))$. The result immediately follows from the assumptions on the cost of f and g. □

There are a number of applications of this approach to solve constrained versions of quantum dynamics. One such application involves the simulation of quantum field theories within a particular gauge, which describes a choice of a dynamically unobservable feature of the system that is needed to unambiguously determine the dynamics. For example, the

Lorenz gauge involves choosing the vector potential such that $\partial_\mu A^\mu = 0$. Rather than fixing the gauge by a clever choice of an equation of motion, this approach allows us to impose such gauges by penalizing all configurations that violate this.

Another application involves Gauss' law in quantum electrodynamics [52, 112]. For D-dimensional quantum electrodynamics, Gauss' law reads $\nabla \cdot \hat{E}(s) - \hat{\rho}(s) = 0$, where $\hat{E}(s)$ is the electric field operator at position s and $\hat{\rho}(s)$ is the charge density there. On a lattice, this can be further simplified to $G(s) := \sum_{i=1}^{D}(\hat{E}(s) - \hat{E}(s-e_i)) - \sum_{s,\sigma} e_\sigma n_\sigma(s)$, where $n_\sigma(s)$ is the number of electrons (or positrons) at a site and e_σ is ± 1 depending on the site. From this, the constraint projector P_c can be expressed using the properties of discrete Fourier transforms as $P_c = I - \frac{1}{N}\sum_{i=1}^{N} e^{-i2\pi G(s)/N}$ [112]. This is relevant because Gauss' law is only approximately held for methods such as Trotter-Suzuki simulations and so the application of this constraint oracle can be used to filter out the unphysical components of simulation error. Specifically, consider a constraint on $|\psi\rangle \in \mathcal{P}^0$ given by the projector P_c that commutes with the Hamiltonian. In other words, $[P_c, \sum_j H_j] = 0$; however there may exist k' such that $[H_{k'}, P_c] \neq 0$. This creates problems for the Trotter-Suzuki expansion, but we can address this by transforming into the interaction frame as discussed below

$$e^{-i\sum_{j=1}^{M} H_j t}|\psi\rangle = e^{-i(\sum_{j=1}^{M} H_j + \lambda P_c)t}|\psi\rangle = e^{-i\lambda P_c t}\exp_\tau\left(\int_0^t e^{i\lambda P_c s}\sum_j H_j e^{-i\lambda P_c s}\mathrm{d}s\right)|\psi\rangle. \quad (2.75)$$

We can then implement the time-ordered operator exponential in (2.75) using one of our previous methods, such as a hybridized Trotter-qDRIFT method or that used in the previous section. This allows us to impose a constraint, such as Gauss' law, on the integration formula at low cost. By contrast, if we were to try to do so using a high-order Trotter formula, we would have remainder terms in the Trotter-Suzuki expansion that are in $O(\mathsf{poly}(\lambda t))$. From Theorem 2.5.1, this is in $O(\mathsf{poly}(\|H_f\|_\infty^2 t^2/\epsilon))$ and thus cannot be implemented at low cost in the limit where $\epsilon \ll 1$, unlike in the interaction picture approach.

A similar simple example of this is solving the Schrodinger equation for a particle constrained to a given surface. As an example, consider solving the Schrodinger equation for a particle constrained to be on the surface of a Figure-8 immersion of a Klein bottle, which is

a non-orientable surface with no boundary. Such a surface is given, for some fixed value of $r > 2$, by the following parameterized surface over the angles $\theta, v \in [0, 2\pi)$

$$\begin{aligned} x &= (r + \cos(\theta/2)\sin(v) - \sin(\theta/2)\sin(2v))\cos(\theta) \\ y &= (r + \cos(\theta/2)\sin(v) - \sin(\theta/2)\sin(2v))\sin(\theta) \\ z &= \sin(\theta/2)\sin(v) + \cos(\theta/2)\sin(2v), \end{aligned} \tag{2.76}$$

where in Cartesian coordinates $\theta = \arctan(y/x)$ and v is found implicitly through the above expressions. As all these coordinate functions are Lipshitz continuous, a least square solution can be found using gradient descent after dividing up the surface in (θ, v) coordinates into a finite number of regions and then performing gradient descent of $\|\vec{x} - \vec{x}(r, \theta, v)\|$. Thus by following this procedure, we can decide within ϵ error whether a given (x, y, z) lies on the surface of Klein bottle. In turn, P_c can be constructed by using reversible logic to evaluate this in time $O(\mathsf{poly}(\log(1/\epsilon)))$. Thus, complicated quantum dynamics on unusual manifolds can be simulated through the use of our approach to constraints, even in cases like the figure-8 immersion of the Klein bottle where no simple coordinate system is available that makes the computation of the Laplacian operator in (r, v, θ) coordinates trivial. This is because the gradient fails to be defined there, as the normal vector cannot be defined at the intersection in the figure-8. Instead, we can rely on the constraint operator to force the dynamics to lie on the surface of the bottle and use the standard Laplacian in Cartesian coordinates.

As a final point of discussion, let us consider applying these ideas to simulate a universal Hamiltonian with a quantum circuit. A universal Hamiltonian is a Hamiltonian such that the groundstate of the Hamiltonian encodes a quantum superposition of the history of a quantum computer via a clockstate of the form $\frac{1}{\sqrt{T}}\sum_t |t\rangle|\psi(t)\rangle$ where $|\psi(t)\rangle$ is the state of the quantum computer after t gates have been applied to it [113, 114, 115, 116]. In order to minimize the locality needed by these constructions, techniques such as "perturbative gadgets" are employed which allow restricted interactions such as 2-local ones to simulate the action of a Hamiltonian of greater k-locality. It is tempting therefore to ask whether our techniques could be used to accelerate the simulation of these constrained Hamiltonians.

As an example, the work of Ref. [116] shows that a translationally invariant 1D Hamiltonian of the following form for parameters T, Δ is universal

$$H = \sum_{\langle i,j \rangle} \Delta h_{ij}^{(3)} + T \sum_{i} h_i^{(2)}, \tag{2.77}$$

where each $h_{ij}^{(3)}$ is a two-body translationally invariant Hamiltonian and each h_i is a translationally invariant one-body Hamiltonian. At first glance, the latter term appears to be fast forwardable. This is significant, because the value T corresponds to the number of gates employed in the circuit. Thus we would be able to fast forward an arbitrary calculation if this were, by itself, true.

However, the value of Δ needed to provide a close approximation to the dynamics generically dominates the remaining term in Ref. [116]. In particular if we demand a simulation error on the order of ϵ, then it suffices to take $\Delta \in O(T^4/\epsilon)$ (for all other simulation parameters fixed). Thus the translationally invariant 2-body term dominates asymptotically and even if were possible to fast-forward the simulation of this Hamiltonian, the best case scenario would lead to a method that has scaling $O(T \log(1/\epsilon))$ from the one-body term. However, this construction is not self-evidently fast-forwardable and so a polynomial improvement is expected at best from transitioning to the interaction frame of the two-body operator.

2.6 Conclusions

We have developed novel simulation protocols that combine the standard simulation protocols of Trotterization, continuous qDRIFT, and qubitization in the interaction picture to simulate time-independent Hamiltonians. By exploiting the interaction picture, we can enter into the interaction frame of a fast-forwardable term with large or unbounded norm. Continuous qDRIFT is used to split the resulting time-ordered exponential into a product of time-independent exponentials with bounds proven for the number of time steps needed to achieve a desired error ϵ. In the case of Hamiltonians with underlying commutator structure, Trotterizing first can reduce the query complexity further. Qubitization is then used for im-

plementing the final time-independent exponentials, though other simulation techniques can be used.

The hybrid protocol using Trotterization before continuous qDRIFT in the interaction frame of a fast-forwardable term in the Hamiltonian has a query complexity of $O(t^2(c_I + \sum_{k\neq l}^{L} \|H_k\|_\infty^2)/\epsilon)$, where c_I depends on the sum of norms of commutators. For Hamiltonian simulation problems with commutator structure, this is a drastic improvement over the complexity $O(\|H_k\|_{\infty,1,1}^2/\epsilon)$ obtained from directly employing conventional qDRIFT methods to a linear combination query model. The qubitization and continuous qDRIFT hybrid I.P. protocol has a query complexity bounded by $\tilde{O}(\lambda_\alpha t + \|H_\alpha\|_\infty^2 t^2/\epsilon)$, where the quantities λ_α and H_α only involve the terms in the interaction Hamiltonian. If the term selected for the interaction frame is unbounded or of large operator norm, this again yields an improvement in the scaling with the ℓ^1-norm of H compared to qubitization. Our approach does not require a complicated clock construction either, which makes it more practical than truncated Dyson series methods [106].

Direct application of these techniques to the Schwinger Model yield a logarithmic scaling in the electric field cutoff Λ for the query complexity. For the Hamiltonian model of collective neutrino oscillations, the query complexity is independent of the typically large constant $\lambda = \sqrt{2}G_f n_e$ representing the electron density, with the same scaling with respect to other parameters compared to conventional Trotter-Suzuki methods. The scaling with these parameters outperforms those achieved by current simulation methods.

Further applications of these methods appear in simulating constrained dynamics. We show that the magnitude of the constraint term in the Hamiltonian needs to be prohibitively large to apply such a constraint using traditional simulation methods, such as qubitization. However, using our approaches we can simulate the dynamics using a number of gate operations that (for certain constraints) is independent of the magnitude of the constraint. This allows approximation methods similar to Trotter-Suzuki simulations to be employed while guaranteeing that the simulation does not break important symmetries present in the underlying dynamics (such as Gauss' law).

Another interesting fact to note is that even when Trotter formulas are used for the entire simulation, transforming to the interaction picture can have an advantage over performing the simulation in the laboratory frame. This is because Trotter formulas have costs that scale with fractional powers of the derivatives and lead to costs that are linear in the strength of the interaction term, rather than a super-linear function as would be expected from a simulation in the laboratory frame [17, 19]. Although hybrid methods that provide L^1-norm scaling are shown to be advantageous in this regard, this advantage can be useful and may lead to improved methods to reduce the cost of simulation purely within the Trotter-Suzuki formalism wherein the structure of commutators can be more easily exploited.

These hybrid techniques are primarily useful in contexts where there are not only terms of large operator norm in a Hamiltonian but when those terms are diagonalizable, one-sparse, or more generally fast-forwardable. However, situations often arise in quantum simulation where it might be desirable to enter the interaction frame of terms that are not fast-forwardable, such as the hopping term H_h of the Schwinger model in the continuum limit. As a naïve application of the present methods would involve doubling the number of times the non fast-forwardable term would be need to be simulated (see equation (2.17)), additional work is needed to develop interaction picture algorithms, hybrid or otherwise, that are more optimized with respect to parameters that define certain physical regimes of interest.

Algorithm	Number of oracle calls to W_k or PREPARE/SELECT and W_l
Trotter [19]	$O\left(\frac{\tilde{\alpha}^{1/p} t^{1+1/p}}{\epsilon^{1/p}}\right)$
qDRIFT [28]	$O\left(\frac{t^2}{\epsilon}\left[\sum_{k=1}^{L}\|H_k\|_\infty\right]^2\right)$
Qubitization [21, 22]	$O\left(\lambda t + \frac{\log(1/\epsilon)}{\log(\log(1/\epsilon))}\right)$
Trotter + qDRIFT + I.P. [Cor. 2.1.3]	$O\left(\frac{t^2}{\epsilon}\sum_{k\neq l}^{L}\left[\|H_k\|_\infty^2 + \left\|\left[H_k, \sum_{q>k, q\neq l}^{L} H_q\right]\right\|_\infty\right]\right)$
qDRIFT + Qubitization + I.P. [Th. 2.2.2]	$O\left(\lambda_\alpha t + \left(\frac{\|H-H_l\|_\infty^2 t^2}{\epsilon}\right)\frac{\log(\|H-H_l\|_\infty t/\epsilon)}{\log\log(\|H-H_l\|_\infty t/\epsilon)}\right)$

Table 2.1: Query complexities for standard qDRIFT, Trotter, qubitization, and the hybrid schemes from Corollary 2.1.3 and Theorem 2.2.2 where $H = \sum_{i=1}^{L} H_i$ with L the number of summands in the Hamiltonian H, t the simulation time, and ϵ the simulation error. In the Trotter formula, p is the order of the Trotter formula and $\tilde{\alpha}$ involves sums of commutators nested p times. The query complexity for qDRIFT and Trotter-based algorithms are given in terms of upper bounds for queries to each of the W_k oracles that implement time-evolution under a summand H_k. The query complexity for the qubitization methods are given in queries to the oracles W_l implementing time-evolution for the interaction picture transformation, SELECT, and PREPARE. For the latter methods, H is decomposed as a linear combination of unitaries $H = \sum_k \omega_k U_k$ with real $\omega_l > 0$. Then $\lambda = \sum_k \omega_k$ and $\lambda_\alpha = \lambda - \omega_l$. The hybrid I.P. schemes can become advantageous when $\lambda_\alpha \ll \lambda$ or $\|H - H_l\|_\infty \ll \|H\|_\infty$.

Chapter 3

QUANTUM ERROR CORRECTION WITH GAUGE SYMMETRIES

We develop fault-tolerant algorithms for error correction suitable for $\mathbb{Z}_2$ or truncated U(1) lattice gauge theories in 1+1, and 2+1 spacetime dimensions with a cutoff of 1 on the links. These algorithms combine the physicality constraint provided by Gauss' Law with bit and phase flip error correction codes [117] to detect and correct errors stemming from device noise that occur on a site or its adjacent links. Error correction for a 1D lattice with $2N$ links and $2N$ staggered fermionic sites with periodic boundary conditions is accomplished by tessellating these encodings across the whole lattice so that errors occurring on a particular site and its adjacent link can be detected by a Gauss' Law check on the next set of sites and links. The extension to 2D follows a similar idea using instead plaquettes and links coming out of them as the fundamental blocks of the partition.

3.1 Integrating the repetition code with Gauss' Law

We are now in the position to discuss how the expression of Gauss' Law in Eq. (1.30) can remove the need to use an explicit bit flip code in fault tolerant LGT simulations. To simplify the discussion we will henceforth consider the one dimensional case ($D = 1$) with staggered fermionic sites, where only one fermionic flavor with charge e_s is present on any given site s. The Gauss' Law operator at site s between links l and $l + 1$ simplifies then to

$$\hat{G}_s = \hat{E}_{l+1} - \hat{E}_l - e_s \hat{n}(s) \ , \tag{3.1}$$

and in the simpler pure gauge case with no matter to

$$\hat{G}_s = \hat{E}_{l+1} - \hat{E}_l \ . \tag{3.2}$$

We start by discussing the latter case and consider a general state of the two links in the electric basis

$$|\Psi_{l,l+1}\rangle = \sum_{\epsilon_l} \sum_{\epsilon_{l+1}} \Psi_{\epsilon_l,\epsilon_{l+1}} |\epsilon_l\rangle \otimes |\epsilon_{l+1}\rangle \ . \tag{3.3}$$

Following the discussion in Sec. 1.2.1, a physical state needs to be in the kernel of the Gauss' Law operator. Using Eq. (3.2) above we see that $\hat{G}_s$ acts on $|\Psi_{l,l+1}\rangle$ as

$$\hat{G}_s |\Psi_{l,l+1}\rangle = \sum_{\epsilon_l} \sum_{\epsilon_{l+1}} \Psi_{\epsilon_l,\epsilon_{l+1}} (\epsilon_{l+1} - \epsilon_l) |\epsilon_l\rangle \otimes |\epsilon_{l+1}\rangle \ , \tag{3.4}$$

and is zero only if the coefficient matrix is diagonal

$$\Psi_{\epsilon_l,\epsilon_{l+1}} (\epsilon_{l+1} - \epsilon_l) = 0 \ \Leftrightarrow \ \Psi_{\epsilon_l,\epsilon_{l+1}} = \Psi_{\epsilon_l} \delta_{\epsilon_l,\epsilon_{l+1}} \ . \tag{3.5}$$

This argument, which can be easily generalized to the case where the state of the links is mixed, shows that gauge invariant states are analogous to a bit-flip repetition code with only two copies: $|0\rangle_L = |0\rangle \otimes |0\rangle$ and $|1\rangle_L = |1\rangle \otimes |1\rangle$. The distance between these codewords is not sufficient to allow for error correction but is sufficient for error detection (see e.g. Ref. [118]). This explains in an intuitive way why oracles like those presented in Ref. [56] are capable of detecting bit-flip errors without requiring additional qubits for the encoding.

It is now easy to see how the use of Gauss' Law allows for a space reduction in the bit-flip encoding: the standard procedure described above will require three link registers to encode a logical link as

$$\begin{aligned} |\Phi_l\rangle_L &\coloneqq \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle \otimes |\epsilon_l\rangle \otimes |\epsilon_l\rangle \\ &= \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle_L . \end{aligned} \tag{3.6}$$

The fact that physical states satisfy Eq. (3.4) means we need only two registers per link and can use one of the two registers for the $l+1$ link across a site s when we measure stabilizers and perform error recovery. Since only three registers are involved in this procedure, we will consider a construction with two qubit registers for even links and only one register for the

odd links in order to minimize the memory cost. More explicitly, we will use the alternative encoding

$$|\Phi_l\rangle_{GLE} \coloneqq \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle \otimes |\epsilon_l\rangle = \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle_{GLE} \ , \tag{3.7}$$

for the even links in the lattice and

$$|\Phi_l\rangle_{GLO} \coloneqq \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle = \sum_{\epsilon_l} \Phi_{\epsilon_l} |\epsilon_l\rangle_{GLO} \ , \tag{3.8}$$

equivalent to a bare encoding, for the odd links. For physical states that satisfy the Gauss' law constraint in Eq. (3.5) we have then

$$\begin{aligned} |\Phi_{l,l+1}\rangle_{GL} &= \sum_{\epsilon_l} \sum_{\epsilon_{l+1}} \Psi_{\epsilon_l,\epsilon_{l+1}} |\epsilon_l\rangle_{GLE} \otimes |\epsilon_{l+1}\rangle_{GLO} \\ &= \sum_{\epsilon_l} \Psi_{\epsilon_l} |\epsilon_l\rangle_{GLE} \otimes |\epsilon_l\rangle_{GLO} \\ &= \sum_{\epsilon_l} \Psi_{\epsilon_l} |\epsilon_l\rangle \otimes |\epsilon_l\rangle \otimes |\epsilon_l\rangle \ , \end{aligned} \tag{3.9}$$

and we can now use the stabilizer measurements and recovery operation on the three qubits as in the bit-flip repetition code. The second line is obtained by ensuring that the logical state $|\Phi_{l,l+1}\rangle_{GL}$ is in the kernel of the logical Gauss' Law operator derived from Eq. (3.2)

$$\begin{aligned} \hat{G}_s^{(GL)} &= \hat{E}_{l+1}^{(GLO)} - \hat{E}_l^{(GLE)} \\ &= \hat{E}_{l+1} - \hat{E}_l \otimes \hat{E}_l \ . \end{aligned} \tag{3.10}$$

The expressions above apply directly to sites s between l even and $l+1$ odd, but it is straightforward to generalize them to the site $s+1$ where the left link is odd and the right link is even as follows

$$\begin{aligned} |\Phi_{l+1,l+2}\rangle_{GL} &= \sum_{\epsilon_{l+1}} \sum_{\epsilon_{l+2}} \Psi_{\epsilon_{l+1},\epsilon_{l+2}} |\epsilon_{l+1}\rangle_{GLO} \otimes |\epsilon_{l+2}\rangle_{GLE} \\ &= \sum_{\epsilon_{l+1}} \Psi_{\epsilon_{l+1}} |\epsilon_{l+1}\rangle_{GLO} \otimes |\epsilon_{l+1}\rangle_{GLE} \\ &= \sum_{\epsilon_{l+1}} \Psi_{\epsilon_{l+1}} |\epsilon_{l+1}\rangle \otimes |\epsilon_{l+1}\rangle \otimes |\epsilon_{l+1}\rangle \ . \end{aligned} \tag{3.11}$$

The gauge invariant logical state is again equivalent to the standard bit-flip encoding shown in Eq. (3.6). Since Gauss' Law only detects errors in the flux value, which in our representation corresponds to bit-flip errors, each register of qubits in the construction presented above needs to be encoded in a phase-flip code to ensure that all single errors are correctable. With this concatenation, the gauge-invariant logical states $|\Phi_{l,l+1}\rangle_{GL}$ and $|\Phi_{l+1,l+2}\rangle_{GL}$ become equivalent to a full 9-qubit encoding [117]. However, we encode two full links using the 9-qubit encoding as opposed to each link separately. This encoding therefore has better memory efficiency than error correcting codes that act on individual qubits, the best one requiring 5 qubits [119, 120]. Note that it is still possible to find codes with even higher memory efficiency when multiple logical qubit are encoded. For instance, 3 logical qubits can be encoded into 8 while correcting all single errors (see e.g. Ref. [121, 122]).

