

Practical design and demonstration of algorithms for quantum devices

by

Annie Ray

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics (Quantum Information)

Waterloo, Ontario, Canada, 2024

© Annie Ray 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Simon Benjamin
Professor, University of Oxford,
Dept. of Materials

Supervisor(s): Raymond Laflamme
Professor, University of Waterloo,
Dept. of Physics and Astronomy

Internal Member: Roger Melko
Professor, University of Waterloo,
Dept. of Physics and Astronomy

Internal-External Member: Graeme Smith
Associate Professor, University of Waterloo,
Dept. of Applied Mathematics

Other Member(s): David Gosset
Associate Professor, University of Waterloo,
Dept. of Combinatorics and Optimization

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The contents of this thesis are based on research undertaken by Annie Ray in collaboration with past and present members of the Institute for Quantum Computing, Perimeter Institute for Theoretical Physics, and with researchers at Phasecraft Ltd.. Listed below are Annie Ray’s original contributions to the content matter in each chapter of the thesis.

1. Chapter 1: **Introduction**

- This chapter is written entirely by Annie Ray.
- This chapter presents a general overview of the field of quantum computation, some mathematical preliminaries of quantum information, and a brief summary of the results presented in this thesis.
- Any information presented in this chapter is either common knowledge or attributed to its source, though some new notational conventions are presented.

2. Chapter 2: **Sweep rule: A cellular automaton decoder for topological codes**

- The writing and results presented in this chapter are modified from (and overlap with) the published article [1], of which Annie Ray was first author.
- The aforementioned article was co-authored by Aleksander Kubica, and Raymond Laflamme.
- Annie Ray’s contributions to the research presented in this chapter and in the aforementioned article include development of code for numerical simulations, numerical analysis, and writing.

3. Chapter 3: **Fault-tolerant logical operations in QECC**

- The writing and results presented in this chapter are modified from (and overlap with) the published article [2], of which Annie Ray was second author.
- The aforementioned article was co-authored by Daniel Honciuc Menendez (first author), and Michael Vasmer.
- Annie Ray’s contributions to the research presented in this chapter and in the aforementioned article include design of experiments demonstrating fault-tolerant logical operations with the $[[8, 3, 2]]$ code, and writing.

4. Chapter 4: **An algorithm for imaginary time evolution of quantum systems**

- The writing and results presented in Sec. 4.1 are based on ongoing research (preprint in progress) led by Annie Ray in collaboration with Ningping Cao, Anirban Chowdhury, Michael Vasmer, and Esha Swaroop.
- Annie Ray’s contributions to the material presented in Sec. 4.1 include development of code for numerical simulations, analysis of sampling complexity of proposed algorithms, implementation of experiments on IBM’s superconducting devices, numerical analysis, and writing.

Abstract

The emergence of noisy intermediate-scale quantum (NISQ) devices represents a significant milestone in the journey towards the development of large-scale fault tolerant quantum computers. These devices have not only opened avenues for demonstrating a quantum advantage but have also advanced the practical development of quantum algorithms for solving challenging problems in physics, chemistry, and computer science. Most notably, this progress has necessitated a tailored approach to algorithm development that considers the specific architecture and hardware constraints of these quantum devices in order to effectively use them. However, the most useful instances of problems that we hope to solve with quantum computers require significant hardware improvements over the state-of-the-art, including at least a hundred fold increase in the number of qubits. The transition from intermediate-scale to large-scale quantum computers also presents other formidable challenges, particularly for engineering precise quantum control at scale. This thesis attempts to narrow the gap between intermediate and large-scale devices by proposing methods to mitigate noise effects on NISQ devices and by enhancing standard quantum algorithms to minimize resource overhead. One focus is on error correction strategies capable of managing noise on quantum devices. Specifically, we demonstrate the robustness of the sweep rule (a decoder for topological quantum codes) against measurement errors in quantum codes. Additionally, we experimentally demonstrate the improvement in performance of entangling non-Clifford operations when encoded in the $[[8,3,2]]$ code, strengthening the case for error correction. Furthermore, we improve a well-known technique known as imaginary time evolution to reduce the associated qubit and entangling gate overhead, making it more amenable to implementation on NISQ devices. By exploring these avenues, we aim to strike a balance, leveraging NISQ devices to expand their computational capabilities in the short term while serving as a sandbox for the development of future large-scale fault-tolerant quantum computers.

Acknowledgements

Over the course of this PhD, I have met several wonderful people from whom I have learnt a lot. This journey would not have been possible without their support and I would like to express my gratitude here.

Foremost, I would like to thank my supervisor Ray Laflamme, especially for giving me the opportunity to be a visiting student at IQC in 2018, and for taking me on as a PhD student in 2019. In the time since, he has opened many doors that have led to exciting opportunities to pursue research for which I am especially indebted to him. I'm particularly grateful to Ray for teaching me very early on the importance of effectively communicating scientific ideas and for always urging me to think about the bigger picture and how my research fits into it. These valuable skills have helped me think deeply about the problems I worked on during my PhD.

I would like to thank Alex Kubica for mentoring me on the first research project of my PhD during his time at PI. From him, I learnt how to approach challenging new problems from first principles. I am grateful to him for teaching me how to rigorously and diligently work through each of the small details that together make up the scientific method. Thanks to Alex, I also learnt the importance of brevity and clarity in scientific writing.