Adding fermions to the system modifies the relation between link states across a site so that they do not necessarily need to be the same. The full state around a site s is a generalization of Eq. (3.3) and should now be written as

$$|\Psi^s_{l,l+1}\rangle = \sum_{\epsilon_l}\sum_{\epsilon_{l+1}}\sum_{n=0,1} \Psi^n_{\epsilon_l,\epsilon_{l+1}} |\epsilon_l\rangle \otimes |\epsilon_{l+1}\rangle \otimes |n\rangle_s \ . \tag{3.12}$$

The gauge-invariance constraint becomes therefore

$$\begin{aligned} &\Psi^0_{\epsilon_l,\epsilon_{l+1}}(\epsilon_{l+1} - \epsilon_l) = 0 \ , \\ &\Psi^1_{\epsilon_l,\epsilon_{l+1}}(\epsilon_{l+1} - \epsilon_l - e_s) = 0 \ . \end{aligned} \tag{3.13}$$

A state belonging to the physically meaningful portion of the Hilbert space is then

$$|\Psi^s_{l,l+1}\rangle = \sum_{\epsilon_l}\sum_{n=0,1} \Psi^n_{\epsilon_l} |\epsilon_l\rangle \otimes |\epsilon_l + n e_s\rangle \otimes |n\rangle_s \ , \tag{3.14}$$

where we recall that $e_s = 1$ for a fermionic site and $e_s = -1$ for an anti-fermionic site. In order to obtain a useful encoded state for error correction, we can then apply a lowering operator $\hat{U}^\dagger_{l+1}$ (for $e_s = 1$) or a raising operator $\hat{U}_{l+1}$ (for $e_s = -1$) to the link $l+1$ controlled on the state of the site qubit. Calling this operation $\hat{W}_s$, for a site between an even and an

odd link we have that

$$\begin{aligned}\hat{W}_s|\Psi^s_{l,l+1}\rangle_{GL} &= \sum_{\epsilon_l}\sum_{n=0,1}\Psi^n_{\epsilon_l}|\epsilon_l\rangle_{GLE}\otimes|\epsilon_l\rangle_{GLO}\otimes|n\rangle_s \\ &= \sum_{\epsilon_l}\sum_{n=0,1}\Psi^n_{\epsilon_l}|\epsilon_l\rangle\otimes|\epsilon_l\rangle\otimes|\epsilon_l\rangle\otimes|n\rangle_s \end{aligned} \tag{3.15}$$

and the error correction procedure can be carried out on this new state. After one site has been processed, we apply the inverse $\hat{W}_s^\dagger$ and move to the next site. In this approach, we effectively move from a gauge-invariant encoding that satisfies Eq. (3.13) to a logical encoding equivalent to the bit-flip code by using $\hat{W}_s$ and its inverse. This operation needs to be performed fault-tolerantly in order to ensure a controlled propagation of errors. In the next section we will restrict the discussion to the simpler case where the link registers are formed by a single qubit (i.e. the flux cutoff is set to one). In that case, we show that the construction can be done in a fault-tolerant way.

It is preferable to avoid directly implementing the $\hat{W}_s$ operation in practice as bit-flip errors can propagate from the site used as the control for the $l+1$ link. This can be done for instance by adopting a full bit-flip encoding for the site alone and using the measurement of its stabilizers to correct for the induced error in the link. A more efficient strategy however is to compute the parity between the between the $l+1$ link and the adjacent site and store the result in a physical ancilla qubit, as shown schematically in Equation 3.1. The ancilla qubit can then be used as part of a logical encoding for the even-numbered links. We can catch errors of weight two or higher that occur from errors propagating past the logical CNOTs with the use of flag qubits (see Ref. [66]) and an example of this is shown in Figure 3.3. This construction can be readily extended to situations with large cutoffs if we employ a unary encoding for the flux state using m qubits per link, in which case we will require m ancilla qubits. Generalizing the approach to larger cutoff values with various encodings is an important issue which will need to be addressed in future work (see also Sec. 3.4). As we show in more detail in the next section, using ancilla qubits to temporarily store the parity of the two links allows for a fault-tolerant error correction of a gauge theory with dynamical

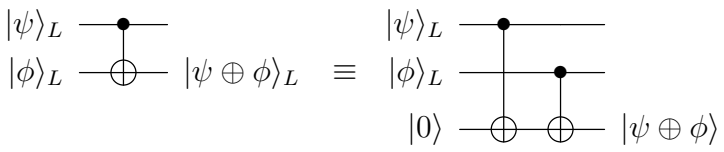


Figure 3.1: Equivalence for logical CNOT between logical qubits and for two logical-to-physical CNOTs.

fermions using the unencoded sites directly, providing a significant saving in qubits.

In summary, the number of link registers required for the Gauss' Law aided bit-flip error correction is $3N$ for a system with a total of $2N$ links. The total number of qubits required in a pure gauge simulation is therefore $9N+$ O(1) when restricting the unit electric cutoff case as done above and accounting for the additional 3 physical qubits required for the phase-flip encoding. With the addition of dynamical fermions on $2N$ sites, one can exploit the relationship given by Gauss' law between the state of a site and its adjacent links to do error correction with a total of $15N+$ O(1) qubits, $9N$ for links and $6N$ for sites. These are impossible results if one instead attempts to perform error correction on individual qubits. The total space required would instead be $10N$ for the non-dynamical case and $20N$ for the dynamical one using the perfect 5-qubit encoding from Refs. [119, 120]. As mentioned previously, better ratios are possible if one encodes multiple logical qubits at a time and we leave the extension to the more efficient encodings for future work [121, 122].

3.2 Applications to 1D Lattice Gauge Theories

We specialize to a $\mathbb{Z}_2$ or truncated U(1) gauge theory in 1+1 dimensions with a cutoff of $\Lambda = 1$ for the flux in each link in this section. We work with a system with $2N$ sites and $2N$ links with PBCs. Site k, for an integer $k \geq 1$, is denoted by S_k^1 and the links coming into and out of the site are written as L_k^1 and L_{k+1}^1 respectively.

Unless otherwise specified, every circuit in this section is a concatenated one, with every

qubit encoded within the phase flip error-correcting code as in Figure 1.8. Logical qubits arising in this manner will be denoted with the subscript L.

This section presents applications of the previous analysis to a few gauge theories in order of increasing generality. Beginning with the pure gauge theory case with no matter present on the sites, Gauss' Law gives the equality between the incoming and outgoing flux on a site. This allows us to use $|L_k^1\rangle$ and $|L_{k+1}^1\rangle$ as part of a single logical qubit for a concatenated bit-flip error correction procedure, where an additional qubit is introduced only if the links are even-numbered.

This situation is easily generalized to a theory with static charges or non-dynamical fermions on the sites by applying a link lowering operator on an odd-numbered link conditioned on the classical state of the adjacent site. We then perform the error correction procedure on the appropriate logical link qubit before restoring the odd link's flux value with a raising operator.

The scenario with dynamical fermions is dealt with by computing the parity between an odd-numbered link and its adjacent site and storing the result in a physical ancilla qubit. This qubit is then used as part of a logical encoding for the even-numbered links. Bit-flip checks can be performed on two sets of adjacent links and sites and the information from them can be used to correct errors on all the relevant qubits, provided there is a single bit-flip error per two such checks.

Note that we cannot perform this analysis entirely within the stabilizer formalism since our controlled operations are not logical ones at the level of the bit-flip encoding. While certain aspects of our treatment therefore rely on standard constructions within the stabilizer formalism, others involving Gauss' Law require us to move between physical and logical states.

3.2.1 Error Correction for Pure Gauge Theory

We adopt the notation $|L_k^n\rangle_L$ and $|S_k^n\rangle_L$ to denote link and site logical qubits respectively in the phase flip code, and O_k^n for operators acting on the k-th logical link or site qubit.

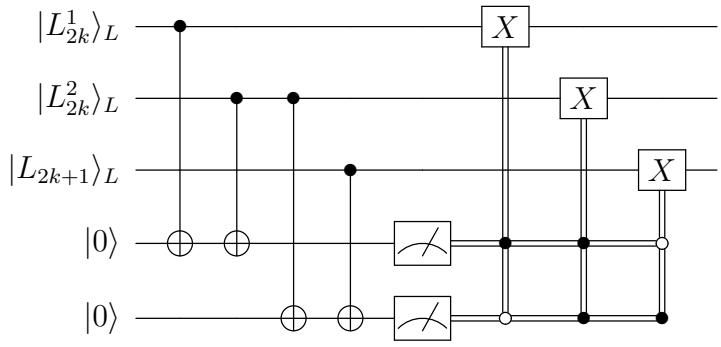


Figure 3.2: Bit-flip syndrome measurement and correction operations for links. The subscript L denotes the phase-flip encoding of these qubits shown in Figure 1.8. Each CNOT operation here consists of three individual CNOTs from the qubits in the underlying phase-flip encoding. Here X is the operation $X \otimes X \otimes X$ which acts as a logical X.

The case where $n = 1$ in the superscript denotes the original logical qubit and logical qubit operator, while values of n greater than 1 denote the auxiliary link or site qubits introduced for the error correction procedures. If a given logical site or link is the only one of its kind, we will drop the superscript.

In each of the subsequent circuits, we introduce no additional qubits for the odd numbered links and an auxiliary qubit $|L^2_{2k}\rangle_L$ for each even numbered link $|L^1_{2k}\rangle_L$ that is arranged to have the same state as $|L^1_{2k}\rangle_L$. The constraint that the fluxes satisfy Gauss' Law at site $|S^1_{2k}\rangle_L$ with associated link qubits $\{|L^1_{2k}\rangle_L, |L^2_{2k}\rangle_L, |L_{2k+1}\rangle_L\}$ will then ensure that $|L^1_{2k}\rangle_L = |L^2_{2k}\rangle_L = |L^1_{2k+1}\rangle_L$, or more precisely that the wave-function factorizes as in Eq. (3.9) above.

As noted in the previous section, the grouping $\{|L^1_{2k}\rangle_L, |L^2_{2k}\rangle_L, |L_{2k+1}\rangle_L\}$ acts as a 3-qubit logical encoding for the link logical qubit $|L^1_{2k}\rangle_L$ that can be used in the bit flip error correction code outlined in Figure 3.2. The circuit is however sensitive to single qubit phase flip errors that propagate to weight two or three phase flip errors. This is due to the use of logical to physical CNOT gates, which enable the propagation of physical Z errors from the target of these CNOTs to logical Z errors on the links. We therefore employ the fault-

tolerant logical-to-physical CNOT depicted in Figure 3.3. This CNOT is implicitly used in all the subsequent circuits where logical-to-physical CNOT operations occur.

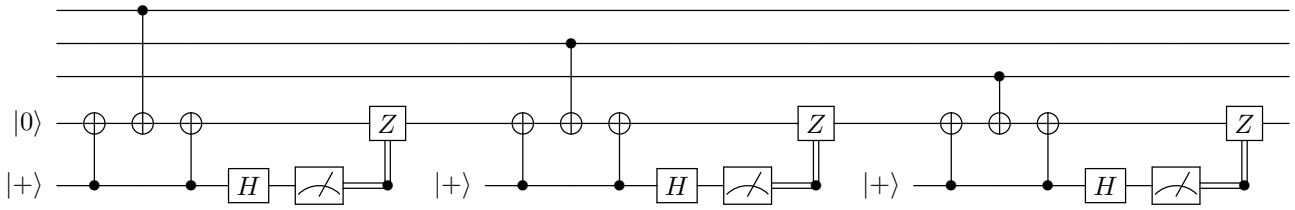


Figure 3.3: Fault tolerant logical-to-physical CNOT using flag qubits.

There is an alternative way to accomplish this error correction procedure without introducing an additional logical qubit for the even numbered links. This method simply requires doing overlapping Gauss' law checks on the qubits $\{|L_{2k}\rangle_L, |L_{2k+1}\rangle_L, |L_{2k+2}\rangle_L\}$ and the qubits in Figure 3.2 can then be replaced with these respectively. The number of qubits required to fully error correct a 1+1 lattice system with $2N$ links would then be $6N$ instead of $9N$. However, we can then only tolerate at most one error occurring over a larger portion of the lattice, namely a group of three consecutive links, compared to the previous scenario where we could tolerate at most one error over the more localized grouping $\{|L^1_{2k}\rangle_L, |L^2_{2k}\rangle_L, |L_{2k+1}\rangle_L\}$. Thus there is a trade-off between the number of qubits needed for our compressed encoding and the spatial region over which the Gauss' law checks need to be performed to localize a bit-flip error. This pattern will recur in the subsequent cases we consider and becomes less favorable in larger dimensions.

3.2.2 Error Correction for Non-Dynamical Fermions

It is straightforward to generalize the results of the preceding subsection to the case where we have non-dynamical fermions on the sites. Since we know in this case whether or not a certain physical site S^1_{2k} (this is not a logical site qubit but a classical binary variable) has a fermion, it is not necessary to introduce an ancilla qubit and use the equivalence in Equation 3.1.

The controlled operation $\hat{W}_s$ introduced above (see Eq. (3.15)) is instead replaced by its classically controlled counterpart. As such, we apply a classically conditioned link-lowering operator to $|L_{2k+1}\rangle_L$ to "lower" its flux value and have it match that of $|L^1_{2k}\rangle_L$. We then apply the preceding bit-flip error correction code and restore the flux value of $|L^1_{2k+1}\rangle_L$ back to its original value with a classically conditioned link-raising operation.

Given our assumptions on the flux cutoff, we have the circuit in Figure 3.4 for the non-dynamical fermionic case. This construction also holds for an arbitrary site S_k and not just even numbered ones. Note that the circuit can be expressed in a simpler form by exploiting commutation relations to move the initial classically controlled X to the end of the circuit. This effectively replaces this operation with a classically controlled flip of the last measurement outcome.

As in the pure gauge theory case, we can dispense with the extra logical qubit $|L^2_{2k}\rangle_L$ and extend the Gauss' law check to the group of qubits $\{|L^1_{2k}\rangle_L, |L^2_{2k}\rangle_L, |L_{2k+1}\rangle_L\}$, replacing the link qubits in Figure 3.4 accordingly. The number of qubits needed for this approach is again $6N$ like the pure gauge case instead of $9N$ with the above encoding, and the trade-offs are identical as well.

3.2.3 Error Correction for Dynamical Fermions

A naïve approach to extend the preceding constructions to the case with dynamical staggered fermions on the sites is to apply a CNOT gate to $|L_{2k+1}\rangle_L$ conditioned on the state of $|S_{2k}\rangle_L$. We could then introduce additional qubits $|S^2_{2k}\rangle_L$ and $|S^3_{2k}\rangle_L$ to create a bit-flip encoding of $|S_{2k}\rangle_L$ and thereby protect it from bit-flip errors. Errors propagating to $|L_{2k+1}\rangle_L$, $|L^1_{2k}\rangle_L$, and $|L^2_{2k}\rangle_L$ can be corrected with a separate bit-flip code involving these qubits, with additional recovery operations included to account for the propagation of X errors past the control of the CNOT.

This construction only involves Clifford operations and can easily be made fault tolerant. It however requires $18N + 6N + 3N = 27N$ qubits for an $2N$ site and $2N$ link system excluding ancillas, which is suboptimal compared to the $20N$ qubits obtained from using the

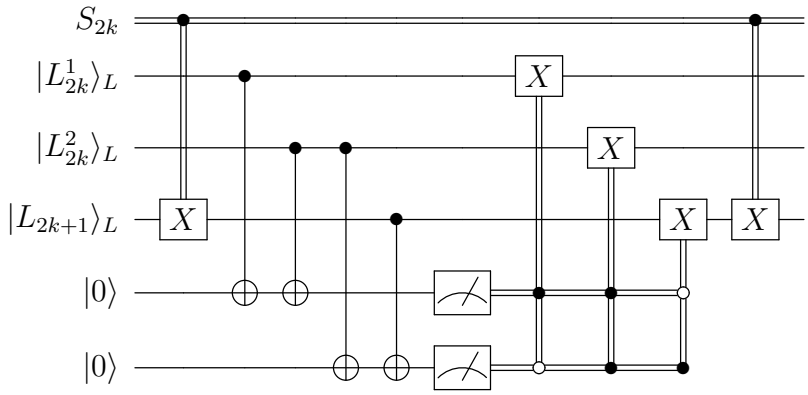


Figure 3.4: Syndrome measurement and correction operations for LGT system with non-dynamical fermions. The circuit holds for arbitrary sites S_k and not just even numbered ones. Note that the circuit can be simplified by using commutation relations to move the initial classically controlled X to the end of the circuit, thereby replacing the classical controls with a classically controlled flip of the last measurement outcome.

$[5, 1, 3]$ encoding for each site and link.

We instead develop a procedure that further exploits Gauss' law to reduce the cost to $15N$ qubits. This is achieved by noticing that, as discussed in Sec. 3.1, given the state on a site S_{2k+1} and its adjacent links L_{2k+1} and L_{2k+2}, a bit-flip error on any of these qubits will cause a violation of Gauss' law. The ambiguity in where the error occurred can be resolved by overlapping Gauss' law checks on the preceding and subsequent site and their adjacent links. In fact, we will show only two such checks are needed to resolve the ambiguity.

Figure 3.5 depicts to overlapping Gauss' law checks $\mathcal{G}_{2k}$ and $\mathcal{G}_{2k+1}$ on sites S_{2k} and S_{2k+1} respectively. Consider $\mathcal{G}_{2k+1}$ in particular. We can compute the parity between $|S_{2k+1}\rangle_L$ and $|L_{2k+1}\rangle_L$ as per Equation 3.1 and store this result into a physical ancilla qubit $|0\rangle$, which serves as a proxy for $|L_{2k+1}\rangle_L$ but with its flux value reset. We can then treat $|0\rangle$, $|L^1_{2k+2}\rangle_L$, $|L^2_{2k+2}\rangle_L$ as a logical qubit for a bit-flip error correction code and uncompute the value in $|0\rangle$ afterwards. These steps constitute $\mathcal{G}_{2k+1}$ for the odd sites and their neighboring links and

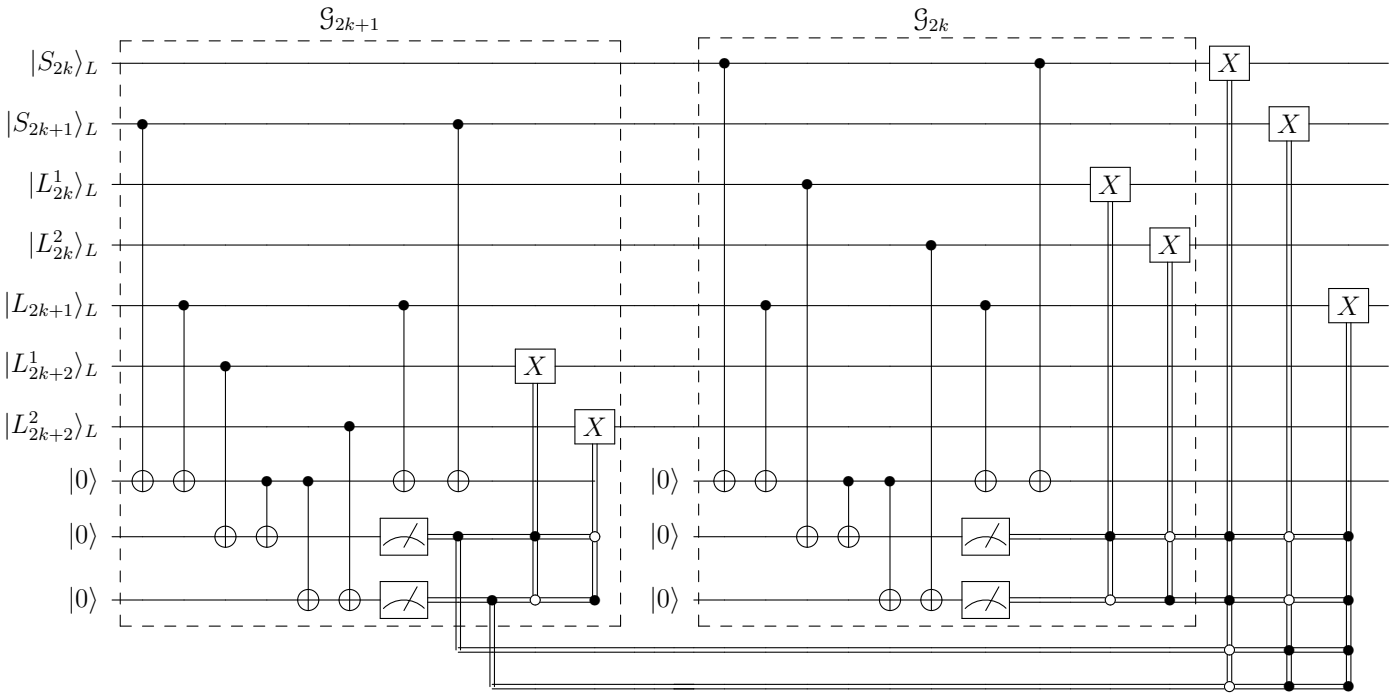


Figure 3.5: Circuit for using information from Gauss' Law checks $\mathcal{G}_{2k}$ and $\mathcal{G}_{2k+1}$ on S_{2k}, S_{2k+1}, and their adjacent links to correct bit-flip errors on them. Bit-flip errors on even-numbered links can be corrected using the information obtained from one such check. Bit-flip errors on sites and odd-numbered links require information from two such checks to be corrected. See Table 3.1 for the corresponding syndrome outcomes.

are repeated analogously across the lattice.

To see how the information obtained from the stabilizer measurements employing the last two ancilla qubits in Figure 3.5 helps us correct errors on the data qubits, consider $\mathcal{G}_{2k+1}$ first. The syndrome measurement outcomes 10 and 01 uniquely identify bit-flip errors occurring on the even links $|L^1_{2k+2}\rangle_L$ and $|L^2_{2k+2}\rangle_L$ respectively. The remaining non-trivial outcome 11 indicates a bit-flip error on either $|L_{2k+1}\rangle_L$ or $|S_{2k+1}\rangle_L$ with no way to resolve the ambiguity.

If the error occurred on $|S_{2k+1}\rangle_L$ however, we would obtain the syndrome 00 from the check $\mathcal{G}_{2k}$ as Gauss' law is satisfied between S_{2k}, L_{2k}, and L_{2k+1}. If the error occurred on $|L_{2k+1}\rangle_L$, we would then obtain the syndrome 11 from $\mathcal{G}_{2k}$. As these two cases yield distinct

syndromes, we can resolve the aforementioned ambiguity as shown in Table 3.1. Note that this analysis will only hold if a bit-flip error occurs once during the checks $\mathcal{G}_{2k}$ and $\mathcal{G}_{2k+1}$.

To assess the fault-tolerance of this construction, we first consider the possibility that single-qubit bit flip errors occurring at the physical level can propagate to undetectable errors at level of the phase-flip code. Consider WLOG an error of the form $X \otimes I \otimes I$ occurring at the level of the underlying phase-flip code. On the logical codespace defined by the basis $|0\rangle_L = (|++\rangle + |--\rangle)/\sqrt{2}$ and $|1\rangle_L = (|++\rangle - |--\rangle)/\sqrt{2}$, this error maps $|0\rangle_L \mapsto |1\rangle_L$. In other words, a physical bit-flip error propagates to a logical bit-flip error which can be detected by our bit-flip correction code. It is easily seen that two bit flip errors leave our code space invariant. The circuit's sensitivity to single qubit phase flip errors propagating to weight two or three phase flip errors from the logical-to-physical CNOT gates is again resolved with the fault-tolerant CNOT construction shown in Figure 3.3.

Logical bit-flip errors occurring on the logical qubits only propagate to the ancilla qubits and not to the data qubits. The possibility of more than one such error occurring on the data qubits is suppressed by higher powers of p, where p is the probability of an error occurring.

We can again consider the situation where we dispense with the extra qubits for the even-numbered links and correct bit-flip errors solely based on information obtained from Gauss' law. This modified situation is depicted in Figure 3.6 and shows how three overlapping Gauss' law checks can be used to detect an error on $|S_{2k+1}\rangle$ or its adjacent links. Note that this

$\mathcal{G}_{2k+1}$	$\mathcal{G}_{2k}$	Error Location
00	00	None
00	11	$\|S_{2k}\rangle_L$
11	00	$\|S_{2k+1}\rangle_L$
11	11	$\|L_{2k+1}\rangle_L$

Table 3.1: Table showing how to resolve the ambiguity in the location of a bit-flip error associated to the syndrome 11 for $\mathcal{G}_{2k+1}$ using the syndrome results of $\mathcal{G}_{2k}$. See Figure 3.5.

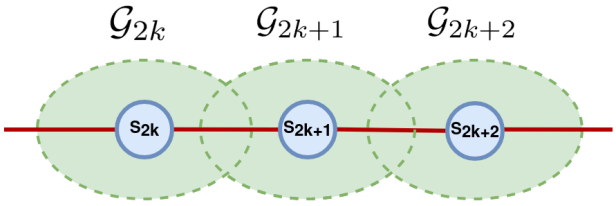


Figure 3.6: Schematic illustration of how one can correct any single-qubit error on S_{2k+1} and its adjacent logical link qubits by 3 overlapping Gauss' law checks. This requires that only one single-qubit error occurs over the links and qubits involved in the 3 Gauss' law checks.

requires only one single-qubit error to occur on the links and qubits involved in the checks, and so we can lessen the memory overhead required for error-correction in this setting at the expense of a worse error tolerance over a larger region of the lattice. This modified approach then requires $3(2N) + 3(2N) = 12N$ qubits for a 1+1 dimensional lattice system with $2N$ staggered fermions and $2N$ links. The corresponding circuit is similar to Figure 3.5, with the exception that the extra even-numbered link qubits and intermediate classical recovery operations in each $\mathcal{G}_j$ are removed and the logical qubits $|S_{2k+2}\rangle_L, |S_{2k+3}\rangle_L$ are present. We simply perform the parity checks $\mathcal{G}_{2k}, \mathcal{G}_{2k+1}, \mathcal{G}_{2k+2}$, store the results into 3 ancilla qubits, and use the outcomes to correct the error that happened on $|S_{2k+1}\rangle_L, |L_{2k+1}\rangle_L, |L_{2k+2}\rangle_L$. Table 3.2 depicts the possible outcomes from these checks and shows that we can localize the error.

3.3 Extension to two dimensions

The scheme described in the previous sections can be also extended to higher spatial dimensions. In order to give a concrete example we describe here how one could implement a compressed error correcting code for a simple 2+1 dimensional lattice gauge theory. We consider in detail the case of $\mathbb{Z}_2$ but the construction can be also applied to U(1). We start from a different form of Gauss' law which allows for a more direct connection to the stabilizer formalism. The Gauss' operator at each site $\mathbf{s} = (x_s, y_s)$ and it's neighboring links is defined

$\mathcal{G}_{2k}$	$\mathcal{G}_{2k+1}$	$\mathcal{G}_{2k+2}$	Error Location
0	0	0	None
0	1	1	$\|L_{2k+2}\rangle_L$
0	1	0	$\|S_{2k+1}\rangle_L$
1	1	0	$\|L_{2k+1}\rangle_L$

Table 3.2: Table showing how to localize the error occurring on $|S_{2k+1}\rangle_L$ or its adjacent logical link qubits $|L_{2k+1}\rangle_L, |L_{2k+2}\rangle_L$ by overlapping Gauss' law checks. Each error is associated with a unique syndrome, enabling error correction. See Figure 3.6 as well.

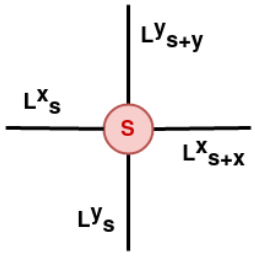


Figure 3.7: Convention for the indexing of the link variables around a fermionic site $\mathbf{s} = (x_s, y_s)$.

as

$$\hat{G}(\mathbf{s}) = Z^x_{\mathbf{s}} Z^y_{\mathbf{s}} Z^x_{\mathbf{s}+x} Z^y_{\mathbf{s}+y} (-1)^{\hat{q}(\mathbf{s})} \,, \tag{3.16}$$

and physical states satisfy the condition

$$\hat{G}(\mathbf{s})|\Psi\rangle_{phys} = |\Psi\rangle_{phys} \,. \tag{3.17}$$

In the expression Eq. (3.16) we use the indexing convention for the links as displayed in Fig. 3.7 and the charge operator using staggered fermions is given by

$$\hat{q}(\mathbf{s}) = \psi^\dagger(\mathbf{s})\psi(\mathbf{s}) - \frac{1}{2}\left(1 - (-1)^{x_s+y_s}\right), \tag{3.18}$$

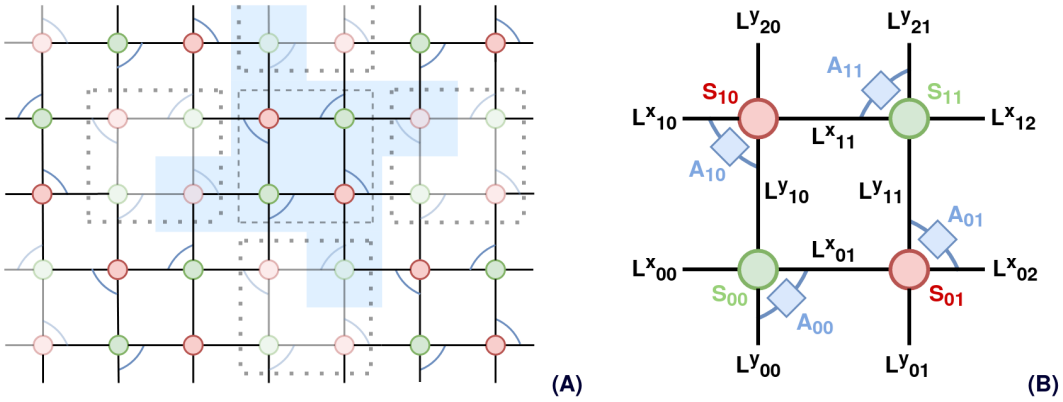


Figure 3.8: Panel (A) shows the adopted bi-partition of the 2D lattice into plaquettes used for the two rounds of the error correction procedure: the first round acts on bold links and sites (e.g. the region delimited by the dashed line) the second round acts on the light colored links and sites and the 8 external links attached to them (e.g. the regions delimited by dotted lines). The blue shaded are shown in panel (A) is the patch of the 2D lattice where one bit flip error can occur and be corrected. Panel (B) shows in detail the structure of the relevant region for one round of the error correction procedure.

with $\hat{\psi}^\dagger(\mathbf{s})$ and $\hat{\psi}(\mathbf{s})$ the fermionic creation and annihilation operators at site $\mathbf{s}$. With this definition, at even site the states $\{|0\rangle, |1\rangle\}$ indicate the absence/presence of a fermion of charge 1 while for odd sites the role is reversed and $\{|0\rangle, |1\rangle\}$ represent the presence/absence of an anti-fermion of opposite charge. In the following we will denote fermionic sites with a red color (as in Fig. 3.7) and anti-fermion sites with a green color. The convention for the indexing of links is unaffected. The presence of static charges at different sites is easily included by adding their contributions in the charge at the site. In order to keep the current exposition simple we neglected this possibility.

Using the Pauli Z operator at the site we can express the Gauss' law operator more explicitly as

$$\hat{G}(\mathbf{s}) = (-1)^{x_s+y_s} Z^x_{\mathbf{s}} Z^y_{\mathbf{s}} Z^x_{\mathbf{s}+x} Z^y_{\mathbf{s}+y} Z_{\mathbf{s}} \ , \tag{3.19}$$

and we see then that physical states are eigenvectors of

$$Z^x_{\mathbf{s}} Z^y_{\mathbf{s}} Z^x_{\mathbf{s}+x} Z^y_{\mathbf{s}+y} Z_{\mathbf{s}} \tag{3.20}$$

with positive eigenvalue on even sites and negative eigenvalue on odd sites. As alluded above, this form is reminiscent of the stabilizer formalism.

The error correcting procedure proceeds as follow: (1) we first partition the full lattice into square plaquettes and their neighboring links and proceed in two rounds by first considering the plaquettes indicated with bold colors in panel (A) of Fig. 3.8 (see e.g. the one indicated by the dashed line) and in the second round we consider the second half of the plaquettes denoted with light colors (for the highlighted plaquette, these correspond to those indicated by dotted lines); (2) for each individual plaquette we perform a sequence of parity checks. Before describing these, note that we require an additional qubit for every pair of links denoted as a blue curved line in the diagrams of Fig. 3.8. These qubits are used to store the parity of the two links they touch, for instance

$$Z^x_{(1,0)} Z^y_{(1,0)} |\Phi\rangle = Z_{A_{10}} |\Phi\rangle \ , \tag{3.21}$$

$P^a_{(0,0)}$	$P^b_{(0,0)}$	$P^c_{(0,0)}$	$P^a_{(0,1)}$	$P^b_{(0,1)}$	$P^c_{(0,1)}$	$P^a_{(1,1)}$	$P^b_{(1,1)}$	$P^c_{(1,1)}$	$P^a_{(1,0)}$	$P^b_{(1,0)}$	$P^c_{(1,0)}$	error location
+	+	+	+	-	-	+	+	+	+	-	-	none
+	+	+	+	-	-	+	+	+	-	-	+	A_{10}
+	+	+	+	-	-	-	+	-	+	-	-	A_{11}
+	+	+	-	-	+	+	+	+	+	-	-	A_{01}
-	+	-	+	-	-	+	+	+	+	-	-	A_{00}
+	+	+	+	-	-	+	+	+	-	+	-	L^x_{10}
+	+	+	+	-	-	-	-	+	+	-	-	L^y_{21}
+	+	+	-	+	-	+	+	+	+	-	-	L^x_{02}
-	-	+	+	-	-	+	+	+	+	-	-	L^y_{00}
+	-	-	+	-	-	+	+	+	-	+	-	L^y_{10}
+	+	+	+	-	-	-	-	+	+	+	+	L^x_{11}
+	+	+	-	+	-	-	-	+	+	-	-	L^y_{11}
-	-	+	+	+	+	+	+	+	+	-	-	L^x_{01}
+	+	+	+	-	-	+	+	+	+	+	+	s_{10}, L^y_{20}
+	+	+	+	-	-	+	-	-	+	-	-	s_{11}, L^x_{12}
+	+	+	+	+	+	+	+	+	+	-	-	s_{01}, L^y_{01}
+	-	-	+	-	-	+	+	+	+	-	-	s_{00}, L^x_{00}

Table 3.3: Syndrome measurements outcomes, and corresponding error location, for the 12 stabilizers used on one complete plaquette. The labeling conventions follow Fig. 3.8 (B). We have excluded outcomes corresponding to more than one error.

where with $|\Phi\rangle$ we denote the full state. This can be achieved by simply applying two CNOT gates with control on the links and target on the ancillary qubit initially set in $|0\rangle$. Then, whenever we modify the state of a link, we correspondingly update the state of the ancilla by applying another CNOT. These are CNOT operations between phase-flip encoded qubits and can be done transversally. At this point three separate parity checks are performed on each site participating in the plaquette and its four connected links. Starting from the site $(0,0)$ on the bottom left, we have the stabilizers

$$\begin{aligned} P^a_{(0,0)} &= Z^y_{(0,0)} Z^x_{(0,1)} Z_{A_{00}} \\ P^b_{(0,0)} &= Z^y_{(0,0)} Z^x_{(0,1)} Z^y_{(1,0)} Z^x_{(0,0)} Z_{(0,0)} \\ P^c_{(0,0)} &= Z_{A_{00}} Z^y_{(1,0)} Z^x_{(0,0)} Z_{(0,0)} \ , \end{aligned} \tag{3.22}$$

where $Z_{(0,0)}$ is the operator acting on the qubit at the site. The parity check table for one site is reported in Tab. 3.4, together with the possible error location. We can see that only 4 outcomes correspond to either one error or no errors. In the following discussion on the complete plaquette we restrict our attention to these situations only. Note however that errors on the ancillas $A_{\mathbf{s}}$ can be uniquely determined by only stabilizer measurements on sites and we could therefore detect and fix an arbitrary number of them.

We will use the same convention for all the other sites and take $P^a_{\mathbf{s}}$ to be the weight 3 stabilizer between the ancilla $A_{\mathbf{s}}$ and the two link connected to it, $P^b_{\mathbf{s}}$ the weight 5 stabilizer from the Gauss' law operator Eq. (3.20) and $P^c_{\mathbf{s}}$ the weight 4 stabilizer obtained applying Gauss' law to the ancilla $A_{\mathbf{s}}$, the two links not connected to it and the site. Accounting for the difference between the correct eigenspace of the Gauss' law stabilizer for even (fermionic) and odd (anti-fermionic) sites, the parity check table for a complete plaquette is reported in Tab. 3.3. Note that the last four syndrome outcomes do not allow to uniquely localize the error as it could have happened on either the site or the external link opposite to the location of the ancilla $A_{\mathbf{s}}$. Similarly to the construction described above for the one dimensional case, this ambiguity can be completely resolve by measuring one stabilizer on the neighboring plaquettes. In particular, under the assumption that only one error can occur in the shaded

$P^a_{(0,0)}$	$P^b_{(0,0)}$	$P^c_{(0,0)}$	error location
+	+	+	none
+	+	-	2+
+	-	+	2+
+	-	-	s_{00},L^x_{00},L^y_{10}
-	+	+	2+
-	+	-	A_{00}
-	-	+	L^y_{00}, L^x_{01}
-	-	-	2+

Table 3.4: Syndrome measurement outcomes and corresponding error location(s) for the parity check around a single site. Two or more errors are denoted generically as 2+.

area in panel (A) of Fig. 3.8 during the two rounds of check, it is enough to measure the P^a stabilizers on the four highlighted sites. For completeness, we report the full syndrome measurement table required to distinguish these cases in Tab 3.5.

The extended 2D scheme presented here uses only standard stabilizer measurements over phase-flip encoded qubits and can thus be implemented in a fault tolerant way using similar techniques to those discussed in the 1D case. The main non-standard component of the scheme is the presence of the ancillae $A_{\mathbf{s}}$ which do not store the state of another qubit but the parity of a pair. The initial logical state can be initialized in a straightforward way by starting the system in a physical product state and making sure that any change performed to the state of the links connected to a given ancilla are also applied to the latter. For example, when implementing a hopping term which changes the flux on one of the links, the same change has to be enacted on the ancilla touching that link before continuing. In the $\mathbb{Z}_2$ case these extended operations can be achieved with just Clifford gates as they correspond simply to doubling the number of CNOT required.

The total qubit count for a square system with $4N_xN_y$ sites and $8N_xN_y$ links using a

$P^a_{(0,0)}$	$P^b_{(0,0)}$	$P^c_{(0,0)}$	$P^a_{(0,1)}$	$P^b_{(0,1)}$	$P^c_{(0,1)}$	$P^a_{(1,1)}$	$P^b_{(1,1)}$	$P^c_{(1,1)}$	$P^a_{(1,0)}$	$P^b_{(1,0)}$	$P^c_{(1,0)}$	$P^a_{(0,-1)}$	$P^a_{(-1,1)}$	$P^a_{(1,2)}$	$P^a_{(2,0)}$	location
+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	s_{10}
+	+	+	+	-	-	+	-	-	+	-	-	+	+	+	+	s_{11}
+	+	+	+	+	+	+	+	+	+	-	-	+	+	+	+	s_{01}
+	-	-	+	-	-	+	+	+	+	-	-	+	+	+	+	s_{00}
+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	-	L^y_{20}
+	+	+	+	-	-	+	-	-	+	-	-	+	+	-	+	L^x_{12}
+	+	+	+	+	+	+	+	+	+	-	-	+	-	+	+	L^y_{01}
+	-	-	+	-	-	+	+	+	+	-	-	-	+	+	+	L^x_{00}

Table 3.5: Full parity check outcomes required to uniquely localize the errors on sites or external links opposite to the location of the $A_{\mathbf{s}}$ ancillas obtained using four additional P^a stabilizer measurement on neighboring sites.

standard $[5, 1, 3]$ code, and neglecting ancillae required for the stabilizer measurements, would be $60N_xN_y$ while the scheme proposed in this work requires only $48N_xN_y$ qubits. Similarly to the 1D case discussed in previous sections, the proposed mapping has the additional benefit of allowing simpler implementations of transversal operations. The savings afforded by this scheme can be improved in a similar way as what we did for the one dimensional case: if one can tolerate a single fault in a larger region than the one depicted in panel (A) of Fig. 3.8 then some of the ancilla registers $A_{\mathbf{s}}$ could be removed. For instance, if one extends the region where we can tolerate one error to enclose the four plaquettes neighboring a specific plaquette, a fault-tolerant scheme can be devised using only half of the $A_{\mathbf{s}}$ registers.