I'm grateful to the postdocs in our group and other colleagues that I have had the pleasure of collaborating with during my PhD. In particular, I would like to thank Mike Vasmer for being an encouraging mentor and a patient, kind, resourceful, and hospitable colleague. I am also grateful to him for initiating our research collaboration with UCL and Phasecraft. I would like to thank Anirban Chowdhury for lending his scientific expertise and clarity when I was studying Hamiltonian simulation in preparation for my Comprehensive Exam, and in our latest research project. I am grateful to both Anirban and Sayonee for the wonderful chats about life and our shared Indian roots! I would also like to thank Ningping Cao for being an inspiring and motivating role model to me in addition to being a kind and resourceful collaborator. I am also thankful to the members of Ray's group—past and present—for giving me a friendly environment in which to ask my numerous questions about everything quantum. Thanks are also due to Joel Klassen, Charlie Derby, and the team at Phasecraft for our collaboration during my internship in Fall 2023.

I am grateful to my friends, old and new, for supporting me throughout my PhD. I would especially like to thank Harikrishnan Unnithan for his friendship since undergrad, for the many philosophical conversations had over gaming sessions, and for being my go-to person for discussing operating systems and code optimizations. I am also thankful to Achyudh Ram, for helping me during the first year of my stay in Canada and for his friendship throughout my PhD. I will fondly remember the time spent playing Civ,

talking about philosophy and religion, and geeking out on classical music and composers. I would like to thank Emma Dunbar for her friendship during my PhD, especially in the Covid years, and for being an unexpected but sure source of motivation as I watched her gracefully navigate the many challenges of life as a musician through prayer, hard work, and sheer determination. I am grateful to Natalie Parham for introducing me to powerlifting in the 3rd year of my PhD; it transformed my relationship with exercise and helped me prioritize my health. Special thanks are due to Stefanie Beale, for her friendship, for helping me navigate the challenges of grad student life, for the musicals, the dancing, the sleepovers, the late night chats in the parking lot, and the adventures in Hawaii. Thanks are also due to my flatmates Yishu Zhang and Michelle Chien for their sisterly company and for making our apartment a peaceful home to live in. I am grateful to Tim Min for encouraging me to get back to playing piano, and for teaching me to recontextualize personal and intellectual challenges as new stories waiting to be written. I am thankful for the many friends I made at the 2022 IBM QEC summer school in New York, especially Arthur Pesah and Aurélie Denys, with whom I have enjoyed many conversations about research and life. I would like to thank George Umbrascu, for always engaging with and fueling my curiosity, for our consistently riveting discussions about life and work, for being an excellent badminton/snooker/poker partner, for our adventures in London and Cardiff, for keeping me motivated during the last months of thesis writing, and for proofreading this thesis.

I am immensely grateful to my family away from home at the First Baptist Church of Waterloo for their love, friendship, and godly encouragement. I would like to thank the College & Careers group at FBC, especially Abigail, Alysse, Daniel, Elora, Emily, Emma, Isaac, Jasson, Joseph, Justine, Oswin, Rachel, Tia, Tim Tedford, and Tim Min for encouraging me in all my endeavors with their prayers and with many-a-conversation had over Bible study, dinners, sleepovers, and impromptu bubble teas. I am also grateful to Jenn and Kevin Zehr for hosting wonderful feasts in their home, for the student community in our church. Their hospitality during my PhD has cushioned the blow of being so far away from my family in India. I am thankful to the musicians of the Kitchener-Waterloo Symphony for coloring my life in Waterloo with glorious music.

Finally, I would like to thank my parents, my champions, my role models—Sherly and Debabrata Ray. I am forever indebted to them for their constant care, support, and encouragement. I am blessed to have their unconditional love, which has seen me through every challenge in life.

Dedication

solī deo gloria
(Glory to God alone)

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Statement of Contributions	iv
Abstract	vi
Acknowledgements	vii
Dedication	ix
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Preliminaries of quantum information	3
1.2 Noise characterization of quantum systems	7
1.2.1 Quantum process tomography	9
1.2.2 Randomized benchmarking	10
1.3 Error correcting codes	13
1.3.1 The stabilizer formalism	13

1.3.2	Shor’s 9-qubit code	16
1.3.3	Topological quantum codes: The 2D Toric code reviewed	17
1.4	Fault-tolerant error correction	20
1.5	Error mitigation protocols	23
1.5.1	Probabilistic error cancellation	24
2	Sweep rule: A cellular automaton decoder for topological codes	27
2.1	The 3D Toric code	27
2.2	Error correction with sweep rule	30
2.2.1	Simulating the sweep rule	33
2.3	Numerical analysis of memory lifetime of sweep rule	36
2.4	Statistical-mechanical analysis of memory phase transition of sweep rule . .	37
2.5	Analytical model of sweep rule	40
2.6	Discussion	42
3	Fault-tolerant logical operations in QECC	44
3.1	The $[[8,3,2]]$ color code	45
3.2	Fault-tolerant circuits	46
3.2.1	GHZ state preparation	47
3.2.2	$ +++ \rangle$ state preparation	47
3.3	Demonstration results	49
3.4	Discussion	52
4	An algorithm for imaginary time evolution of quantum systems	55
4.1	Imaginary time evolution (ITE)	55
4.2	Quasiprobability decompositions of linear maps	57
4.2.1	An algorithm for estimating expectation values for states evolving under non-trace-preserving maps	58
4.3	Estimating expectation values for ITE states via QPD	61
4.3.1	Estimating thermal state properties via QPD	66
4.4	Discussion	68