The present discussion of the 2D case can serve as a guideline to design similar error correction codes for higher dimensional geometries by adding additional ancilla registers to store the parity of subsets of links. Finally, extensions of the present scheme to $\mathbb{Z}_N$ or to U(1) with larger flux cutoffs could be constructed in principle using the same unary encoding map proposed for the one dimensional case.

3.4 Conclusion

Error correction involves extending the original Hilbert space of the system to a larger one and endowing the larger Hilbert space with local symmetries that define the codespace. Stabilizers of the codespace are in effect "Gauss' Law" operators of that symmetry. We have outlined a simple fault tolerant algorithm which exploits local symmetries to reduce the space requirements for performing error correction on $\mathbb{Z}_2$ or U(1) LGT systems with a flux cutoff of 1. The logical qubits were concatenated within a phase flip code to attain fault tolerance since phase flip errors commute with the Gauss' Law of these theories.

The construction proposed in Sec. 3.1 for the 1D case using an alternating encoding with two qubits per even link and one qubit per odd link requires $9N$ qubits (excluding ancillas) for the theory with non-dynamical fermions and for pure gauge theory. This improvement in space complexity can also be extended to accommodate dynamical fermions by using Gauss' law to fix errors on the sites and requires $15N$ qubits, excluding ancillas. The 2D scheme presented in Sec. 3.3 allows for an even more compact encoding where only $48N_xN_y$ qubits are required in the dynamical case and $36N_xN_y$ for the pure gauge case for a lattice with $8N_xN_y$ links and $4N_xN_y$ sites. Such costs cannot be attained from error correcting individual qubits only. Moreover, while we considered only local Gauss' law checks, examining the links coming in and out of larger spatial regions can be used identify errors of Pauli weight more than 2. It would be interesting to investigate how the present constructions can be generalized to increase the distance of the effective bit-flip code that LGTs provide via Gauss' law. We also note that when multiple logical qubits are encoded together, an even lower physical-to-logical qubit ratio is achievable [121, 122]. The question of whether more efficient encodings could be found by following a more holistic approach as opposed to those using only Gauss' Law on local degrees of freedom like sites, links, and plaquettes is left for future work.

Chapter 4

BETTI NUMBER ESTIMATION WITH PATH INTEGRAL MONTE CARLO

Existing literature on quantum algorithms for topological data analysis shows there are cases where they can provide exponential advantages over *deterministic* classical algorithms for TDA. The question of how *randomized* classical algorithms perform in this setting is comparably understudied. There are works studying generalizations of random walk operators corresponding to higher order Laplacians for simplicial complexes [123, 124, 125]. These are specific to the combinatorial Laplacian context and while they could lead to more efficient classical approaches, no such result based on these methods is known at present.

To address this gap, we present a high level overview of a randomized classical algorithm for computing approximate normalized Betti numbers of clique complexes in polynomial time under certain assumptions and present a more detailed analysis in the subsequent section. The implication of such an algorithm is that the sufficient conditions needed for quantum algorithms to provide an advantage for this problem are more subtle than initially believed and that simply having a high-dimensional vector space does not necessarily guarantee a super-polynomial speedup. For example, one case where a quantum advantage has been argued is when the clique density, i.e. the ratio of number of k-cliques in the input graph to the number of all possible k-cliques, is high [14, 15]. This corresponds to having a high-dimensional vector space of $(k-1)$-simplices in the associated clique complex. As a corollary of our constructions, we will show that there are cases where the clique density is high but where our randomized classical algorithm can achieve scaling that is polynomially equivalent to quantum algorithms.

We make the assumption throughout that our input graph G is polynomial in size and

is stored in classical memory. Given a set of k vertices in the graph, we can explicitly verify in $O(k^2)$ operations whether they form of a k-clique by checking whether all possible $\binom{k}{2} \approx O(k^2)$ pairs are connected by an edge.

Suppose G has n vertices and let $\mathcal{H}_k$ denote the vector space of all possible k-cliques (or $(k-1)$-simplices) on n vertices, which has $\dim \mathcal{H}_k = \binom{n}{k} = d_{k-1}$. Let $\mathcal{H}_k^G$ denote the vector space of k-cliques that are actually present in the graph. We can define the boundary operator $\partial_{k-1}^G \colon \mathcal{H}_k^G \to \mathcal{H}_{k-1}^G$ acting on $\mathcal{H}_k^G$ as in Definition 1.3.5 and the corresponding combinatorial Laplacian

$$\Delta_{k-1}^G = \partial_{k-1}^{G\dagger} \partial_{k-1}^G + \partial_k^G \partial_k^{G\dagger}. \tag{4.1}$$

Note that we do not need to explicitly perform matrix multiplication to determine Δ_{k-1}^G. We can efficiently compute its matrix elements via the definition of the boundary operator because they are both sparse matrices (see Corollary 4.2.2 for an upper bound on the sparsity of the boundary operator and combinatorial Laplacian).

The main idea of our algorithm is to simulate the imaginary-time dynamics of Δ_{k-1}^G via the path-integral Monte Carlo method to create a projector onto its kernel. Since Δ_{k-1}^G is Hermitian, it has a complete set of orthonormal eigenvectors. This implies that any unit vector $|\psi\rangle \in \mathcal{H}_k^G$ can be decomposed as

$$|\psi\rangle = \cos(\theta)|\psi_g\rangle + \sin(\theta)|\psi_b\rangle, \tag{4.2}$$

where $|\psi_g\rangle$ is the projection of $|\psi\rangle$ onto the kernel of Δ_{k-1}^G and $|\psi_b\rangle$ is its orthogonal complement in $\mathcal{H}_k^G$ with respect to the standard Hermitian inner product.

Then

$$e^{-\Delta_{k-1}^G t}|\psi\rangle = \cos(\theta)|\psi_g\rangle + e^{-\Delta_{k-1}^G t}\sin(\theta)|\psi_b\rangle. \tag{4.3}$$

Let $\Delta_{k-1}^G|\lambda_\mu\rangle = \lambda_\mu|\lambda_\mu\rangle$ such that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|\mathrm{Cl}_k(G)|}$, where $|\mathrm{Cl}_k(G)| = \dim \mathcal{H}_k^G$ is the number of k-cliques in the graph. Since Δ_{k-1}^G is positive semi-definite, we have

$$0 \leq \langle\psi|e^{-\Delta_{k-1}^G t}|\psi\rangle = \cos^2(\theta) + \sin^2(\theta)\langle\psi_b|e^{-\Delta_{k-1}^G t}|\psi_b\rangle \leq \cos^2(\theta) + \sin^2(\theta)e^{-\gamma_{\min} t}, \tag{4.4}$$

where $\gamma_{\min}$ is the smallest non-zero eigenvalue of Δ_{k-1}^G. If we then pick

$$t \geq \log(1/\epsilon)/\gamma_{\min}, \tag{4.5}$$

the expectation value will be at most $\cos^2(\theta) - O(\epsilon)$.

Since we do not a priori know the k-cliques that actually exist in the graph, $|\psi\rangle$ will need to be uniformly sampled from the space of all possible k-cliques $\mathcal{H}_k$ and checked to see if it corresponds to a state in the subspace $\mathcal{H}_k^G$. If $|\psi\rangle \in \mathcal{H}_k$ is such that it is a column of a Haar random unitary over $\mathcal{H}_k$, the expectation value in (4.4) will approach as t tends to infinity the Haar expectation of $\cos^2(\theta)$, which is

$$\mathbb{E}_{\text{Haar}}(\cos^2(\theta)) = \frac{\beta_{k-1}}{d_{k-1}}. \tag{4.6}$$

Note that here that matching the Haar average is necessary as we aim to have an unbiased estimator of the normalized Betti number; however, matching the higher moments of the Haar average is not necessary for our argument[1]. Thus performing imaginary time evolution and a Haar expectation value will give the required normalized Betti number.

Of course, we cannot directly implement the exponential of Δ_{k-1}^G on a classical computer in general. To circumvent this issue, consider a decomposition of the Hamiltonian of the form

$$\Delta_{k-1}^G = \sum_{p=1}^{D} c_p H_p, \tag{4.7}$$

where each H_p is one-sparse and unitary. Then the eigenvalues λ_{p_i,ν_i} of each H_p are $\pm c_p$, where ν_i is an index of the eigenvalue and p_i is the index of the Hamiltonian. The Jordan-Wigner decomposition of Δ_{k-1}^G in terms of fermionic operators is an example of such an efficient one-sparse and unitary decomposition.

With such a decomposition, we use a path-integral Monte-Carlo simulation of $\exp(-\Delta_{k-1}^G t)$ to perform the time-evolution and compute the expectation value while reducing the variance of the Haar averaging procedure via importance sampling. The path integral expansion

[1]More specifically, a unitary 1-design suffices.

works by first breaking up $e^{-t\Delta_{k-1}^G}$ into r timeslices as $(e^{-t\Delta_{k-1}^G/r})^r$, Trotterizing over the matrices in the one-sparse decomposition of Δ_{k-1}^G, and then expanding each one-sparse matrix in its eigenbasis. Since one sparse matrices can be efficiently diagonalized, this process is classically efficient.

As notation, let Γ denote a particular path of eigenvectors in the path integral representation, $W(\Gamma)$ be the product of overlaps between the eigenvectors and λ_{p_i,Γ_i} be the eigenvalue corresponding to the eigenvector that appears in the i^{th} step in the path Γ. Finally, let $\Pr(\Gamma)$ be a probability distribution from which the paths are drawn that can be chosen to reduce the variance (as is standard in importance sampling). We show in Section 4.1 that taking the Haar-expectation of the result leads to

$$\mathbb{E}_{\text{Haar}}(\cos^2(\theta)) = \frac{1}{d_{k-1}}\mathbb{E}_\Gamma\left(\frac{\exp\left(-\lambda_{p_1,\Gamma_1}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/2r\right)W(\Gamma)\delta_{k_1,k_{2rD}}}{\Pr(\Gamma)}\right). \tag{4.8}$$

We then average over a finite ensemble of these random paths to estimate the expectation value drawn from an appropriate probability distribution. We propose, for general purposes, a Metropolis-Hastings based algorithm for selecting appropriate paths in the decomposition that are unlikely to have zero values of $W(\Gamma)$.

The algorithm works by drawing an initial eigenstate of the first term in the one-sparse decomposition of Δ_{k-1}^G uniformly. Then a path is drawn by transitioning to one of the two possible connected eigenstates for it randomly. As the terms are Hermitian by assumption, the eigenvalues at each step in the path integral are the same up to a sign. This means that there are only two choices when constructing a path: either we choose to traverse the positive eigenvalue or the negative eigenvalue. Thus each path can be described using $O(\log(d_{k-1})rD)$ bits. A path that has non-zero overlaps between the neighboring eigenstates can then be selected in $O(\log(d_{k-1})rD)$ time. This is used as an initial guess that is improved using Metropolis-Hastings, wherein the probability of transitioning between two randomly chosen paths $\Gamma^{(a)}$ and $\Gamma^{(b)}$ is:

$$P(\Gamma^{(b)}|\Gamma^{(a)}) = \frac{\exp\left(-2\lambda_{p_1,\Gamma_1^{(b)}}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i^{(b)}}t/r\right)}{\exp\left(-2\lambda_{p_1,\Gamma_1^{(a)}}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i^{(a)}}t/r\right)}. \tag{4.9}$$

The equilibrium distribution leads to a thermal distribution over the path with

$$\Pr(\Gamma) = \frac{\exp\left(-2\lambda_{p_1,\Gamma_1}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/r\right)\delta_{\Gamma\in S_\Gamma}}{\sum_{\Gamma\in S_\Gamma}\exp\left(-2\lambda_{p_1,\Gamma_1}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/r\right)}, \tag{4.10}$$

where S_Γ is the set of all valid paths that start and end at the same vertex. The number of such updates needed to achieve this distribution (within fixed error) scales as $O(1/\gamma_M)$, where γ_M is the gap of the Markov chain. We show in Section 4.1 that, provided this gap is large, this distribution can be efficiently sampled from and forms a good choice for the importance distribution for the paths that minimizes the variance over the paths.

We ultimately find that the number of arithmetic operations needed to estimate the ratio of the kernel of Δ_{k-1}^G to the size of the set of all k-simplices within additive error ϵ is, assuming that the sample variance in the estimates yielded by Algorithm 1 is σ^2, is in

$$\widetilde{O}\left(\frac{\sigma^2}{\epsilon^2}\left(\frac{k^2 d_{k-1}}{|\mathrm{Cl}_k(G)|} + \frac{D^4}{\gamma_M}\frac{\kappa^3}{\epsilon}\left(\log(d_{k-1})D^{-2} + \frac{\kappa^3}{\epsilon}\right)\right)\right) \tag{4.11}$$

where κ is the ratio of the largest eigenvalue to the smallest non-zero eigenvalue, i.e. the condition number of the combinatorial Laplacian. This shows that even in cases where $|\mathrm{Cl}_k(G)|$ is exponentially large, we can use path integration to estimate the ratio of $\dim\ker\Delta_k$ to d_{k-1} using a number of operations that scales polynomially with the number of vertices n provided that σ, D, κ, and γ_M^{-1} are at most $\mathrm{poly}(n)$. We show that σ can be polynomially large in some cases in Section 4.1.

To summarize, the algorithm we propose proceeds as follows:

1. Given an input graph, sample a collection of k-points from the graph.

2. Explicitly verify whether they form a k-clique. If so, use this as a starting vertex in a Markov chain we construct with stationary distribution (4.10). The vertices are the

eigenvectors of the one-sparse matrices in the one-sparse decomposition of Δ_k and the edges are determined by the Metropolis-Hasting rule of (4.9).

3. Sample from the stationary distribution (4.10) of our Markov chain. This yields a path indicated by a set of labels corresponding to vertices in the Markov chain. Note that the starting and ending point of the path are the same and fixed prior to the sampling, as they are determined by the procedure in the previous step. The expression for the Haar average (4.8) shows that the choice of the initial vertex does not matter.

4. Compute the eigenvalues λ_{p_i} corresponding to the eigenvectors given in the sample and compute the expression in the parentheses of (4.8). This only requires knowing the eigenvalues corresponding to the 1 or 2 dimensional blocks of the one-sparse matrices that act on the initial vertex or simplex state and can therefore be efficiently determined.

5. Repeat steps 3-4 a number of times given by σ^2/ϵ^2 and compute the average as in (4.8) to get an estimate of the normalized Betti number β_k of the corresponding clique complex.

Quantum algorithms for TDA were thought to outperform classical counterparts in the clique dense case [15], but this algorithm serves as a counter-example. Thus while an exponentially large dimension is a necessary condition for an exponential speedup for quantum TDA, it is not a sufficient condition. This implies that further work is needed in order to understand when, and even if, quantum algorithms can provide truly exponential advantages relative to all classical randomized algorithms for TDA.

4.1 Dequantization using path integral Monte Carlo

As discussed in the previous section, our algorithm looks at imaginary time evolution of the Hermitian operator Δ_{k-1}^G. Given an upper bound on the sparsity s of the combinatorial

Algorithm 1: Classical randomized algorithm for Betti number computation.

Data: $k > 0, n > 0, N_{\text{samp}} > 0$, $t \geq 0$, $r \geq 0$, a function $\Pr(\Gamma)$ which assigns a non-zero probability to each vector $\Gamma \in \mathbb{R}^{2rD}$, a function $W(\Gamma) = \langle \lambda_{p_1,\Gamma_1} | \lambda_{p_2,\Gamma_2} \rangle \cdots \langle \lambda_{p_{2rD-1},\Gamma_{2rD-1}} | \lambda_{p_1,\Gamma_1} \rangle$ where $|\lambda_{p_j,\Gamma_j}\rangle$ is the Γ_j^{th} eigenvector of the one sparse matrix U_{p_j}.

Result: Estimate $\bar{E}$ which is an unbiased estimator of β_{k-1}/d_{k-1}

for q *from* 1 *to* N_{samp} **do**

$\Sigma \leftarrow$ a set of k points encoded as an integer

while Σ is not a $(k-1)$-simplex **do**

$\Sigma \leftarrow$ a random set of k points encoded as an integer

end

Draw a vector $\Gamma = [\Sigma, \Gamma_2, \ldots, \Gamma_{2rD}]$ from the probability distribution $\Pr(\Gamma)$.

$E_q \leftarrow \frac{1}{d_{k-1}} \left(\frac{\exp\left(-\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i} t/2r\right) W}{\Pr(\Gamma)} \right)$

end

$\bar{E} \leftarrow \frac{1}{N_{\text{samp}}} \sum_q E_q$ average of E.

Laplacian, we can obtain the decomposition $\Delta_{k-1}^G = \sum_{p=1}^{D} c_p H_p$, where H_p is one-sparse and unitary and $D = O(s^2)$, in polynomial time using distributed graph coloring algorithms. These algorithms also let us compute the position of the non-zero matrix element in row x of U_α using a number of queries to Δ_{k-1}^G that scales as $O(\log^*(D))$ [16]. We will provide explicit bounds on the sparsity of the combinatorial Laplacian in the subsequent section.

In order to set up the relevant path integrals, we must first employ a Trotter-decomposition. This allows us to represent the exponential in terms of exponentials of the one-sparse matrices, which can then be simulated through randomization. This leads us to the conclusion that

$$e^{-\Delta_{k-1}^G t} = (e^{-\Delta_{k-1}^G t/r})^r = \left(\prod_{p=1}^{D} e^{-c_p H_p t/2r} \prod_{p=D}^{1} e^{-c_p H_p t/2r} + O\left(\frac{\left(\sum_p |c_p|\right)^3 t^3}{r^3} \right) \right)^r . \quad (4.12)$$

As H_p is Hermitian, it has a complete set of eigenvectors. H_p is also one-sparse and since such matrices can be written as the direct sum of irreducible one and two-dimensional matrices, we can parameterize the eigenvectors to respect the structure of the two dimensional space via

$$H_p|\lambda_{p,\nu}\rangle = \lambda_{p,\nu}|\lambda_{p,\nu}\rangle \tag{4.13}$$

Note that each eigenvector $|\lambda_{p,\nu}\rangle$ is such that $\langle p|\lambda_{p,\nu}\rangle$ is non-zero for only two different computational basis vectors.

Note that the first term on the RHS of (4.12) contains $2rD$ terms. If we introduce a vector of indices $p = \{1,\ldots,r,1,\ldots,r,\ldots,r\}$ with $2rD$ entries denoted by p_i, we can express the expectation of the exponential of the boundary operator as (H stands for the Haar average)

$$\mathbb{E}^H_{|\psi\rangle}\left(\prod_{i=1}^{2rD} e^{-c_{p_i}H_{p_i}t/2r}\right) \coloneqq \mathbb{E}_H\langle\psi|\left(\prod_{i=1}^{2rD} e^{-c_{p_i}H_{p_i}t/2r}\right)|\psi\rangle. \tag{4.14}$$

Next we set up our path integrals by selecting sets of $2rD$ indices that correspond to the eigenstates that we transition to in the path integral. We denote such a path via the vector Γ where Γ_j corresponds to the index of the j^{th} eigenstate in the path. Using this notation we can insert resolutions of the identity of the form $\sum_{\Gamma_j}|\lambda_{p_j,\Gamma_j}\rangle\langle\lambda_{p_j,\Gamma_j}|$ consisting of the eigenvectors $|\lambda_{p_j,\Gamma_j}\rangle$ of each H_{p_j} in between each of the $2rD$ terms and defining $\rho = |\psi\rangle\langle\psi|$ gives

$$\begin{aligned}
&\mathbb{E}^H_{|\psi\rangle}\left(\prod_{i=1}^{2rD} e^{-c_{p_i}H_{p_i}t/2r}\right) = \mathbb{E}_H\ \mathrm{Tr}\left(\rho\prod_{i=1}^{2rD} e^{-c_{p_i}H_{p_i}t/2r}\right)\\
&= \mathbb{E}_H\mathrm{Tr}\left(\rho\sum_{\Gamma_1,\ldots,\Gamma_{2rD}}\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i}t/2r\right)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_1,\Gamma_1}|\cdots|\lambda_{p_{2rD},\Gamma_{2rD}}\rangle\langle\lambda_{p_{2rD},\Gamma_{2rD}}|\right)\\
&= \mathbb{E}_H\mathrm{Tr}\left(\rho\sum_{\Gamma_1,\ldots,\Gamma_{2rD}}\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i}t/2r\right)W(\Gamma)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_{2rD},\Gamma_{2rD}}|\right)\\
&= \mathbb{E}_H\mathrm{Tr}\left(\rho\,\mathbb{E}_\Gamma\frac{\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i}t/2r\right)W(\Gamma)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_{2rD},k_{2rD}}|}{\Pr(\Gamma)}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_H \mathbb{E}_\Gamma \mathrm{Tr}\left(\rho \frac{\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i} t/2r\right) W(\Gamma)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_{2rD},\Gamma_{2rD}}|}{\Pr(\Gamma)}\right) \\
&= \frac{1}{d_{k-1}} \mathbb{E}_\Gamma \mathrm{Tr}\left(\frac{\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i} t/2r\right) W(\Gamma)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_{2rD},\Gamma_{2rD}}|}{\Pr(\Gamma)}\right) \\
&= \frac{1}{d_{k-1}} \mathbb{E}_\Gamma \left(\frac{\exp\left(-\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i} t/2r\right) W(\Gamma)\delta_{\Gamma_1,\Gamma_{2rD}}}{\Pr(\Gamma)}\right)
\end{aligned} \tag{4.15}
$$

where we have defined for convenience the quantity

$$
W(\Gamma) = \langle\lambda_{p_1,\Gamma_1}|\lambda_{p_2,\Gamma_2}\rangle \cdots \langle\lambda_{p_{2rD-1},\Gamma_{2rD-1}}|\lambda_{p_{2rD},\Gamma_{2rD}}\rangle \tag{4.16}
$$

with $\Gamma = [\Gamma_1, \ldots, \Gamma_{2rD}]$. In the fourth line, we divided and multiplied by a probability $\Pr(\Gamma)$ to express the sum as an average, which allows to use importance sampling to minimize the variance via a judicious choice of $\Pr(\Gamma)$. In the last line, we used the fact that $p_{2rD} = p_1$ for the symmetric Trotter formula.

If we wish to estimate this value by sampling, the primary driver of the complexity will be the estimation of the expectation value through the sample mean which corresponds to the optimal unbiased estimator of the population mean. The number of samples scales with the variance of the set that one averages over and the variance over Γ of the above Haar expectation is then simply

$$
\begin{aligned}
&\mathbb{V}_\Gamma\left(\frac{1}{d_{k-1}}\mathrm{Tr}\left(\frac{\exp\left(-\sum_{i=1}\lambda_{p_i,\Gamma_i} t/2r\right) W(\Gamma)|\lambda_{p_1,\Gamma_1}\rangle\langle\lambda_{p_{2rD},\Gamma_{2rD}}|}{\Pr(\Gamma)}\right)\right) \\
&= \frac{1}{d_{k-1}^2}\sum_{\Gamma_1,\ldots,\Gamma_{2rD-1}} \frac{\exp\left(-2\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i} t/r\right)|W(\Gamma)|^2\delta_{\Gamma_1,\Gamma_{2rD}}}{\Pr(\Gamma)} \\
&\quad - \left(\frac{1}{d_{k-1}}\mathbb{E}_\Gamma\left(\frac{\exp\left(-\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i} t/2r\right) W(\Gamma)\delta_{\Gamma_1,\Gamma_{2rD}}}{\Pr(\Gamma)}\right)\right)^2 \\
&\leq \frac{1}{d_{k-1}^2}\sum_{\Gamma_1,\ldots,\Gamma_{2rD-1}} \frac{\exp\left(-2\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i} t/r\right)|W(\Gamma)|^2\delta_{\Gamma_1,\Gamma_{2rD}}}{\Pr(\Gamma)}
\end{aligned} \tag{4.17}
$$

There are many probability distributions that we could choose to sample from to minimize the variance in (4.17). The most straight forward distribution to choose, and the appropriate

one to pick in the limit of short t, is a uniform distribution. However, in practice the eigenvalues in the sum may have wildly varying sizes and so the importance of each of the different paths can swing substantially. A more natural choice to make for the probability of drawing each path is

$$\Pr(\Gamma) = \frac{\exp\left(-2\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i} t/r\right)\delta_{\Gamma\in S_\Gamma}}{\sum_{\Gamma\in S_\Gamma} \exp\left(-2\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i} t/r\right)}, \tag{4.18}$$

where S_Γ is the set of valid paths with $2rD$ vertices such that each edge corresponds to a path of connected eigenvectors for the one-sparse matrices used in the decomposition of Δ_{k-1}^G. The delta function constraint ensures that we further restrict to paths that are valid and have a non-zero expectation.

The central challenge in employing this formula is to estimate the value of the sums over the values of Γ_i. Let us further assume that the H_j in used in the derivation are Hermitian and unitary such that

$$\lambda_{p_i,\Gamma_i} = \pm c_{p_i}. \tag{4.19}$$

Furthermore, a one-sparse decomposition is chosen the matrix elements are all off diagonal with the exception of any diagonal matrix that appears in the decomposition. This can be seen explicitly using the discussion of the Jordan-Wigner representation of the Dirac operator. This means that each eigenvector couples to at most two eigenvectors. At most one term is diagonal in the standard Trotter decomposition of the Dirac operator [16]. A simple combinatorial argument leads to the conclusion that the total number of valid paths is at most $d_{k-1}2^{2r(D-1)-1}$.

Given this choice, the normalization constant (which is analogous to a partition function) can be expressed (assuming that the diagonal element is always $p_i = D$) as

$$\begin{aligned}
&\sum_{\Gamma\in G} \exp\left(-2\lambda_{p_1,\Gamma_1} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i} t/r\right) \\
&\quad = d_{k-1} 2^{2rD-1-r} \cosh(2c_{p_1} t/r) \prod_{i=2}^{2rD-1} (\cosh(c_{p_i} t/r)\delta_{p_i\neq D} + \delta_{p_i=D} e^{-\lambda_{p_i,\Gamma_i} t/r}/2).
\end{aligned} \tag{4.20}$$

This can be computed using $O(\text{poly}(rD))$ arithmetic operations and so does not ruin the efficiency of the algorithm. Note that were the sum over the $W(\Gamma)$ terms considered instead, then the result would be computationally difficult to compute as these terms generate correlations that would prevent us from performing an independent sum for each of the factors.

The variance σ^2 over the values of k chosen in the path integrals for the expression for the Haar average is then

$$\begin{aligned}
\sigma^2 &= \frac{1}{d_{k-1}^2}\left(\sum_{\Gamma\in S_\Gamma}|W(\Gamma)|^2\right)\left(\sum_{\Gamma\in S_\Gamma}\exp\left(-2\lambda_{p_1,\Gamma_1}t/r-\sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/r\right)\right)\\
&-\frac{1}{d_{k-1}^2}\left|\sum_{\Gamma\in S_\Gamma}\exp\left(-\lambda_{p_1,\Gamma_1}t/r-\sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/2r\right)W(\Gamma)\right|^2\\
&\leq \frac{2^{2rD}e^{2Dt\max_i\lambda_{p_i,\Gamma_i}}}{d_{k-1}}\\
&\leq \frac{2^{2rD}(1/\delta)^{2D\max_i|\lambda_{p_i,\Gamma_i}|/\gamma_{\min}}}{d_{k-1}}
\end{aligned} \tag{4.21}$$

This shows that the variance after making this substitution is precisely equal to the gap in the Cauchy-Schwarz inequalty in the latter sum. This suggests that in certain cases where the Cauchy-Schwarz inequality is tight, the variance may be extremely small given that we have the ability to sample from the distribution $\Pr(\Gamma)$.

4.1.1 Metropolis Hastings algorithm

Outside of specific cases such as graphs, it is difficult in general to sample from the probability distribution $\Pr(\Gamma)$ to employ the above variance reduction strategy. It is therefore necessary to provide a general method to obtain these samples if we wish to understand how we could address the problem more generally. One way to address the issue of how to sample from the distribution $\Pr(\Gamma)$ is to use the Metropolis Hastings algorithm. The idea behind the algorithm is to design a Markov chain whose stationary distribution equals our choice of $\Pr(\Gamma)$.

We first start with a connected, undirected graph on the set of all possible states $\Gamma_1, \Gamma_2, \ldots, \Gamma_{2rD-1}$ which represent the "paths" involved in our Trotter decomposition. Each vertex of the graph then represents one possible collection of values for $\Gamma_1, \ldots, \Gamma_{2rD-1}$.

At each vertex a, we have the following transition rules: if a neighboring vertex b is selected and the probability of transitioning

$$p_b \coloneqq \Pr(\Gamma)_b = \exp\left(-2\lambda_{p_1,\Gamma_1^{(b)}} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i^{(b)}} t/r\right) \tag{4.22}$$

is at least as great as the probability of remaining $p_a \coloneqq \Pr(\Gamma)_a$, we transition to b. If $p_b < p_a$, then we transition to b with probability p_b/p_a, where $b \in \mathcal{N}(a)$ with $\mathcal{N}(a)$ referring to the neighbors of a and

$$\frac{p_b}{p_a} = \frac{\exp\left(-2\lambda_{p_1,\Gamma_1^{(b)}} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i^{(b)}} t/r\right)}{\exp\left(-2\lambda_{p_1,\Gamma_1^{(a)}} t/r - \sum_{i=2}^{2rD-1} \lambda_{p_i,\Gamma_i^{(a)}} t/r\right)}. \tag{4.23}$$

Otherwise we remain at a with probability $1 - p_b/p_a$. Defining

$$p_{ab} \coloneqq \frac{1}{R} \min\left(1, \frac{p_b}{p_a}\right) \tag{4.24}$$

and

$$p_{aa} \coloneqq 1 - \sum_{b \neq a} p_{ab} \tag{4.25}$$

we can easily verify $p_a p_{ab} = p_b p_{ba}$. By the Fundamental Theorem of Markov Chains [126], it follows that the stationary probabilities are p_a as needed.

The cost of computing the ratio p_b/p_a is $O(rD)$ arithmetic operations, which coincides with the cost of performing an update. The number of such updates needed to reach δ error from the stationary distribution, where δ is the total variational distance (TVD) from the stationary distribution desired, is

$$T^* \in O\left(\frac{\log(1/\delta)}{\gamma_M}\right), \tag{4.26}$$

where $\gamma_M := 1 - \lambda_2$ is the eigenvalue gap of for the transition matrix p. This implies that the number of arithmetic operations needed to sample from a distribution that is $\delta-$close to

the stationary distribution is in

$$O\left(\frac{rD\log(1/\delta)}{\gamma_M}\right). \tag{4.27}$$

If one samples from a distribution, P', that is $\delta-$close to the intended distribution P then the expectation value of any function f is $|\sum_j (P(j)f(j)) - \sum_j (P'(j)f(j))| \leq \delta \max|f(j)|$. Similarly, the variance obeys $|\sum_j P'(j)f(j)^2 - (\sum_j P'(j)f(j))^2| \leq \mathbb{V}(f) + O(\delta \max|f(j)|^2)$. Thus if we want the error in the mean to be less than some error ϵ_M, we require

$$\delta = \epsilon_M / \max|f(j)|. \tag{4.28}$$

Note that in our situation

$$|f| = \left|\frac{\exp\left(-\lambda_{p_1,\Gamma_1}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/2r\right)W(\Gamma)}{\Pr(\Gamma)}\right| \tag{4.29}$$

where r is the number time-steps needed in the Trotterization procedure to attain a desired Trotter error ϵ_T. Further, we have the following bound from the Cauchy-Schwartz inequality

$$\begin{aligned}&\sum_{\Gamma_1,\ldots,\Gamma_{2rD-1}}\left|W(\Gamma)\exp\left(-2\lambda_{p_1,\Gamma_1}t/r - \sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/r\right)\right| \\ &\leq \sqrt{\sum_{\Gamma_1,\ldots,\Gamma_{2rD-1}}|W(\Gamma)|^2}\sqrt{\sum_{\Gamma\in S_\Gamma}\exp\left(-4\lambda_{p_1,\Gamma_1}t/r - 2\sum_{i=2}^{2rD-1}\lambda_{p_i,\Gamma_i}t/r\right)}\end{aligned} \tag{4.30}$$

Note that when we sum over a specific p_i, Γ_i in $|W(\Gamma)|^2$, we get 1 since the eigenvectors are orthonormal. Performing all $2rD-2$ sums therefore gives 1 from all the inner products and the last sum involving the $(2rD-1)$-th index gives a factor of d_{k-1}. Additionally, each λ_{p_i,Γ_i} can either be positive or negative but is upper-bounded by $\|\Delta^G_{k-1}\|_{\max} \leq \|\Delta^G_{k-1}\|_\infty = \gamma_{\max}$, i.e. the largest eigenvalue of Δ^G_{k-1}. We can then bound $f_{\max}$ as follows

$$|f| \leq f_{\max} \leq d_{k-1}e^{2\gamma_{\max}tD}2^{r(D-1/2)}. \tag{4.31}$$

4.1.2 Trotter error in path integration

From Corollary 12 of Ref. [127], the multiplicative Trotter error m for a p-th order Trotter formula is asymptotically bounded by

$$O\left(\alpha\left(\frac{t}{r}\right)^{p+1}\exp\left(\frac{2t}{r}\Upsilon\sum_{\ell=1}^{\Gamma}\|H_\ell\|\right)\right) \tag{4.32}$$

where the operator H is decomposed into a sum of Γ terms,

$$\alpha = \sum_{\ell_1,\ell_2,\ldots,\ell_{p+1}=1} \|[H_{\ell_{p+1}},\ldots,[H_{\ell_2},H_{\ell_1}]\cdots]\| \tag{4.33}$$

and Υ is the number of "stages" of the formula. For the symmetric Trotter-Suzuki formula, $\Upsilon = 2(5)^{q-1}$ for a TS formula of order $2q$, where $q = 1$ and $p = 2$ for our case. $\Gamma = 2rD$ and α can be upper bounded by

$$\alpha \leq (2rD)4\max_p(c_p)^3 \leq 8rD\gamma_{\max}^3 \tag{4.34}$$

Since this is the short time multiplicative error bound for simulating an operator for time t/r, we want to bound the resulting error when simulating for large t. To this end note that if we have an operator A we approximate by an operator B up to some multiplicative error m, then $B = A(I + mC)$ where C is an operator such that $\|C\| \leq 1$, and m is a constant. Then

$$\begin{aligned}
\|B\|^r &\leq (\|A\|(I + m\|C\|)^r \leq \|A\|^r\left(1 + \sum_{q=1}^{r}(m\|C\|)^q\binom{r}{q}\right) \\
&\leq \|A\|^r\left(1 + \sum_{q=1}^{r}m^q\binom{r}{q}\right) \leq \|A\|^r\left(1 + \sum_{q=1}^{r}\left(\frac{mre}{q}\right)^q\right) \\
&\leq \|A\|^r\left(1 + \sum_{q=1}^{r}(mre)^q\right) \leq \|A\|^r\left(1 + \frac{mre}{1-mre}\right).
\end{aligned} \tag{4.35}$$

Therefore the long-time multiplicative error is bounded by $mre/(1-mre)$ and we would like this to be less than some desired error $\epsilon_T > 0$. This implies that we must have $mre \leq \epsilon_T/(1+\epsilon_T) \leq \epsilon_T$. In our context, $A = e^{tH/r}$, B is an approximation to A as given by a Trotter formula, and m is the short-time multiplicative Trotter error bound cited above.

Using the bound on m and substituting in the parameters relevant for our situation, we have

$$mre \leq e\alpha \frac{t^3}{r^2} \exp\left(\frac{4t}{r}\sum_{\ell=1}^{2rD}\|H_\ell\|\right) \leq \epsilon_T. \tag{4.36}$$

If $r \geq \frac{4t}{\ln 2}\sum_\ell \|H_\ell\|$, then

$$mre \leq 2e\alpha \frac{t^3}{r} \leq 2e\alpha \frac{t^3}{r^2} \leq \epsilon_T \tag{4.37}$$

which implies that

$$r = t\max\left\{\left(\frac{4et\alpha}{\epsilon_T}\right)^{1/2}, \frac{4}{\ln 2}\sum_\ell \|H_\ell\|\right\}. \tag{4.38}$$

The former term dominates asymptotically, so we will henceforth take $r \in \Theta\left(t\left(\frac{4et\alpha}{\epsilon_T}\right)^{1/2}\right)$.

The systematic error in the estimate of the expectation value from the Trotter-Suzuki formula and the finite length Markov chain is at most

$$\|e^{-\Delta_{k-1}^G t}\|\epsilon_T + \epsilon_M := \epsilon_{TM}. \tag{4.39}$$

The total number of operations needed to draw a single sample from the distribution with bias at most ϵ_{TM} is from (4.27) in

$$O\left(\frac{rD\log(1/\delta)}{\gamma_M}\right) = O\left(\frac{\sqrt{\alpha}t^{3/2}D\log(f_{\max}/\epsilon_M)}{\sqrt{\epsilon_T}\gamma_M}\right). \tag{4.40}$$

Next taking $\epsilon_M = \epsilon_{TM}/2$ and similarly for $\|e^{-\Delta_{k-1}^G t}\|\epsilon_T$, we have that the number of operations needed to draw a sample with the required bias is in

$$O\left(\frac{\sqrt{\|e^{-\Delta_{k-1}^G t}\|\alpha}t^{3/2}D\log(f_{\max}/\epsilon_{TM})}{\sqrt{\epsilon_{TM}}\gamma_M}\right). \tag{4.41}$$

Finally, if we set the error ϵ_{TMH} to be the error also including the bias in the mean estimate of $\dim\ker\Delta_k$ from having $t = \log(1/\epsilon)/\gamma$, we have after choosing both sources of error to be equal that the systematic error can be made less than ϵ_{TMH} using a number of operations in

$$\widetilde{O}\left(\frac{\sqrt{\|e^{-\Delta_{k-1}^G \log(1/\epsilon_{TMH})/\gamma}\|\alpha}D\log(f_{\max}/\epsilon_{TMH})}{\sqrt{\epsilon_{TMH}}\gamma_M\gamma_{\min}^{3/2}}\right) \subseteq \widetilde{O}\left(\frac{D\sqrt{\alpha}\log(f_{\max}/\epsilon_{TMH})}{\sqrt{\epsilon_{TMH}}\gamma_M\gamma_{\min}^{3/2}}\right). \tag{4.42}$$

4.1.3 Sample bounds

We finally need to consider the sampling error ϵ_S that arises from taking only a finite number of samples. Standard probabilistic arguments show that the number of samples N_S needed to achieve a given ϵ_S scales as σ^2/ϵ_S^2. The mean-squared error ϵ^2 is then

$$\epsilon^2 = \epsilon_S^2 + \epsilon_{TMH}^2. \tag{4.43}$$

As before, we choose to make the two contributions to the error equal. There is a final source of complexity that needs to be considered though. Algorithm 1 begins by drawing a valid $(k-1)$-simplex to start at to ensure that we are within the space of interest. This means that we need to randomly draw vertices until we find a k-clique. The probability of drawing such a simplex is $|\mathrm{Cl}_k(G)|/d_{k-1}$, which is the clique density for the graph. Thus with high probability, a number of samples proportional to the reciprocal of this will be needed. Each such sample requires clique detection, which scales as $O(k^2)$. This leads us to a cost of

$$N_{\mathrm{op}} \in \widetilde{O}\left(\frac{k^2 d_{k-1}\sigma^2}{|\mathrm{Cl}_k(G)|\epsilon^2} + \frac{D\sqrt{\alpha}\log(f_{\max}/\epsilon)}{\epsilon^{5/2}}\frac{\sigma^2}{\gamma_M \gamma_{\min}^{3/2}}\right) \tag{4.44}$$

We now substitute $\alpha, f_{\max}$ for variables related directly to the properties of the combinatorial Laplacian. We substitute $t \geq \log(1/\epsilon)/\gamma_{\min}$ throughout and drop subdominant logarithmic terms. Firstly, from (4.34) we get

$$\alpha \leq 8rD\gamma_{\max}^3 \in \widetilde{\Theta}\left(\frac{Dt^{3/2}\alpha^{1/2}\gamma_{\max}^3}{\epsilon^{1/2}}\right) \implies \sqrt{\alpha} \in \widetilde{\Theta}\left(D\frac{\gamma_{\max}^3}{\gamma_{\min}^{3/2}\epsilon^{1/2}}\right). \tag{4.45}$$