5	Conclusion	70
	References	72
	APPENDICES	83
A	Simulation of sweep rule	84
A.1	Details of numerical analysis of memory lifetime	84
A.1.1	Obtaining data	84
A.1.2	Identifying phase transitions	85
A.1.3	Analysis of the sweep rule on triangular lattices	85
A.1.4	Analysis of Toom’s rule on square lattices	86
A.2	Details of statistical-mechanical analysis	86
A.3	Critical exponents of variants of the sweep rule	90
B	Additional details of the demonstration of $[[8,3,2]]$code	93
B.1	Post-selection rates	93
B.2	Additional hardware details	93
C	Sampling complexity of quasiprobabilistic algorithms	103
C.1	Proof of Lemma 1	103
C.2	Proof of Lemma 2	105
C.3	Proof of Lemma 3	106

List of Figures

1.1	Shor's 9-qubit code	17
1.2	Stabilizers of the 2D Toric code	18
1.3	Logical operators of the 2D Toric code	19
1.4	Errors as excitations in the 2D Toric code	21
1.5	Transversal CNOT gate in the Steane code	23
1.6	Schematic diagram of PEC	25
2.1	Stabilizers of the 3D Toric code	28
2.2	Errors as excitations in the 3D Toric code	29
2.3	Sweep rule update	30
2.4	Schematic model of classical memory	31
2.5	Application of deterministic rule	32
2.6	Evolution of magnetization in sweep rule	34
2.7	Evolution of smoothened magnetization of sweep rule	34
2.8	Snapshots of spin configurations of sweep rule	36
2.9	Memory lifetime of sweep rule	36
2.10	Phase diagram of the sweep rule	38
2.11	Critical behavior of sweep rule at $h = 0$	39
2.12	Analytical model of memory lifetime of sweep rule I	41
2.13	Analytical model of memory lifetime of sweep rule II	41
3.1	Geometric representation of the $((8,3,2))$ code	46

3.2	Fault-tolerant $ \text{GHZ}\rangle$ state preparation in the $((8,3,2))$ code	48
3.3	Fault-tolerant $ +++ \rangle$ state preparation in the $((8,3,2))$ code	49
3.4	Performance of encoded GHZ state prep with X basis measurement	50
3.5	Performance of encoded $ +++ \rangle$ state prep with X basis measurement	51
3.6	Performance of encoded $ +++ \rangle$ state prep with Z basis measurement	52
3.7	Performance of encoded GHZ state prep with Z basis measurement	53
4.1	Estimation of energy of the 2-qubit Heisenberg model using ITE	62
4.2	Scaling of γ with inverse temperature	64
4.3	A schematic of trotterized imaginary time evolution	65
4.4	QPD for TPQ states	67
A.1	Estimating parameters a, b, c of fitting ansatz	86
A.2	Plots of memory lifetime $\tau_{L,T}$ of the sweep rule as a function of linear lattice size L	87
A.3	Plot of parameter b as a function of temperature T (sweep rule)	88
A.4	Phase diagram of the Toom's rule in the (T, h) plane	88
A.5	Plots of parameter b as a function of temperature T (Toom's rule)	89
A.6	Displacement vectors of the honeycomb lattice	89
A.7	Critical behavior of the asynchronous sweep rule at $h = 0$	90
A.8	Critical behavior of synchronous Toom's rule at $h = 0$	91
A.9	Critical behavior of asynchronous Toom's rule at $h = 0$	92
B.1	Layout of the <code>ibm_mumbai</code> device	98

List of Tables

4.1	Single qubit QPD basis (Endo)	59
4.2	Universal basis for QPD of $2q$ CPTP maps (Takagi basis)	63
A.1	Critical exponents of the sweep rule	92
B.1	Post-selection rates for the results of Figs. 3.5, 3.6	94
B.2	Post-selection rates for the results of Figs. 3.4, 3.7 on IonQ device	95
B.3	Post-selection rates for the results shown in Figs. 3.4, 3.7 on IBM device	96
B.4	Characterization data for <code>ionq-11q</code>	97
B.5	1-qubit gate characterization data for <code>ibm_mumbai</code> on 14/04/23.	99
B.6	2-qubit gate characterization data for <code>ibm_mumbai</code> on 14/04/23.	101
B.7	Qubit characterization data for <code>ibm_mumbai</code> on 14/04/23.	102

Chapter 1

Introduction

The notion of quantum computation was first proposed by Richard Feynman [3] for simulating quantum many-body physics as a natural and potentially more powerful analog of classical computation. This idea was concretized in Seth Lloyd’s seminal paper [4] on simulating time evolution of quantum Hamiltonians via Trotterization. In 1994, Peter Shor also proposed a quantum algorithm for integer factorization [5] that provides a superpolynomial speedup over its classical counterpart. Together, these ideas led to the development of quantum computing as a theoretical framework for solving difficult computational tasks by exploiting the laws of quantum mechanics. The quest to realize a potential quantum advantage for computation led to the development of quantum devices which implemented qubits—the quantum analog of bits—using atoms, molecules, trapped ions, superconductors, as well as photonic systems. These advances also revealed the many challenges that lay in the way of developing a practical large-scale quantum computer. In particular, the sensitivity of the atoms and molecules (used for building quantum devices) to noise from their environment was a serious stumbling block. Noise in quantum systems threatened to destroy important resources such as quantum entanglement and (coherent) superpositions.

The late 1990s saw many developments that presented ways to circumvent noise in quantum systems. Most notably, constructions of error correcting codes [6–8] showed that it is possible to suppress decoherence and noise in quantum circuits using encoding schemes similar to those used in classical error correction. These early constructions of error correcting codes led to the formalization of the principles of quantum error correction and fault-tolerance. The threshold theorem [9–11] for fault-tolerant quantum computation established that under reasonable assumptions about noise in quantum hardware, it is possible to use a collection of noisy (physical) qubits to encode a logical qubit whose error rate can be made arbitrarily small, with the physical qubit overhead only scaling polynomially

with the number of logical qubits encoded. Although this presented a positive outlook for the future of quantum computing, one major drawback of fault-tolerant error correction schemes is that the estimated number of qubits required to reduce the logical error rate to levels that are desirable for useful quantum computation is still beyond the reach of state-of-the-art quantum computers. Moreover, the high error rates in the best available quantum devices also suggest that there remain significant hurdles in the development of precise quantum control of large collections of qubits, particularly in the implementation of multi-qubit entangling operations.

Meanwhile there have also been significant developments in other approaches to managing noise in quantum systems. Most notably, the field of error mitigation is aimed at extracting useful information from noisy quantum computation without using the resource-intensive machinery of quantum error correction. Error mitigation strategies are a hybrid quantum-classical approach where several instances of noisy quantum computation can be used to emulate the noiseless computation using classical post-processing. This approach is best suited for the current generation of quantum devices, which have lower error rates than ever before, but for which error correction schemes still lie beyond reach. In literature, error-mitigation strategies have quite often been treated as merely a stepping stone (useful only in the short term) on the path towards large-scale fault-tolerant error correction. While in the long term we expect large-scale fault-tolerant devices to have reduced logical error rates (compared to physical error rates), the logical error rates will still be non-zero owing to the finite size of the device. Thus, even in the long term, error mitigation strategies will serve as an important extra step that allows noisy quantum computation to be refined with classical post-processing. This fits into the paradigm of hybrid quantum-classical or quantum-accelerated computation that has long been foreshadowed in blueprints for large-scale quantum architectures.

In this thesis, I will discuss my research that contributes to, and is inspired by strategies in error correction and error mitigation. In the remainder of chapter 1, I will first introduce some fundamentals of quantum information, and a few useful models of noise in quantum devices. Then I will discuss some strategies for error correction and error mitigation that are relevant for the results presented in this thesis. In chapter 2, I will present our research on error correction with the sweep rule—a cellular automaton decoder for topological quantum error correcting codes. We numerically and analytically show that the sweep rule has a phase transition under biased bit-flip noise which indicates that it is robust to quantum (syndrome) measurement errors when used as a decoder for topological quantum codes. In chapter 3, I will present our research on the implementation of logical operations in the $[[8, 3, 2]]$ quantum code on superconducting and trapped ion devices. We demonstrate a fault-tolerant implementation of the $[[8, 3, 2]]$ code’s non-Clifford logical gates on

quantum hardware and show that the encoded gates have an improved performance over the unencoded version. In chapter 4, I discuss a quasiprobabilistic algorithm for quantum simulation, particularly for implementing imaginary time evolution of Hamiltonians. We develop a new method to implement imaginary time evolution of quantum Hamiltonians (such as the 1D Heisenberg model) and estimate their thermal properties. Our algorithm is inspired by an error mitigation strategy, making it inherently noise-resilient, as demonstrated via experiments on IBM’s superconducting device. Finally in chapter 5, I summarize my research contributions and discuss the future scope and impact of my work for the development and use of the next generation of quantum devices.

1.1 Preliminaries of quantum information

A qubit is the basic unit of quantum information. It is the quantum analog of a classical bit and is typically represented by a two-dimensional vector in the complex Hilbert space $\mathcal{H} = \mathbb{C}^2$. The computational basis elements of this Hilbert space are (conventionally) chosen to be

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1.1)$$

Quantum states and their representations

A pure quantum state can be expressed as a normalized vector $|\psi\rangle \in \mathcal{H}$, where

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1. \quad (1.2)$$

The quantities $|\alpha|^2, |\beta|^2$ are called the probability amplitudes of the state and denote the probability of the outcomes 0 and 1 when a quantum measurement (in the computational basis) is performed on the quantum state. A quantum state $|\psi\rangle$ can also be represented by its density (matrix) operator¹ σ , which is the outer product of $|\psi\rangle$ with itself,

$$\sigma = |\psi\rangle \langle \psi|. \quad (1.3)$$

A statistical mixture (ensemble) of pure quantum states is called a mixed state and can be denoted by its density matrix ρ , where

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|, \quad \sum_i p_i = 1, \quad (1.4)$$

¹In literature, the terms density operator and density matrix are often used interchangeably.

and $\{p_i\}$ is a probability distribution. As a consequence of Eqs. 1.2 and 1.4, the trace of the density matrix of a quantum state is 1.