This bound on $\sqrt{\alpha}$, Eq. (4.38), and the fact that Δ_{k-1}^G is positive semi-definite imply

$$r \in \widetilde{O}\left(\|e^{-\Delta_{k-1}^G t/2}\|\sqrt{\alpha}t^{3/2}/\sqrt{\epsilon}\right) = \widetilde{O}\left(D\gamma_{\max}^3/\epsilon\gamma_{\min}^3\right). \tag{4.46}$$

From (4.31), the logarithm scales as

$$\log\left(\frac{f_{\max}}{\epsilon}\right) \in \widetilde{\Theta}\left(\log(d_{k-1}) + \frac{D^2\gamma_{\max}^3}{\gamma_{\min}^3\epsilon}\right). \tag{4.47}$$

The prior bound on the variance in (4.21) evaluates to

$$\begin{aligned}\sigma^2 &\leq \frac{2^{2rD}(1/\epsilon)^{2D\max_i|\lambda_{p_i,\Gamma_i}|/\gamma_{\min}}}{d_{k-1}} \\ &\in \frac{2^{O(D^2\kappa^3/\epsilon))}(1/\epsilon)^{2D\kappa}}{d_{k-1}},\end{aligned} \tag{4.48}$$

where we have defined the condition number $\kappa = \gamma_{\max}/\gamma_{\min}$. Substituting these expressions in Eq. (4.44) then implies the number of operations for the algorithm obeys

$$\begin{aligned}N_{op} &\in \widetilde{O}\left(\frac{k^2 d_{k-1}\sigma^2}{|\mathrm{Cl}_k(G)|\epsilon^2} + \frac{D^4\sigma^2\gamma_{\max}^3}{\gamma_{\min}^3\epsilon^3\gamma_M}\left(\log(d_{k-1})D^{-2} + \frac{\gamma_{\max}^3}{\gamma^3\epsilon}\right)\right) \\ &\in \widetilde{O}\left(\frac{\sigma^2}{\epsilon^2}\left(\frac{k^2 d_{k-1}}{|\mathrm{Cl}_k(G)|} + \frac{D^4}{\gamma_M}\frac{\kappa^3}{\epsilon}\left(\log(d_{k-1})D^{-2} + \frac{\kappa^3}{\epsilon}\right)\right)\right).\end{aligned} \tag{4.49}$$

This implies that in the event we assume the worst case bound on the variance the total number of operations is in

$$N_{\mathrm{op}} \in \widetilde{O}\left(\frac{2^{O(D^2\kappa^3/\epsilon))}}{d_{k-1}\epsilon^{2+2D\kappa}}\left(\frac{k^2 d_{k-1}}{|\mathrm{Cl}_k(G)|} + \frac{D^4}{\gamma_M}\frac{\kappa^3}{\epsilon}\left(\log(d_{k-1})D^{-2} + \frac{\kappa^3}{\epsilon}\right)\right)\right). \tag{4.50}$$

This shows that under worst case scenario scaling for the variance, our algorithm is efficient if $D\kappa$ is poly-logarithmic in n, κ/ϵ is a constant and γ_M^{-1} and the inverse density of cliques are polynomial in n.

While the above restrictions on the situations where the classical randomized algorithm is efficient are significant, they do imply that the TDA algorithm can be efficient even in cases where $d_{k-1} = \binom{n}{k}$ is exponentially large provided the graph is clique-dense. This possibility is not obvious if one only compares to classical algorithms like diagonalization, which scales polynomially with the dimension.

Finally, the number of operations varies in (4.50) with the variance of the path integrals which we upper bound with an exponential in D. While this scaling may seem prohibitive in the case where the graph is nearly complete, the variance bound in this case is extremely loose and using a particular bound designed for this scenario yields much better scaling as we will see later.

4.2 Analysis

In order to give bounds on the sparsity of the combinatorial Laplacian, we first define a few terms. Let K be a simplicial complex with N vertices. The **up-degree** of a k-simplex $\sigma \in K$, denoted by $\deg_U(\sigma)$, is the number of $k+1$ simplices in K that σ is in the boundary of. The **lower-degree** of σ, denoted by $\deg_L(\sigma)$, is the number of $k-1$ simplices in K that are in the boundary of σ. If two k-simplices $\sigma_1, \sigma_2 \in K$ both contain a $k-1$ simplex in their boundary, they are said to be **lower adjacent**. If σ_1, σ_2 are in the boundary of a $k+1$ simplex, they are said to be **upper adjacent**. Lemma 3.2.4 of Ref. [128] shows that if σ_1, σ_2 are distinct and lower adjacent, their common $(k-1)$-simplex is $\sigma_1 \cap \sigma_2$ and is unique if it exists. Similarly, Lemma 3.2.2 of Ref. [128] shows that if σ_1 and σ_2 are upper adjacent, their common $(k+1)$-simplex is unique.

The down degree of a k-simplex $\sigma \in K$ is always $k+1$ for any simplicial complex simply because any k-simplex contains $\binom{k+1}{k} = k+1$ simplices of dimension $k-1$ in its boundary. For the up-degree, we have the following bound:

Proposition 4.2.1. *Let K be a simplicial complex with N vertices and let σ be a k-simplex in K. The up-degree of σ is bounded by $\min\{N-k-1, d\}$, where d is maximum (up)-degree of all the vertices in K.*

Proof. We present two arguments for the $N-k-1$ bound. The most simple argument is that the up-degree is bounded by the number of possible ways to extend a k-simplex to a $k+1$ simplex by adding another point. Since we have $N-k-1$ other points to choose from, the up-degree is bounded by this quantity.

A more rigorous argument is to consider the largest eigenvalue of the combinatorial Laplacian $\Delta_k = \partial_k^\dagger \partial_k + \partial_{k+1}\partial_{k+1}^\dagger$ has the bound $\lambda_{\max}(\Delta_k) \leq N$ [129]. The diagonal matrix elements of Δ_k are given in Theorem 3.3.4 of Ref. [128] as $\deg_U(\sigma_i) + k + 1$ for all the k-simplices $\sigma_i \in K$ when $k > 0$. Since Δ_k is a real symmetric matrix, the Courant-Fischer minimax principle and the preceding bound on its largest eigenvalue together imply $\deg_U(\sigma) + k + 1 \leq N$, which in turn shows $\deg_U(\sigma) \leq N - k - 1$.

The other upper bound of d holds because a vertex added to a k-simplex to turn it into a $(k+1)$-simplex has to be connected to all vertices in the k-simplex. The number of vertices connected to all the vertices in the k-simplex is upper bounded by d. Hence we may take $\deg_U(\sigma) \leq \min\{N-k-1,\ d\}$. □

Corollary 4.2.2. *∂_k has row-sparsity equal to $\min\{N-k-1, d\}$ and column sparsity equal to $k+1$. Δ_k has row and column sparsity bounded by $(k+1)(N-k-1)$.*

Proof. Recall that ∂_k acts on the vector space of k-simplices in K with the standard basis vectors corresponding to the k-simplices themselves. Its column sparsity equals the down-degree of any k-simplex in K, i.e. $k+1$, because ∂_k has non-zero elements in a column corresponding to a fixed k-simplex only when a $k-1$-simplex is the boundary of that k-simplex. Its row sparsity is given by the largest up-degree of all k-simplices in K and from the preceding proposition is bounded by $\min\{N-k-1,\ d\}$.

Δ_k is Hermitian, so the column and row sparsity of Δ_k coincide. It in fact equals the maximum of the number of k-simplices c' such that $c \cap c'$ is a $(k-1)$-simplex and $c \cup c'$ is *not* a $(k+1)$-simplex for all k-simplices c (see Theorem 3.3.4 of [128]). In order to construct for a given k-simplex c another k-simplex c' as above, we can remove any of the $k+1$ vertices from c and add any of the $(N-k-1)$ vertices not in c to form c', giving an upper bound of $(k+1)(N-k-1)$. □

Note that the bound on the sparsity of Δ_k only considers constructing k-simplices with a common $(k-1)$-simplex and does not take into account whether they form a $(k+1)$-simplex together. It is possible to give tighter bounds by using the largest degree d of a vertex. Thus another upper bound on the sparsity of Δ_k is $O(kd)$. Thus we can set $D = O(k^2 d^2)$ or $D = O(k^2(N-k)^2)$ for the number of terms D in the one-sparse decomposition of the combinatorial Laplacian depending on if d or $N-k-1$ is smaller (see the discussing preceding equation (4.12)).

We now consider a few cases where our algorithm can run efficiently. Our first example will be the extreme case where the input to the algorithm is the completely disconnected

graph on N vertices. The only non-zero Betti number of the associated clique complex in this case is $\beta_0 = N$. Since $\dim\ker(\Delta_0) = \beta_0 = N$, Δ_0 is the $N \times N$ zero matrix and κ is undefined. We therefore cannot directly use the preceding asymptotic expressions for the number of samples required. The analysis is simple if we refer to the general formula for σ^2 in (4.21) however and gives the following result:

Proposition 4.2.3. *Let K be the completely disconnected graph on N vertices. The variance of the path-integral Monte Carlo sampling procedure for the Betti numbers of K is 0.*

Proof. In this case, $\lambda = 0$ for all the eigenvalues of Δ_0. We can choose the standard basis vectors for the N-dimensional vector space of 0-simplices, where each basis vector corresponds to a 0-simplex, as the eigenvectors of Δ_0. Then $W(\Gamma)$ in (4.21) is merely a product of Kronecker delta functions. There are precisely N valid paths (loops) for the Markov chain corresponding to each of the N vertices, so $\sum_{\Gamma\in S_\Gamma} W(\Gamma) = \sum_{\Gamma\in S_\Gamma} |W(\Gamma)|^2 = \sum_{\Gamma\in S_\Gamma} = N$. The variance then reduces to $\sigma^2 = \frac{N^2}{d_0^2} - \frac{N^2}{d_0^2} = 0$. □

Even though the variance is precisely 0 for this case, we do not have a priori knowledge of the structure of the input graph from the standpoint of this algorithm, aside from an upper bound on the degree of the vertices. We only possess access to it via oracle queries that verify whether a collection of vertices forms a k-clique. Therefore, we cannot conclude anything definite about the Betti numbers with only a single sample in this situation as the zero variance result might suggest, unless given an additional promise that the *least* upper bound is a fixed value or 0.

Another simple case is when the input is a complete graph on $N = n + 1$ vertices:

Proposition 4.2.4. *Let K be the clique complex of the complete graph on $N = n+1$ vertices. The variance of the path-integral Monte Carlo sampling procedure for the Betti numbers of K is 0.*

Proof. The clique complex of this graph corresponds to an n-simplex. From Theorem 3.3.4 of Ref. [128], $(\Delta_k)_{ii} = \deg_U(\sigma_i) + k + 1$ and the off-diagonal entries are either 0 or ± 1 if

$k > 0$. We argue that in this case, all the off-diagonal entries are 0. From the same theorem, it suffices to show that for a fixed k-simplex σ_i, the sum of the number of other upper adjacent simplices and other non-lower adjacent simplices must equal $\binom{n+1}{k+1} - 1$ (since Δ_k is a $\binom{n+1}{k+1} \times \binom{n+1}{k+1}$ matrix).

We first calculate the up-degree of any k-simplex in our n-simplex. The number of $(k+1)$-simplices in an n-simplex is $\binom{n+1}{k+2}$. The number of k-simplices in a $(k+1)$-simplex is $k+2$. The number of k-simplices in an n-simplex is $\binom{n+1}{k+1}$. Intuitively, we can determine the up-degree by finding the number of k-simplices contained in the boundary of all $(k+1)$-simplices and dividing by the actual number of k-simplices in an n-simplex to account for over-counting. This quantity is precisely

$$\frac{(k+2)\binom{n+1}{k+2}}{\binom{n+1}{k+1}} = n - k$$

and we thus get that the diagonal entries of Δ_k are all equal to $n+1$ when $k > 0$ (note that this saturates the bound on the up-degree of a k-simplex in any simplicial complex given earlier with $N = n + 1$).

Now fix a particular k-simplex σ in the n-simplex. We want to count the number of *other* k-simplices that are *not* lower adjacent to σ. σ has $k+1$ simplices of dimension $k-1$ in its boundary. From the above result, each of these $(k-1)$-simplices has up-degree $n - (k-1) = n - k + 1$. Then the number of potential lower adjacent k-simplices is $(k+1)(n-k+1)$. But by the uniqueness of common upper simplices, we have counted σ $k+1$ times, one for each of the $(k-1)$-simplices in the boundary of σ. Thus the number of *other* lower adjacent simplices is $(k+1)(n-k+1) - (k+1) = (k+1)(n-k)$. Thus the number of *other* k-simplices *not* lower adjacent to σ is $\binom{n+1}{k+1} - (k+1)(n-k) - 1$. On the other hand, the number of *other* k-simplices that are upper adjacent to σ is given by multiplying the up-degree $n-k$ of σ, which gives the number of $k+1$ simplices that are upper-adjacent to σ, by the number of k-simplices in each $k+1$-simplex which is $k+2$. But this overcounts σ by $n-k$, so $(n-k)(k+2) - (n-k) = (n-k)(k+1)$. The sum of this and $\binom{n+1}{k+1} - (k+1)(n-k) - 1$ is clearly $\binom{n+1}{k+1} - 1$, so the only non-zero entries in Δ_k are the

diagonal ones when $k > 0$.

When $k = 0$, Theorem 3.3.4 in Ref. [128] shows the diagonal entries of Δ_0 are the degree of the vertices in our n-simplex, which is precisely n. It also implies all the off-diagonal entries are 0 since every vertex is upper-adjacent in an n-simplex. Thus in either the $k = 0$ or $k > 0$ case, Δ_k is proportional to the identity. We can then set $D = r = 1$ in (4.21) and the number of valid paths summed over is precisely d_k. By an analogous reasoning to the completely disconnected case, $\sum_{\Gamma \in S_\Gamma} W(\Gamma) = \sum_{\Gamma \in S_\Gamma} |W(\Gamma)|^2 = \sum_{\Gamma \in S_\Gamma} = d_k$ and

$$\sigma^2 = e^{-2t\lambda} \left(\frac{d_k^2}{d_k^2} - \frac{d_k^2}{d_k^2} \right) = 0,$$

where $\lambda = n + 1$ when $k > 0$ and $\lambda = n$ when $k = 0$. □

Analogous arguments on needing a promise on the least upper bound on the degree of the vertices hold as in the case of the completely disconnected graph above.

We briefly comment on the general behavior of some parameters in Equation (4.49). For κ, it is known that orientable simplicial complexes of dimension $d \leq 2$ have $\kappa \in O(n_k^2)$, where n_k denotes the number of k-simplices and $0 \leq k \leq 2$, and it is further conjectured that $\kappa \in O(n_k^{2/d})$ in most cases [83]. This latter bound is quite favourable for high-dimensional simplices, i.e. when $d \sim n$ and with the approximation that $n_k \sim 2^n$. While naive arguments from perturbation theory on the behavior of κ and σ^2 when Δ_k is perturbed slightly away from the two extreme cases considered above imply κ and σ^2 remain $O(1)$, it is difficult to give tight bounds on σ^2 in (4.50) for arbitrary simplicial complexes. The behavior of the Markov chain spectral gap γ_M is likewise generically difficult to determine, though we know from the Peron-Frobenius theorem that it has the bound $0 < \gamma_M \leq 1$.

4.3 Conclusion

We developed above a path integral Monte Carlo algorithm for estimating the normalized Betti numbers of an arbitrary clique complex via an estimate for ratio of the kernel of the combinatorial Laplacian Δ_k to the number of k-simplices (or $(k-1)$-cliques) present in

the clique complex. Unlike deterministic classical algorithms for estimating Betti numbers of simplicial complexes, this algorithm does not generically depend exponentially on the number of data points in the input graph and, given suitable bounds on the variance, performs well in the clique-dense regimes where quantum algorithms were thought to prove advantageous [14, 15].

Apers *et al.* in Ref. [130] subsequently developed another classical path-integral Monte Carlo algorithm which can also be efficient in some of the regimes where the quantum algorithm for Betti number estimation works best. They claim a runtime of $n^{O(\gamma^{-1}\log(1/\epsilon))}$ for general simplicial complexes, where n is the number of vertices, ϵ the additive error, and $\gamma \leq \lambda_2(\Delta_k)/\hat{\lambda}$ with $\hat{\lambda} \in \Theta(\lambda_{\max}(\Delta_k))$. For clique complexes in particular, their runtime is generically $\mathrm{poly}(n) \cdot (n/\hat{\lambda})^{O(\gamma^{-1}\log(1/\epsilon))}$, or $2^{O(\gamma^{-1}\log 1/\epsilon)}$ when k or the maximum up-degree (see the last section of Section 4.1 for the definition) of the k-simplices are $O(n)$.

For general simplical complexes, their runtime depends exponentially on γ^{-1} of Δ_k, which is an upper bound on the condition number κ. They thus require constant γ and ϵ for general simplicial complexes. By contrast, our classical algorithm above has a run time depending on fixed polynomials in γ^{-1} and ϵ^{-1}, and can thus tolerate $\mathrm{poly}(n)$ scaling for both provided σ^2 also depends polynomially on κ and $1/\epsilon$. For clique complexes, their best-case runtime is polynomial if $\gamma \in \Omega(1)$ and $\epsilon = 1/\mathrm{poly}(n)$, or $\gamma \in \Omega(1/\log n)$ and $\epsilon \in \Omega(1)$. Analogous conclusions hold for our classical algorithm in these regimes if we are once again given suitable promises on the polynomial dependence of σ^2 on those parameters. Given the difficulty in analyzing the dependence of σ^2 on κ and ϵ^{-1} in general, our algorithms cannot be easily compared. In the worst case bound for σ^2 given in (4.48), we would require $D\kappa$ to be polylogarithmic in n and κ/ϵ a constant at worst and thus also cannot tolerate inverse polynomial error scaling or spectral gap.

While we do not ultimately expect our classical TDA method to be efficient generically, there do exist sufficient conditions under which it can be used to extract Betti numbers in polynomial time without suffering from the exponential dependence on the number of data points characteristic of other classical TDA algorithms. Furthermore, it creates uncertainty

about the necessary and sufficient conditions for an exponential advantage for quantum TDA as the cases where this algorithm has an exponential advantage relative to the deterministic classical algorithm are analogous to those for quantum algorithms. This means that an identification of clear cases where exponential advantage is likely remains an important open problem within the domain of quantum algorithms for TDA.

Chapter 5

CONCLUDING REMARKS

We have presented three main results in this thesis. The first is the development of novel hybrid quantum algorithms that combine existing quantum simulation algorithms after an interaction picture transformation. The second is showing how Gauss' law symmetries in certain abelian LGTs can be leveraged to reduce the number of qubits needed to correct all single qubit errors on the system. The third is developing a classical randomized algorithm for topological data analysis. We briefly review these results and discuss avenues for future research in these topics.

5.1 Hybridized Quantum Algorithms in the Interaction Picture

In Chapter 2, we proved rigorous error bounds for a variety of hybridized quantum algorithms that take an interaction picture Hamiltonian as their input. Since these algorithms scale with the norm of the input Hamiltonian and since the interaction picture transformation is a unitary transformation, their query complexity is independent of the Hamiltonian summand whose interaction frame we entered. Improvements in gate complexity with respect to certain parameters compared to existing algorithms in the literature were also demonstrated for simulating the Schwinger model, collective neutrino oscillations, and the generic class of Hamiltonians subject to dynamical constraints.

One drawback of these methods is that even though the *query* complexity of the simulation algorithms is independent of the norm of summand we transform into the interaction frame of, it is still necessary to simulate the unitary operator involved in the interaction picture transformation (see equation (1.23)) at the gate level. If this unitary operator is difficult to implement, then the cost of implementing the interaction picture transformation may

dominate the *gate* complexity of the simulation. These hybrid simulation protocols therefore work best when applied to summands of a Hamiltonian that are simple to exponentiate such as diagonal, one-sparse, or fast-forwardable Hamiltonians. Developing simulation protocols that exploit the interaction picture transformation but that can also be applied to terms whose operator-exponential is difficult to implement is therefore an important generalization of these current methods.