Quantum channels and their representations

A quantum system comprising qubits is assumed to evolve under quantum channels that are modelled as completely positive trace-preserving (CPTP) maps which map one quantum state to another. The action of a quantum channel $\mathcal{E}$ on an n -qubit input state ρ (in the $d = 2^n$ Hilbert space) is given by—

$$\mathcal{E}(\rho) = \sum_{i=1}^N K_i \rho K_i^\dagger, \quad N \leq d^2, \quad (1.5)$$

where the $\{K_i\}$ are known as the Kraus operators of the channel and satisfy the *completeness relation*

$$\sum_i K_i^\dagger K_i = \mathbb{1} \quad (1.6)$$

The Kraus operators in Eq. 1.5 are not unique and the Kraus representation depends on the choice of basis.

A quantum state is said to undergo *unitary evolution* if the quantum channel has a single Kraus operator given by some unitary operator U . Unitary operators acting on quantum states are usually referred to as *quantum gates*. When a unitary gate U is applied to an initial (pure) quantum state $|\psi\rangle$, the final state $|\phi\rangle$ can be obtained by matrix multiplication of the operator and the state vector,

$$|\phi\rangle = U |\psi\rangle. \quad (1.7)$$

This same operation can be viewed as applying a (unitary) quantum channel to the state $\rho = |\psi\rangle \langle\psi|$ to obtain the final state $\sigma = |\phi\rangle \langle\phi|$ as shown below—

$$\sigma = |\phi\rangle \langle\phi| = U \rho U^\dagger. \quad (1.8)$$

The notation of quantum channels (as in Eqs. 1.5, 1.8) is a convenient way of expressing the action of quantum operations on general quantum states—pure or mixed.

The Pauli operators X , Y , and Z are examples of single-qubit unitary operators that can act on quantum states and are given by—

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1.9)$$

The Pauli operators are themselves Hermitian, and particularly useful because together with the single-qubit Identity operator $\mathbb{1}$, they form a basis $\mathcal{P} = \{\mathbb{1}, X, Y, Z\}$ for the real vector space over 2×2 Hermitian matrices. Thus, any 2×2 Hermitian matrix ρ can be expressed as a linear combination (with real coefficients) of the elements of $\mathcal{P}$ —

$$\rho = \frac{1}{2}(\mathbb{1} + r_x X + r_y Y + r_z Z), \quad (1.10)$$

where $\vec{r} = (r_x, r_y, r_z) \in \mathbb{R}^3$. When ρ is the density matrix of a quantum state, $\vec{r}$ is called its *Bloch vector* and $|\vec{r}| = 1$. The set $\mathcal{P}$ is also a basis for all 2×2 complex-valued matrices.

For a general quantum channel given by Eq. 1.5, the n -qubit Pauli basis $\mathcal{P}^{\otimes n}$ is a conventional choice of basis for the Kraus operators—

$$K_i = \sum_{j=1}^{d^2} a_{ij} P_j, \quad P_j \in \mathcal{P}^{\otimes n}. \quad (1.11)$$

Thus, a general quantum channel $\mathcal{E}$ can be represented in the n -qubit Pauli basis as

$$\mathcal{E}(\rho) = \sum_{j,k=1}^{d^2} \Gamma_{jk} P_j \rho P_k \quad (1.12)$$

where Γ_{jk} is known as the *process matrix* of the channel. It is a $d^2 \times d^2$ complex-valued, Hermitian, positive semi-definite matrix that parameterizes the channel [12]. Another important representation of a channel is its *Pauli transfer matrix (PTM)* [12]. It captures the effect of the channel on each element of the Pauli basis $\mathcal{P}^{\otimes n}$ as represented in the same basis. The PTM $\Lambda_{ij}^{\mathcal{E}}$ of a channel $\mathcal{E}$ is a $d^2 \times d^2$ real-valued matrix given by—

$$\Lambda_{ij}^{\mathcal{E}} = \frac{1}{d} \text{Tr}[P_i \mathcal{E}(P_j)]. \quad (1.13)$$

The Pauli transfer matrix (PTM) is a useful representation because the composition of two channels can be evaluated by taking the matrix product of their respective PTMs. The trace-preserving nature of quantum channels imposes d^2 constraints on both the process matrix and the PTM to ensure that a general quantum channel can be expressed using $d^4 - d^2$ free parameters.

As per Eq. 1.7, the quantum state vector (of dimension $d = 2^n$) resulting from the action of an operator U ($d \times d$) can be obtained from direct matrix multiplication. Since the conventional representation of a quantum channel does not permit a similar treatment for the density operator, the representation of the density operator ρ can be lifted from

a $d \times d$ matrix in the d -dimensional Hilbert space to a vector $|\rho\rangle\rangle$ in the d^2 -dimensional *Hilbert-Schmidt space* [12]. For example, a general single-qubit state ρ (see Eq. 1.10) may be expressed in the n -qubit Pauli basis² $\mathcal{P}^{\otimes n}$ as the Hilbert-Schmidt vector

$$|\rho\rangle\rangle = \begin{bmatrix} 1 \\ r_x \\ r_y \\ r_z \end{bmatrix}. \quad (1.14)$$

The Hilbert-Schmidt inner product of two vectors $|\rho\rangle\rangle$ and $|\sigma\rangle\rangle$ is given by

$$\langle\langle\rho|\sigma\rangle\rangle = \frac{1}{d}\text{Tr}[\rho^\dagger\sigma] \quad (1.15)$$

where ρ, σ on the RHS are density operators. This lifting of operators from the Hilbert space to the Hilbert-Schmidt space is known as the *superoperator formalism* [12]. In this picture, quantum operations (which map density operators to other density operators) are represented as $d^2 \times d^2$ matrices, and the state resulting from applying a quantum channel to a quantum state can be obtained by matrix multiplication.

Quantum measurements

Quantum measurements are another important type of quantum channels. In this thesis the term *measurement* will be used to refer specifically to *projective measurements*. Such measurements comprise Kraus operators that are projectors of the form $\Pi_m = |e_m\rangle\langle e_m|$, where $\{|e_m\rangle\}$ is an orthonormal eigenbasis of the observable $\mathcal{O}$ (having real eigenvalues $\{m\}$) that is being measured. Thus, for an observable $\mathcal{O}$ defined as

$$\mathcal{O} = \sum_m m |e_m\rangle\langle e_m| = \sum_m m \Pi_m, \quad (1.16)$$

the action of the measurement channel on the input state ρ is given by—

$$\mathcal{M}(\rho) = \sum_m \Pi_m \rho \Pi_m, \quad \sum_m \Pi_m = \mathbf{1}. \quad (1.17)$$

Although Eq. 1.17 appears to be a simple case of the general quantum channel in Eq. 1.5, the measurement channel is special because the action of each projector Π_m on the input

²For convenience, when considering representations in the n -qubit Pauli basis $\mathcal{P}^{\otimes n}$, the Pauli operators are rescaled from P_i to $P_i/\sqrt{2^n}$.

state is associated with a classical *measurement outcome* given by the real eigenvalue m of the observable [13]. The probability $\Pr[m]$ of occurrence of the measurement outcome m , and the resulting post-measurement quantum state σ_m are given by—

$$\Pr[m] = \text{Tr}[\Pi_m \rho], \quad \sigma_m = \frac{\Pi_m \rho \Pi_m}{\Pr[m]}. \quad (1.18)$$

The quantum state described by Eq. 1.17 is the mixed state corresponding to the statistical ensemble [14] of post-measurement quantum states resulting from measuring the observable $\mathcal{O}$. In practice, the expected value of measurement outcomes is a quantity of interest and is known as the expectation value $\langle \mathcal{O} \rangle_\rho$ of $\mathcal{O}$ with respect to the input state ρ . It is defined as

$$\langle \mathcal{O} \rangle_\rho = \sum_m m \Pr[m] \quad (1.19)$$

$$= \sum_m m \text{Tr}[\Pi_m \rho] \quad (1.20)$$

$$= \text{Tr} \left[\left(\sum_m m \Pi_m \right) \rho \right] \quad (1.21)$$

$$= \text{Tr}[\mathcal{O} \rho]. \quad (1.22)$$

While there are various perspectives from which to approach quantum computation, it is generally modelled as a *circuit*, i.e., a sequence of quantum gates and measurements applied to some initial quantum state. This is known as the *circuit model* of quantum computation and will be the approach used throughout this thesis.

1.2 Noise characterization of quantum systems

Information stored in a physical system is prone to errors due to imperfect control of the system's interactions with its surroundings. While both classical and quantum systems are affected by noise, the sensitive nature of the building blocks of quantum hardware makes the rate of errors in quantum systems several orders of magnitude higher than that in classical systems. The noise in quantum computers threatens to destroy resources such as quantum entanglement and interference that may provide a significant advantage over classical computers in performing difficult computational tasks. Understanding the noise in a quantum system is a crucial first step in the development of strategies for

protecting information from the noise. In this section, a few standard ways of modelling and characterizing the noise in quantum systems are briefly described.

In classical systems, where information is stored in classical bits, these errors belong to one of two main categories—*bit-flip* errors ($1 \leftrightarrow 0$) and bit *erasure* errors (loss of the bit). Analogously, in quantum systems too the qubits can incur bit-flip errors and erasure errors. Additionally, qubits are also prone to *phase-flip* errors which may occur in combination with bit-flip errors.

Examples of noise channels

Noise affecting quantum systems is typically modelled as a CPTP map acting on a quantum state. In this model, the bit-flip noise channel is given by

$$\mathcal{B}_p(\rho) = (1 - p)\rho + pX\rho X, \quad K_0 = \sqrt{1 - p}\mathbb{1}, \quad K_1 = \sqrt{p}X \quad (1.23)$$

where the parameter p is the probability of a bit-flip and K_0, K_1 are the Kraus operators of the channel. A quantum state affected by the bit-flip channel $\mathcal{B}_p$ is affected by the X operator with probability p and is unchanged with probability $(1 - p)$. Similarly the phase-flip channel is given by

$$\mathcal{Z}_p(\rho) = (1 - p)\rho + pZ\rho Z, \quad K_0 = \sqrt{1 - p}\mathbb{1}, \quad K_1 = \sqrt{p}Z \quad (1.24)$$

As mentioned previously, quantum states can also be affected by a combination of bit-flip and phase-flip noise, where with probability p , the state is affected by the Y operator.