Another area for further investigation is whether or not other combinations of quantum simulation algorithms combined with the interaction picture transformation can yield additional improvements. Since we perform the interaction picture transformation first, only simulation algorithms that act on time-dependent Hamiltonians such as continuous qDRIFT or Trotterization can applied in the outer-loop of the algorithms. Other possibilities include mixing the qDRIFT and Trotter steps or even applying an interaction picture transformation after one or more of these have been applied. Whether these alternatives yield any practical advantages will require further analytical and numerical investigations, like in Ref. [131].

Lastly, interaction picture methods are particularly suited to dealing with unbounded Hamiltonians. Since the exponential of an unbounded operator is a bounded one, moving into the interaction frame of an unbounded term in a Hamiltonian can be a powerful technique in the simulation of such systems. Such an analysis has been conducted for classes of unbounded Hamiltonians whose terms satisfy certain constraints [132], but it remains an open problem to extend them to broader classes of unbounded Hamiltonians, particularly in the context of continuous variable systems.

5.2 Error Correction with Gauge Symmetries

As we have primarily investigated 1+1 dimensional systems with a flux cutoff of 1 in Chapter 3, we outline the difficulties one encounters when attempting to generalize these procedures to arbitrary dimensions and flux cutoffs.

For further levels of concatenation, our techniques incorporating Gauss' law can only be applied at the highest logical level where the bit-flip encoding is implemented instead of

within every other level of the concatenation. Our approaches can conceivably be used to suppress simulation errors resulting from the approximation of the true time-evolution via methods such as Trotterization, but the impact of our approach on these errors is left for future work as not all them will lead to violations of Gauss' law.

For arbitrary flux cutoffs and with non-dynamical matter fields on the sites, the classical CNOT gates in Figure 3.4 must be replaced with a classically controlled version of the $\hat{W}_s$ operation described in Sec. 3.1. These are controlled on the state of an appropriate site qubit and can be implemented with adder circuits such as those described in Refs. [56, 133]. A fault-tolerant adder will however require non-Clifford operations for a binary integer encoding. In principle, these can be applied using standard techniques like state injection [134] even with our logical encoding. A more scalable solution however is to instead use a unary integer encoding, for which a fault-tolerant incrementer can be implemented using only Clifford gates. The more efficient construction relying on ancilla qubits to compute the XOR between a site and a link variable shown in Equation 3.1 can also be implemented in a straightforward way when using unary encoding. It has been shown recently that in the context of dynamical simulations, the cutoff only grows as polylog$(1/\varepsilon)$ in many physical situations [132] for a fixed error ε. Thus the scaling in the number of qubits needed for integrating our error correction procedures with quantum simulation of such systems in a unary encoding is more optimal than the naive expectation of linear in the cutoff. However, existing unary encodings require non-stabilizer codes, so constructing generalized unary encodings which are stabilizer codes and exploring how the constructions given in Figure 3.5 generalize to this setting is currently under investigation.

We also do not claim to encode the full continuous gauge group, as this would violate the fact that finite dimensional quantum systems which correct erasure have no continuous symmetries but only a discrete subgroup thereof [135]. As such, we only encode the U(1) symmetry group with a finite cutoff approximation. The construction of error correcting codes analogous to the ones presented here for other structure groups will therefore need to be tailored to their admissible discrete subgroups.

There is the final observation that the loop string hadron formalism allows the diagonalization of Gauss' law for non-abelian gauge theories, converting them into analogues of those in abelian gauge theories [57, 136]. It is expected that this will be an important step in extending these algorithms to more complex theories where additional commuting constraints are needed.

5.3 Quantum and Classical Algorithms for Topological Data Analysis

We have developed a randomized classical algorithm for estimating normalized Betti numbers of clique complexes. This algorithm can run in polynomial time under certain assumptions and without a generic exponential dependence on the number of vertices n in the input graph.

A major drawback of our algorithm is that although it depends polynomially on n, the condition number κ of the combinatorial Laplacian, and on the inverse error $1/\epsilon$, it also depends directly on the variance on the Markov chain sampling procedure. As the variance is generically difficult to estimate and may depend exponentially on n in some cases, this makes comparisons with quantum algorithms and other randomized classical algorithms difficult. While we have sufficient conditions for our algorithm to run in polynomial time in the generic and worst case scenarios of equations (4.49) and (4.50), giving tighter bounds for classes of simplicial complexes which are less trivial than those considered in this thesis but also frequently arise in practical applications is considerably more challenging. Giving analogous bounds the spectral gap γ_M of the Markov chain appearing in this algorithm or relating it to other properties of the input graph and the combinatorial Laplacian is another problem we leave for future work.

Recall that while the problem of determining the low-lying spectral density of the combinatorial Laplacian has been proven to be DQC1-hard for general cochain complexes, an analogous result for clique complexes remains an open problem [84]. In particular, and as pointed out in Apers et. al., it remains an open problem to prove or disprove the existence of a classical algorithm with a runtime of $\text{poly}(n, \kappa, 1/\epsilon)$ [137]. The resolution of this

problem on a complexity theoretic level may involve additional tools at the intersection of topology, complexity theory, and many-body physics beyond those used to prove the existing DQC1-hardness results.

BIBLIOGRAPHY

[1] Abhishek Rajput, Alessandro Roggero, and Nathan Wiebe. Hybridized Methods for Quantum Simulation in the Interaction Picture. *Quantum*, 6:780, August 2022. ISSN 2521-327X. doi:10.22331/q-2022-08-17-780. URL https://doi.org/10.22331/q-2022-08-17-780.

[2] Abhishek Rajput, Alessandro Roggero, and Nathan Wiebe. Quantum error correction with gauge symmetries. *npj Quantum Information*, 9:41, April 2023. ISSN 2056-6387. doi:10.22331/q-2022-08-17-780. URL https://doi.org/10.1038/s41534-023-00706-8.

[3] Dominic W. Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush. Quantifying quantum advantage in topological data analysis, 2023.

[4] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400:97–117, 1985.

[5] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982. ISSN 1572-9575. doi:10.1007/BF02650179.

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

[7] Alán Aspuru-Guzik, Anthony D Dutoi, Peter J Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005. doi:10.1126/science.1113479.

[8] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017. doi:10.1073/pnas.1619152114.

[9] Stephen P Jordan, Keith SM Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012. doi:10.1126/science.1217069.

[10] Alessandro Roggero, Andy C. Y. Li, Joseph Carlson, Rajan Gupta, and Gabriel N. Perdue. Quantum computing for neutrino-nucleus scattering. *Phys. Rev. D*, 101: 074038, Apr 2020. doi:10.1103/PhysRevD.101.074038.

[11] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. doi:10.1103/PhysRevLett.103.150502. URL https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[12] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700.

[13] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature Communications*, 7(1), 2016. doi:10.1038/ncomms10138.

[14] Shashanka Ubaru, Ismail Yunus Akhalwaya, Mark S. Squillante, Kenneth L. Clarkson, and Lior Horesh. Quantum topological data analysis with linear depth and exponential speedup. *arXiv:2108.02811*, 2021. URL https://arxiv.org/abs/2108.02811.

[15] Casper Gyurik, Chris Cade, and Vedran Dunjko. Towards quantum advantage via topological data analysis. *arXiv:2005.02607*, 2020. URL https://arxiv.org/abs/2005.02607.

[16] 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, 2007. doi:https://doi.org/10.1007/s00220-006-0150-x.

[17] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C Sanders. Higher order decompositions of ordered operator exponentials. *Journal of Physics A: Mathematical and Theoretical*, 43(6):065203, 2010. doi:10.1088/1751-8113/43/6/065203.

[18] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space. *Physical Review Letters*, 106(17), Apr 2011. ISSN 1079-7114. doi:10.1103/physrevlett.106.170501.

[19] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021. doi:10.1103/PhysRevX.11.011020.

[20] Dave Wecker, Matthew B. Hastings, Nathan Wiebe, Bryan K. Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A*, 92:062318, Dec 2015. doi:10.1103/PhysRevA.92.062318. URL https://link.aps.org/doi/10.1103/PhysRevA.92.062318.

[21] 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.

[22] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, Jul 2019. ISSN 2521-327X. doi:10.22331/q-2019-07-12-163.

[23] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arith-

metics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. doi:10.1145/3313276.3316366.

[24] Dominic W Berry, Mária Kieferová, Artur Scherer, Yuval R Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. Improved techniques for preparing eigenstates of fermionic hamiltonians. *npj Quantum Information*, 4(1):1–7, 2018. doi:10.1038/s41534-018-0071-5.

[25] David Poulin, Alexei Kitaev, Damian S Steiger, Matthew B Hastings, and Matthias Troyer. Quantum algorithm for spectral measurement with a lower gate count. *Physical review letters*, 121(1):010501, 2018. doi:10.1103/PhysRevLett.121.010501.

[26] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4), dec 2021. doi:10.1103/prxquantum.2.040203.

[27] Yulong Dong, Xiang Meng, K Birgitta Whaley, and Lin Lin. Efficient phase-factor evaluation in quantum signal processing. *Physical Review A*, 103(4):042419, 2021. doi:10.1103/PhysRevA.103.042419.

[28] Earl Campbell. Random compiler for fast hamiltonian simulation. *Physical Review Letters*, 123(7), Aug 2019. ISSN 1079-7114. doi:10.1103/physrevlett.123.070503. URL http://dx.doi.org/10.1103/PhysRevLett.123.070503.

[29] Dominic W Berry, Andrew M Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent hamiltonian simulation with $l1$-norm scaling. *Quantum*, 4:254, 2020. doi:10.22331/q-2020-04-20-254.

[30] Yuan Su, Dominic W Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. *arXiv preprint arXiv:2105.12767*, 2021. URL https://doi.org/10.48550/arXiv.2105.12767.

[31] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear t complexity. *Physical Review X*, 8(4), Oct 2018. ISSN 2160-3308. doi:10.1103/physrevx.8.041015. URL http://dx.doi.org/10.1103/PhysRevX.8.041015.

[32] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la Société Mathématique de France*, 3:103–174, 1875. URL http://eudml.org/doc/85325.

[33] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of resonant equiangular composite quantum gates. *Physical Review X*, 6(4), Dec 2016. ISSN 2160-3308. doi:10.1103/physrevx.6.041067. URL http://dx.doi.org/10.1103/PhysRevX.6.041067.

[34] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, and Mario Szegedy. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*, 2020. URL https://doi.org/10.48550/arXiv.2003.02831.

[35] Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, 2019. doi:10.22331/q-2019-10-07-190.

[36] S. Tomonaga. On a Relativistically Invariant Formulation of the Quantum Theory of Wave Fields*. *Progress of Theoretical Physics*, 1(2):27–42, 08 1946. ISSN 0033-068X. doi:10.1143/PTP.1.27. URL https://doi.org/10.1143/PTP.1.27.

[37] Julian Schwinger. Quantum electrodynamics. i. a covariant formulation. *Phys. Rev.*, 74: 1439–1461, Nov 1948. doi:10.1103/PhysRev.74.1439. URL https://link.aps.org/doi/10.1103/PhysRev.74.1439.

[38] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2 edition, 2017. doi:10.1017/9781108499996.

[39] Steven Weinberg. *The Quantum Theory of Fields*, volume 1. Cambridge University Press, 1995. doi:10.1017/CBO9781139644167.

[40] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995. ISBN 978-0-201-50397-5.

[41] Matthew D. Schwartz. *Quantum Field Theory and the Standard Model*. Cambridge University Press, 2013. doi:10.1017/9781139540940.

[42] John Kogut and Leonard Susskind. Hamiltonian formulation of wilson's lattice gauge theories. *Phys. Rev. D*, 11:395–408, Jan 1975. doi:10.1103/PhysRevD.11.395.

[43] John B. Kogut. An introduction to lattice gauge theory and spin systems. *Rev. Mod. Phys.*, 51:659–713, Oct 1979. doi:10.1103/RevModPhys.51.659. URL https://link.aps.org/doi/10.1103/RevModPhys.51.659.

[44] S. Durr, Z. Fodor, J. Frison, C. Hoelbling, R. Hoffmann, S. D. Katz, S. Krieg, T. Kurth, L. Lellouch, T. Lippert, and et al. Ab initio determination of light hadron masses. *Science*, 322(5905):1224–1227, Nov 2008. ISSN 1095-9203. doi:10.1126/science.1163233. URL http://dx.doi.org/10.1126/science.1163233.

[45] Andreas S. Kronfeld. Twenty-first century lattice gauge theory: Results from the quantum chromodynamics lagrangian. *Annual Review of Nuclear and Particle Science*, 62(1):265–284, Nov 2012. ISSN 1545-4134. doi:10.1146/annurev-nucl-102711-094942. URL http://dx.doi.org/10.1146/annurev-nucl-102711-094942.

[46] S. Aoki et al. FLAG Review 2019: Flavour Lattice Averaging Group (FLAG). *Eur. Phys. J. C*, 80(2):113, 2020. doi:10.1140/epjc/s10052-019-7354-7.

[47] Erez Zohar, J Ignacio Cirac, and Benni Reznik. Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices. *Reports on Progress in Physics*, 79 (1):014401, Dec 2015. ISSN 1361-6633. doi:10.1088/0034-4885/79/1/014401. URL http://dx.doi.org/10.1088/0034-4885/79/1/014401.

[48] M. Dalmonte and S. Montangero. Lattice gauge theory simulations in the quantum information era. *Contemporary Physics*, 57(3):388–412, Mar 2016. ISSN 1366-5812. doi:10.1080/00107514.2016.1151199. URL http://dx.doi.org/10.1080/00107514.2016.1151199.

[49] Mari Carmen Bañuls, Rainer Blatt, Jacopo Catani, Alessio Celi, Juan Ignacio Cirac, Marcello Dalmonte, Leonardo Fallani, Karl Jansen, Maciej Lewenstein, Christine A. Montangero, Simoneand Muschik, Benni Reznik, Enrique Rico, Luca Tagliacozzo, Karel Van Acoleyen, Frank Verstraete, Uwe-Jens Wiese, Matthew Wingate, Jakub Zakrzewski, and Peter Zoller. Simulating lattice gauge theories within quantum technologies. *Eur. Phys. J. D*, 74:165, 2020. doi:10.1140/epjd/e2020-100571-8. URL https://doi.org/10.1140/epjd/e2020-100571-8.

[50] Natalie Klco, Alessandro Roggero, and Martin J. Savage. Standard model physics and the digital quantum revolution: Thoughts about the interface, 2021.

[51] Jad C. Halimeh and Philipp Hauke. Reliability of lattice gauge theories. *Phys. Rev. Lett.*, 125:030503, Jul 2020. doi:10.1103/PhysRevLett.125.030503. URL https://link.aps.org/doi/10.1103/PhysRevLett.125.030503.

[52] P. Hauke, D. Marcos, M. Dalmonte, and P. Zoller. Quantum simulation of a lattice schwinger model in a chain of trapped ions. *Phys. Rev. X*, 3:041018, Nov 2013. doi:10.1103/PhysRevX.3.041018.

[53] Maarten Van Damme, Jad C. Halimeh, and Philipp Hauke. Gauge-symmetry violation quantum phase transition in lattice gauge theories, 2020.

[54] Henry Lamm, Scott Lawrence, and Yukari Yamauchi. Suppressing coherent gauge drift in quantum simulations, 2020.

[55] Minh C. Tran, Yuan Su, Daniel Carney, and Jacob M. Taylor. Faster digital quantum simulation by symmetry protection. *PRX Quantum*, 2:010323, Feb 2021.

doi:10.1103/PRXQuantum.2.010323. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.010323.

[56] Jesse R. Stryker. Oracles for gauss's law on digital quantum computers. *Physical Review A*, 99(4), Apr 2019. ISSN 2469-9934. doi:10.1103/physreva.99.042301. URL http://dx.doi.org/10.1103/PhysRevA.99.042301.

[57] Indrakshi Raychowdhury and Jesse R. Stryker. Solving gauss's law on digital quantum computers with loop-string-hadron digitization. *Phys. Rev. Research*, 2:033039, Jul 2020. doi:10.1103/PhysRevResearch.2.033039. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033039.

[58] Natalie Klco and Martin J. Savage. Hierarchical qubit maps and hierarchical quantum error correction, 2021.

[59] Julian Schwinger. Gauge invariance and mass. ii. *Phys. Rev.*, 128:2425–2429, Dec 1962. doi:10.1103/PhysRev.128.2425. URL https://link.aps.org/doi/10.1103/PhysRev.128.2425.

[60] Sidney Coleman, R Jackiw, and Leonard Susskind. Charge shielding and quark confinement in the massive schwinger model. *Annals of Physics*, 93(1):267–275, 1975. ISSN 0003-4916. doi:https://doi.org/10.1016/0003-4916(75)90212-2. URL https://www.sciencedirect.com/science/article/pii/0003491675902122.

[61] M.C. Bañuls, K. Cichy, J.I. Cirac, and K. Jansen. The mass spectrum of the schwinger model with matrix product states. *Journal of High Energy Physics*, 2013(11), Nov 2013. ISSN 1029-8479. doi:10.1007/jhep11(2013)158. URL http://dx.doi.org/10.1007/JHEP11(2013)158.

[62] T. Pichler, M. Dalmonte, E. Rico, P. Zoller, and S. Montangero. Real-time dynamics in u(1) lattice gauge theories with tensor networks. *Phys. Rev. X*, 6:011023, Mar 2016. doi:10.1103/PhysRevX.6.011023.

[63] Esteban A. Martinez, Christine A. Muschik, Philipp Schindler, Daniel Nigg, Alexander Erhard, Markus Heyl, Philipp Hauke, Marcello Dalmonte, Thomas Monz, Peter Zoller, and et al. Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature*, 534(7608):516–519, Jun 2016. ISSN 1476-4687. doi:10.1038/nature18318. URL http://dx.doi.org/10.1038/nature18318.

[64] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage. Quantum-classical computation of schwinger model dynamics using quantum computers. *Phys. Rev. A*, 98:032331, Sep 2018. doi:10.1103/PhysRevA.98.032331. URL https://link.aps.org/doi/10.1103/PhysRevA.98.032331.

[65] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011. ISBN 1107002176.

[66] Rui Chao and Ben W. Reichardt. Quantum error correction with only two extra qubits. *Physical Review Letters*, 121(5), Aug 2018. ISSN 1079-7114. doi:10.1103/physrevlett.121.050502. URL http://dx.doi.org/10.1103/PhysRevLett.121.050502.

[67] Volker Kaibel and Marc E. Pfetsch. Some algorithmic problems in polytope theory, 2002.

[68] Vin Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic and Geometric Topology*, 7, 04 2007. doi:10.2140/agt.2007.7.339.

[69] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33:249–274, 2005. doi:10.1007/s00454-004-1146-y. URL https://doi.org/10.1007/s00454-004-1146-y.

[70] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008. doi:10.1090/S0273-0979-07-01191-3.

[71] Frederic Chazal and Andre Lieutier. Stability and computation of topological invariants of solids in $\mathbb{R}^n$. *Discrete Computational Geometry*, 37(4):601–617, 2007. doi:10.1007/s00454-007-1309-8.

[72] S. Basu. On bounding the betti numbers and computing the euler characteristic of semi-algebraic sets. *Discrete Computational Geometry*, 22(1):1–18, 1999. doi:10.1007/pl00009443.

[73] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 217–228, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi:10.1145/3313276.3316310. URL https://doi.org/10.1145/3313276.3316310.

[74] Ewin Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Phys. Rev. Lett.*, 127:060503, Aug 2021. doi:10.1103/PhysRevLett.127.060503. URL https://link.aps.org/doi/10.1103/PhysRevLett.127.060503.

[75] Sam Gunn and Niels Kornerup. Review of a quantum algorithm for Betti numbers. *arXiv:1906.07673*, 2019. URL https://arxiv.org/abs/1906.07673.

[76] John Lee. *Introduction to topological manifolds*, volume 202. Springer Science & Business Media, 2010.

[77] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[78] Michał Adamaszek and Juraj Stacho. Complexity of simplicial homology and independence complexes of chordal graphs. *Computational Geometry*, 57:8–18, 2016. doi:10.1016/j.comgeo.2016.05.003.

[79] Peter Scheiblechner. On the complexity of deciding connectedness and computing Betti numbers of a complex algebraic variety. *Journal of Complexity*, 23(3):359–379, 2007. doi:10.1016/j.jco.2007.03.008.

[80] Marcos Crichigno and Tamara Kohler. Clique homology is QMA1-hard. *arXiv:2209.11793*, 2022. URL https://arxiv.org/abs/2209.11793.

[81] Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47(3):274–291, 1989. doi:10.1016/0095-8956(89)90029-4.