The depolarizing noise channel is another important example of a quantum noise channel and is given by—

$$\mathcal{D}_p(\rho) = (1 - p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z) \quad (1.25)$$

where the parameter p is the probability of error. A quantum state ρ affected by the depolarizing channel $\mathcal{D}_p$ remains in its original state with probability $(1 - p)$ while it can be affected by any one of the Pauli operators with equal probability $p/3$. The depolarizing channel is a special case of the general Pauli channel which has the form—

$$\mathcal{E}_p(\rho) = (p_{\mathbb{1}})\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z, \quad p_X + p_Y + p_Z = 1 \quad (1.26)$$

The aforementioned noise channels are simple and useful models for noise in quantum systems. However, in reality, the true nature of the noise affecting quantum systems is often

much more complicated. One major complication is that depending on the hardware, the noise in quantum devices may be time-dependent and affected by previous operations run on the device (non-Markovian noise). A quantum system with several qubits can also be affected by more sophisticated forms of noise due to unwanted interactions between the qubits within the system (for example, cross-talk [15]). In general, an unknown noise channel is hard to characterize because it can require $\mathcal{O}(16^n)$ parameters (see Eqs. 1.12) to describe it completely. Nevertheless, there have been several useful methods developed to probe the nature of noise in a quantum device. These include quantum process tomography and randomized benchmarking, which are briefly introduced below.

1.2.1 Quantum process tomography

The goal of quantum process tomography (QPT) is to fully characterize an unknown quantum channel by identifying the d^4 parameters (see discussion in Sec. 1.1) in the PTM of the channel (see Eq. 1.13). It is useful, however, to first consider quantum state tomography which involves estimating the d^2 parameters that describe a state.

A general n -qubit quantum state may be described as a d^2 vector $|\rho\rangle\rangle$ in the Hilbert-Schmidt space (see discussion on superoperator formalism in Sec. 1.1), where $d = 2^n$ —

$$|\rho\rangle\rangle = \sum_i^{d^2} c_i |P_i\rangle\rangle, \quad P_i \in \mathcal{P}^{\otimes n}. \quad (1.27)$$

The coefficient c_i is given by

$$c_i = \text{Tr}[P_i \rho] \quad (1.28)$$

which is the expectation value $\langle P_i \rangle_\rho$ according to Eq. 1.19. Thus, c_i can be estimated by applying a measurement (see Sec. 1.1) ρ in the P_i basis, requiring d^2 measurements in all. Now, a general quantum channel $\mathcal{E}$ is given by its $d^2 \times d^2$ Pauli transfer matrix $\Lambda^\mathcal{E}$ as in Eq. 1.13. The elements of the PTM are given by³

$$\Lambda_{ij}^\mathcal{E} = \text{Tr}[P_i \mathcal{E}(P_j)]. \quad (1.29)$$

Each P_j may be expressed in its eigenbasis $\{|+\rangle\langle +|_j, |-\rangle\langle -|_j\}$ corresponding to eigenvalues $\{+1, -1\}$ as follows—

$$P_j = (+1)|+\rangle\langle +|_j + (-1)|-\rangle\langle -|_j \quad (1.30)$$

³Note that here the n -qubit Pauli operators have been rescaled from P_i to $P_i/\sqrt{2^n}$ for compact notation.

Thus, Eq. 1.29 can be rewritten as

$$\Lambda_{ij}^{\mathcal{E}} = \text{Tr}[P_i \mathcal{E}(|+\rangle \langle +|_j)] - \text{Tr}[P_i \mathcal{E}(|-\rangle \langle -|_j)]. \quad (1.31)$$

The key idea of quantum process tomography is to essentially apply quantum state tomography to each of the d^2 states $\mathcal{E}(|+\rangle \langle +|_j)$. As described earlier, each state tomography protocol requires d^2 measurements, resulting in d^4 measurements in all for the QPT protocol. QPT is clearly a resource intensive protocol that requires an exponentially growing number of measurements (16^n) in order to characterize an arbitrary n -qubit noise channel.

1.2.2 Randomized benchmarking

Another approach to characterizing an unknown noise channel is to understand its “average” effect on input states. Mathematically, the average fidelity $F_{\text{ave}}(\mathcal{E})$ of a channel $\mathcal{E}$ is defined [16] as

$$F_{\text{ave}}(\mathcal{E}) = \int \langle \psi | (\mathcal{E} |\psi\rangle \langle \psi|) | \psi \rangle d\psi, \quad (1.32)$$

where the integral over all input pure states $|\psi\rangle$ quantifies the average effect of the channel. The idea behind randomized benchmarking is to capture the average behavior of noise by reducing the exponentially growing number of parameters (with number of qubits) to a smaller (constant or polynomially scaling) set of parameters that can be more easily estimated. It was shown in [17] that twirling a quantum channel with elements of the Pauli group results in a Pauli channel, while twirling a channel with elements of the Clifford group results in a depolarizing channel due to the 1-design and 2-design properties of the Pauli and Clifford group respectively [18, 19]. The randomized benchmarking (RB) protocol essentially benchmarks an unknown noise channel by twirling with the Clifford (or Pauli) group and estimating the probabilities of error associated with the resulting depolarizing (or Pauli) channel [20].