[82] Anna Gundert and May Szedláky. Higher dimensional discrete Cheeger inequalities. *Journal of Computational Geometry*, 6(2):54, 2015. doi:10.20382/jocg.v6i2a4.

[83] Joel Friedman. Computing Betti numbers via combinatorial Laplacians. *Algorithmica*, 21(4):331–346, 1998. doi:10.1007/PL00009218.

[84] Chris Cade and P. Marcos Crichigno. Complexity of supersymmetric systems and the cohomology problem. *arXiv:2107.00011*, 2021. URL https://arxiv.org/abs/2107.00011.

[85] Dave Wecker, Matthew B Hastings, Nathan Wiebe, Bryan K Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *Physical Review A*, 92(6):062318, 2015. doi:10.1103/PhysRevA.92.062318.

[86] T. Banks, Leonard Susskind, and John Kogut. Strong-coupling calculations of lattice gauge theories: (1 + 1)-dimensional exercises. *Phys. Rev. D*, 13:1043–1053, Feb 1976. doi:10.1103/PhysRevD.13.1043.

[87] Yuval R. Sanders, Dominic W. Berry, Pedro C.S. Costa, Louis W. Tessler, Nathan Wiebe, Craig Gidney, Hartmut Neven, and Ryan Babbush. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum*, 1(2), Nov 2020. ISSN 2691-3399. doi:10.1103/prxquantum.1.020312. URL http://dx.doi.org/10.1103/PRXQuantum.1.020312.

[88] Yong He, Mingxing Luo, E. Zhang, Hong-Ke Wang, and Xiao-Feng Wang. Decompositions of n-qubit toffoli gates with linear circuit complexity. *International Journal of Theoretical Physics*, 56, 07 2017. doi:10.1007/s10773-017-3389-4.

[89] Johannes Bausch. Fast black-box quantum state preparation, 2020. URL https://arxiv.org/abs/2009.10709.

[90] Cody Jones. Low-overhead constructions for the fault-tolerant toffoli gate. *Physical Review A*, 87(2):022328, 2013. doi:10.1103/PhysRevA.87.022328.

[91] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004. URL https://doi.org/10.48550/arXiv.quant-ph/0410184.

[92] Alexander F. Shaw, Pavel Lougovski, Jesse R. Stryker, and Nathan Wiebe. Quantum algorithms for simulating the lattice schwinger model. *Quantum*, 4:306, Aug 2020. ISSN 2521-327X. doi:10.22331/q-2020-08-10-306. URL http://dx.doi.org/10.22331/q-2020-08-10-306.

[93] James Pantaleone. Neutrino oscillations at high densities. *Physics Letters B*, 287(1): 128 – 132, 1992. ISSN 0370-2693. doi:https://doi.org/10.1016/0370-2693(92)91887-F.

[94] Huaiyu Duan, George M. Fuller, J. Carlson, and Yong-Zhong Qian. Coherent development of neutrino flavor in the supernova environment. *Phys. Rev. Lett.*, 97:241101, Dec 2006. doi:10.1103/PhysRevLett.97.241101.

[95] Huaiyu Duan, George M. Fuller, and Yong-Zhong Qian. Collective neutrino oscillations. *Annual Review of Nuclear and Particle Science*, 60(1):569–594, 2010. doi:10.1146/annurev.nucl.012809.104524. URL https://doi.org/10.1146/annurev.nucl.012809.104524.

[96] Sovan Chakraborty, Rasmus Hansen, Ignacio Izaguirre, and Georg Raffelt. Collective neutrino flavor conversion: Recent developments. *Nuclear Physics B*, 908:366 – 381, 2016. ISSN 0550-3213. doi:https://doi.org/10.1016/j.nuclphysb.2016.02.012. Neutrino Oscillations: Celebrating the Nobel Prize in Physics 2015.

[97] Ermal Rrapaj. Exact solution of multiangle quantum many-body collective neutrino-flavor oscillations. *Phys. Rev. C*, 101:065805, Jun 2020. doi:10.1103/PhysRevC.101.065805.

[98] Michael J. Cervia, Amol V. Patwardhan, A. B. Balantekin, S. N. Coppersmith, and Calvin W. Johnson. Entanglement and collective flavor oscillations in a dense neutrino gas. *Phys. Rev. D*, 100:083001, Oct 2019. doi:10.1103/PhysRevD.100.083001.

[99] Alessandro Roggero. Entanglement and many-body effects in collective neutrino oscillations. *Phys. Rev. D*, 104:103016, Nov 2021. doi:10.1103/PhysRevD.104.103016.

[100] Alessandro Roggero. Dynamical phase transitions in models of collective neutrino oscillations. *Phys. Rev. D*, 104:123023, Dec 2021. doi:10.1103/PhysRevD.104.123023.

[101] Benjamin Hall, Alessandro Roggero, Alessandro Baroni, and Joseph Carlson. Simulation of collective neutrino oscillations on a quantum computer. *Phys. Rev. D*, 104: 063009, Sep 2021. doi:10.1103/PhysRevD.104.063009.

[102] Kübra Yeter-Aydeniz, Shikha Bangar, George Siopsis, and Raphael C. Pooser. Collective neutrino oscillations on a quantum computer. *Quantum Information Processing*, 2021. URL https://doi.org/10.1007/s11128-021-03348-x.

[103] Y. Pehlivan, A. B. Balantekin, Toshitaka Kajino, and Takashi Yoshida. Invariants of collective neutrino oscillations. *Phys. Rev. D*, 84:065008, Sep 2011. doi:10.1103/PhysRevD.84.065008.

[104] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Phys. Rev. Lett.*, 120:110501, Mar 2018. doi:10.1103/PhysRevLett.120.110501.

[105] Mária Kieferová, Artur Scherer, and Dominic W Berry. Simulating the dynamics of time-dependent hamiltonians with a truncated dyson series. *Physical Review A*, 99(4): 042314, 2019. doi:10.1103/PhysRevA.99.042314.

[106] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint arXiv:1805.00675*, 2018. doi:10.48550/ARXIV.1805.00675.

[107] Roberto Oliveira and Barbara M. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice, 2005. URL https://doi.org/10.48550/ARXIV.QUANT-PH/0504050.

[108] Yudong Cao and Sabre Kais. Efficient optimization of perturbative gadgets, 2017. URL https://doi.org/10.48550/arXiv.1709.02705.

[109] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[110] Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982. doi:10.1088/0305-4470/15/10/028.

[111] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceed-*

ings of the thirty-fifth annual ACM symposium on Theory of computing, pages 59–68, 2003. doi:10.1145/780542.780552.

[112] Jesse R Stryker. Oracles for gauss's law on digital quantum computers. *Physical Review A*, 99(4):042301, 2019. doi:10.1103/PhysRevA.99.042301.

[113] Julia Kempe and Oded Regev. 3-local hamiltonian is qma-complete. *arXiv preprint quant-ph/0302079*, 2003. URL https://doi.org/10.48550/arXiv.quant-ph/0302079.

[114] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008. doi:10.1137/080734479.

[115] Tobias J Osborne. Hamiltonian complexity. *Reports on progress in physics*, 75(2): 022001, 2012. doi:10.1088/0034-4885/75/2/022001.

[116] Tamara Kohler, Stephen Piddock, Johannes Bausch, and Toby Cubitt. Translationally invariant universal quantum hamiltonians in 1d. In *Annales Henri Poincaré*, volume 23, pages 223–254. Springer, 2022. doi:10.1007/s00023-021-01111-7.

[117] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. doi:10.1103/PhysRevA.52.R2493. URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[118] A.D. Córcoles, Easwar Magesan, Srikanth J. Srinivasan, Andrew W. Cross, M. Steffen, Jay M. Gambetta, and Jerry M. Chow. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nat. Comm.*, 6:6979, 2015. doi:10.1038/ncomms7979. URL https://doi.org/10.1038/ncomms7979.

[119] David P. DiVincenzo and Peter W. Shor. Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.*, 77:3260–3263, Oct 1996.

doi:10.1103/PhysRevLett.77.3260. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.3260.

[120] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Phys. Rev. Lett.*, 77:198–201, Jul 1996. doi:10.1103/PhysRevLett.77.198. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.198.

[121] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, 54:1862–1868, Sep 1996. doi:10.1103/PhysRevA.54.1862. URL https://link.aps.org/doi/10.1103/PhysRevA.54.1862.

[122] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.*, 78:405–408, Jan 1997. doi:10.1103/PhysRevLett.78.405. URL https://link.aps.org/doi/10.1103/PhysRevLett.78.405.

[123] David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. doi:https://doi.org/10.1145/3219819.3220119.

[124] Sayan Mukherjee and John Steenbergen. Random walks on simplicial complexes and harmonics. *Random structures & algorithms*, 49:379–405, 2016. doi:https://doi.org/10.1002/rsa.20645.

[125] Ori Parzanchevski and Ron Rosenthal. Simplicial complexes: spectrum, homology and random walks. *Random Structures & Algorithms*, 50:225–261, 2017. doi:https://doi.org/10.1002/rsa.20657.

[126] Olle Häggström. *Finite Markov chains and algorithmic applications*. Cambridge University Press, 2002.

[127] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021. doi:10.1103/PhysRevX.11.011020. URL https://link.aps.org/doi/10.1103/PhysRevX.11.011020.

[128] Timothy E Goldberg. Combinatorial laplacians of simplicial complexes. *Senior Thesis, Bard College*, 2002.

[129] Art Duval and Victor Reiner. Shifted simplicial complexes are laplacian integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002.

[130] Simon Apers, Sayantan Sen, and Dániel Szabó. A (simple) classical algorithm for estimating betti numbers. *arXiv:2211.09618*, 2022. URL https://arxiv.org/abs/2211.09618.

[131] Matthew Hagan and Nathan Wiebe. Composite quantum simulations, 2022.

[132] Yu Tong, Victor V. Albert, Jarrod R. McClean, John Preskill, and Yuan Su. Provably accurate simulation of gauge theories and bosonic systems, 2021.

[133] Alexander F. Shaw, Pavel Lougovski, Jesse R. Stryker, and Nathan Wiebe. Quantum algorithms for simulating the lattice schwinger model. *Quantum*, 4:306, Aug 2020. ISSN 2521-327X. doi:10.22331/q-2020-08-10-306. URL http://dx.doi.org/10.22331/q-2020-08-10-306.

[134] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A*, 86(5), Nov 2012. ISSN 1094-1622. doi:10.1103/physreva.86.052329. URL http://dx.doi.org/10.1103/PhysRevA.86.052329.

[135] Philippe Faist, Sepehr Nezami, Victor V. Albert, Grant Salton, Fernando Pastawski, Patrick Hayden, and John Preskill. Continuous symmetries and approximate quantum error correction. *Physical Review X*, 10(4), Oct 2020. ISSN 2160-3308. doi:10.1103/physrevx.10.041018. URL http://dx.doi.org/10.1103/PhysRevX.10.041018.

[136] Zohreh Davoudi, Indrakshi Raychowdhury, and Andrew Shaw. Search for efficient formulations for hamiltonian simulation of non-abelian lattice gauge theories. *Phys. Rev. D*, 104:074505, Oct 2021. doi:10.1103/PhysRevD.104.074505. URL https://link.aps.org/doi/10.1103/PhysRevD.104.074505.

[137] Simon Apers, Sayantan Sen, and Dániel Szabó. A (simple) classical algorithm for estimating betti numbers. *arXiv:2211.09618*, 2022.

[138] E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81:5672–5675, Dec 1998. doi:10.1103/PhysRevLett.81.5672. URL https://link.aps.org/doi/10.1103/PhysRevLett.81.5672.

Appendix A
DIAMOND NORM

The diamond distance is often used as a measure of error between two quantum channels. It is defined as follows:

$$d_\diamond(\mathcal{E},\mathcal{N}) = \frac{1}{2}||\mathcal{E}-\mathcal{N}||_\diamond, \tag{A.1}$$

where $||\dots||_\diamond$ is the diamond norm

$$||\mathcal{P}||_\diamond \coloneqq \sup_{\rho;||\rho||_1=1}||(\mathcal{P}\otimes I)(\rho)||_1 \tag{A.2}$$

and $\mathcal{E}$ and $\mathcal{N}$ are two quantum channels or superoperators. Note that I acts on the same size Hilbert space as $\mathcal{P}$ and ρ is a density matrix. All operators here are expressed as square matrices and ρ can represent states entangled with qubits that are not operated on. We then need an identity matrix to "pad out" the missing dimensions so that $\mathcal{P}\otimes I$ can act sensibly upon ρ.

The diamond norm is simply the trace distance but maximized over all possible input states and satisfies two key properties:

(1) Triangle inequality: $||\mathcal{A}+\mathcal{B}||_\diamond \leq ||A||_\diamond + ||\mathcal{B}||_\diamond$

(2) Sub-multiplicativity: $||\mathcal{A}\mathcal{B}||_\diamond \leq ||\mathcal{A}||_\diamond ||\mathcal{B}||_\diamond$

It follows from the definition of the diamond norm that if we apply the channel $\mathcal{E}$ and $\mathcal{N}$ to the quantum state σ, we have

$$d_{tr}(\mathcal{E}(\sigma),\mathcal{N}(\sigma)) = \frac{1}{2}||\mathcal{E}(\sigma)-\mathcal{N}(\sigma)||_1 \leq d_\diamond(\mathcal{E},\mathcal{N}). \tag{A.3}$$

The trace norm distance is important since it bounds the error in expectation values. To see this, consider the expression $|\mathrm{Tr}(M\mathcal{E}(\sigma)) - \mathrm{Tr}(M\mathcal{N}(\sigma))|$. The expectation value of an

operator M with respect to a state ρ can be found by taking the trace of their product, i.e $\mathrm{Tr}(M\rho)$. Thus, in the above expression, we first send a state σ through our two channels. Then we find the expectation value of M with respect to their outputs and take the absolute value of the difference to find the error in expectation values.

We can bound this error in expectation values by the following inequalities

$$\begin{aligned} |\mathrm{Tr}(M\mathcal{E}(\sigma)) - \mathrm{Tr}(M\mathcal{N}(\sigma))| &= |\mathrm{Tr}[M(\mathcal{E}(\sigma) - \mathcal{N}(\sigma))]| \leq 2||M||d_{tr}(\mathcal{E}(\sigma), \mathcal{N}(\sigma)) \\ &\leq 2||M||d_{\diamond}(\mathcal{E}, \mathcal{N}). \end{aligned} \tag{A.4}$$

In the first inequality, the von-Neumann trace inequality

$$|\mathrm{Tr}(AB)| \leq \sum_{i=1}^{n} \alpha_i \beta_i$$

was used where α_i, β_i are the singular values of the operators A and B respectively. This inequality can be further bounded by recognizing that $\alpha_i \leq \alpha_{\max}$ for all i, where $\alpha_{\max}$ is the largest singular value of A. The largest singular value of A is precisely $||A||_\infty$ so

$$|\mathrm{Tr}(AB)| \leq \sum_{i=1}^{n} \alpha_i \beta_i \leq \sum_{i=1}^{n} \alpha_{\max} \beta_i = ||A||_\infty \, ||B||_1.$$

The second inequality in the above expression follows directly from the definition of the diamond norm.

Now note that if we have a projection operator P, $P^\dagger P = P$ since projection operators are Hermitian and square to themselves. Their eigenvalues are 1 and 0 so it immediately follows that $||P||_\infty = 1$. So if M is a projection operator and we have ε error in the diamond distance, then

$$|\mathrm{Tr}(M\mathcal{E}(\sigma)) - \mathrm{Tr}(M\mathcal{N}(\sigma))| \leq 2\varepsilon.$$

This justifies the statement that measurement statistics are correct up to a factor of 2ε with an ε error in diamond distance.

Appendix B

NOTATION FOR QDRIFT

We establish the following notational conventions from Ref. [29] for describing the time-dependent qDRIFT scaling. Let $\alpha \in \mathbb{C}^L$ be a vector. The notation $||\alpha||_p$ represents the l_p norm of α and we define a few cases as follows:

$$||\alpha||_1 \coloneqq \sum_{j=1}^{L} |\alpha_j|, \quad ||\alpha||_2 \coloneqq \sqrt{\sum_{j=1}^{L} |\alpha_j|^2}, \quad ||\alpha||_\infty \coloneqq \max_{j\in\{1,2,\ldots,L\}} |\alpha_j| \ .$$

If A is a matrix, $||A||_p$ denotes the Schatten-p norm of A. A few important examples are:

$$||A||_1 \coloneqq \mathrm{Tr}(\sqrt{A^\dagger A}), \quad ||A||_2 \coloneqq \sqrt{\mathrm{Tr}(A^\dagger A)}, \quad ||A||_\infty \coloneqq \max_{|\psi\rangle} ||A|\psi\rangle||_2 \ .$$

If $f : [0,t] \to \mathbb{C}$ is a continuous function, $||f||_p$ denotes the L^p norm of the function. Thus,

$$||f||_1 \coloneqq \int_0^t d\tau |f(\tau)|, \quad ||f||_2 \coloneqq \sqrt{\int_0^t d\tau |f(\tau)|^2}, \quad ||f||_\infty \coloneqq \max_{\tau\in[0,t]} |f(\tau)| \ .$$

These norms can be combined to obtain vector and operator-valued functions. Suppose $\alpha : [0,t] \to \mathbb{C}^L$ is a continuous vector-valued function with components at time τ denoted by $\alpha_j(\tau)$. $||\alpha||_{p,q}$ denotes taking the l_p norm $||\alpha(\tau)||_p$ for all τ and computing the L^q norm of the resulting scalar function, e.g.

$$||\alpha||_{1,1} = \int_0^t d\tau \sum_{j=1}^{L} |\alpha_j| \ .$$

Similar reasoning applies when dealing with the Schatten p-norm of a time-dependent operator and then applying an L^q norm to the resulting scalar function, i.e. $||A||_{p,q}$. For example,

$$||A||_{1,2} = \sqrt{\int_0^t d\tau \ (\mathrm{Tr}\big(\sqrt{A^\dagger A}\big))^2} \ .$$

Note that this notation, while compact, is not well suited for describing evolution within a sub-interval of the entire evolution. In the event that a shorter evolution needs to be described, we explicitly use the integral expression over the domain in question.

For time-dependent linear combinations $A(\tau) = \sum_{l=1}^{L} A_l(\tau)$, the notation $||A||_{p,q,r}$ means taking the Schatten p-norm $||A_l(\tau)||_p$ of each term in the sum and applying the l_q and L^r norms to the resulting vector-valued functions, e.g.

$$\|A\|_{1,1,\infty} \coloneqq \max_{\tau\in[0,t]} \sum_{l=1}^{L} \|A_l(\tau)\|_1 \ .$$

Appendix C
COMPLEXITY CLASSES

We provide non-rigorous definitions here of some terminology from complexity theory and of a few quantum and classical complexity classes that are referenced throughout this thesis.

- P: class of decision problems (computational problems with a 'yes' or 'no' answer) that can be solved in polynomial time on a Turing machine
- NP: class of decision problems such that a 'yes' answer can be *verified* in P
- PSPACE: class of decision problems that can be solved by a Turing machine using polynomial amount of space (but not necessarily a polynomial amount of time)
- BPP: class of problems that can be solved in polynomial time if a bounded probability of error (typically 1/3) is allowed in the solution

On the quantum complexity side, we have the following classes:

- BQP: class of decision problems solvable in polynomial time by a quantum Turing machine with at most 1/3 probability of error. This is the quantum counterpart of the class BPP
- QMA: class of decision problems whose solution can be verified in polynomial time by a quantum Turing machine with high probability (typically 2/3). This is the quantum counterpart of the class NP. The class QMA_1 is similar, with the exception that the yes-instances of the problem are accepted with probability 1.

- DQC1: class of problems solvable by a BQP computer where a single qubit is initialized in the '0' state and the remaining qubits are initialized in the maximally mixed state

A problem is *hard* for a given complexity class if it is at least as computationally hard as all the problems in that class. It is *complete* if the problem also resides in that class; more specifically, an algorithm for solving a complete problem can be adapted to solve any other problem in that class. Thus for example, an NP-hard problem need not reside in NP but an NP-complete problem does.

An important result concerning the DQC1 class is that one can efficiently estimate the trace of an arbitrary unitary operator in this model via the Hadamard test [138]. Since trace estimation of a unitary is believed to be a hard problem for classical computers [138], this implies that a DQC1-hard problem cannot be "dequantized", or be solved via an efficient classical algorithm in BPP unless $\mathsf{DQC1} \subseteq \mathsf{BPP}$. This containment is widely believed to be false.