For an n -qubit channel $\mathcal{N}$ described by Kraus operators A_i

$$\mathcal{N}(\rho) = \sum_i A_i \rho A_i^\dagger, \quad (1.33)$$

its Clifford twirl is defined as—

$$\mathcal{N}_{\text{Cliff}}(\rho) = \frac{1}{|\mathcal{C}_n|} \sum_{U \in \mathcal{C}_n} U^\dagger \mathcal{N}(U \rho U^\dagger) U \quad (1.34)$$

$$= \frac{1}{|\mathcal{C}_n|} \sum_{U \in \mathcal{C}_n} \sum_i U^\dagger A_i U(\rho) U^\dagger A_i^\dagger U, \quad (1.35)$$

where $\mathcal{C}_n$ is the n -qubit Clifford group. It can be shown [17] that the Clifford twirl of a channel yields a depolarizing channel $\mathcal{D}_p$ (see Eq. 1.25 in Sec. 1.2) where the probability of error parameter p is dependent on the Pauli decomposition of the Kraus operators of the channel. Similarly a Pauli twirl (where $U \in \mathcal{P}^{\otimes n}$) of the channel results in a Pauli channel whose probabilities of Pauli errors depend on the Kraus operators of the channel. Crucially, the average fidelity of the unique depolarizing channel $\mathcal{D}_p$ and the Pauli channel obtained from the Clifford and Pauli twirls respectively is equal to the average fidelity of the channel $\mathcal{N}$.

The simplest RB protocol implements an effective Clifford twirl with the use of a sequence of randomly sampled Clifford unitary gates. The protocol for a single-qubit noise channel can be explained using the schematic circuit in Eq. 1.36.

$$\rho \text{ --- } \boxed{V_1} \text{ --- } \boxed{V_2} \text{ --- } \boxed{V_3} \text{ --- } \dots \text{ --- } \boxed{V_m} \text{ --- } \boxed{V^{-1}} \text{ ---} \quad (1.36)$$

Here the $\{V_i\}$ are picked uniformly at random from the single qubit Clifford group $\mathcal{C}_1$. The last gate is chosen to be the inverse of the combination of all the gates before so that the ideal circuit implements the identity operations $\mathbb{1}$.

$$V^{-1} = (V_m V_{m-1} \dots V_2 V_1)^{-1} = V_1^{-1} V_2^{-1} \dots V_{m-1}^{-1} V_m^{-1}. \quad (1.37)$$

When the circuit in Eq. 1.36 is executed on noisy quantum hardware that is affected by a noise channel $\mathcal{N}$ given by Eq. 1.33, the noisy version $\tilde{V}_i$ of each ideal gate V_i is implemented. It is assumed that the noisy version of each gate is the composition of the noise channel $\mathcal{N}$ and the ideal channel $\mathcal{V}_i(\cdot) = V_i(\cdot)V_i^\dagger$ i.e.

$$\tilde{\mathcal{V}}_i = \mathcal{N} \circ \mathcal{V}_i. \quad (1.38)$$

For $m \geq 2$, the output of circuit 1.36 is

$$\rho_{out} = \tilde{\mathcal{V}}^{-1} \circ (\tilde{\mathcal{V}}_m \dots \circ \tilde{\mathcal{V}}_2 \circ \tilde{\mathcal{V}}_1)(\rho) \quad (1.39)$$

$$= \sum_{j_1 \dots j_{m+1}} \left(A_{j_{m+1}} V^{-1} A_{j_m} V_m \dots A_{j_2} V_2 A_{j_1} V_1 \rho V_1^\dagger A_{j_1}^\dagger V_2^\dagger A_{j_2}^\dagger \dots V_m^\dagger A_{j_m}^\dagger (V^{-1})^\dagger A_{j_{m+1}}^\dagger \right) \quad (1.40)$$

$$= \sum_{j_1 \dots j_{m+1}} A_{j_{m+1}} V^{-1} A_{j_m} V_m \dots A_{j_2} V_2 \underbrace{V_1 V_1^\dagger}_{\mathbb{1}} A_{j_1} V_1 \rho V_1^\dagger A_{j_1}^\dagger \underbrace{V_1 V_1^\dagger}_{\mathbb{1}} V_2^\dagger A_{j_2}^\dagger \dots V_m^\dagger A_{j_m}^\dagger (V^{-1})^\dagger A_{j_{m+1}}^\dagger \quad (1.41)$$

$$= \sum_{j_1 \dots j_{m+1}} A_{j_{m+1}} V^{-1} A_{j_m} V_m \dots A_{j_2} V_2 V_1 \underbrace{(V_1^\dagger A_{j_1} V_1 \rho V_1^\dagger A_{j_1}^\dagger V_1)}_{\mathcal{N}_{V_1}(\rho)} V_1^\dagger V_2^\dagger A_{j_2}^\dagger \dots V_m^\dagger A_{j_m}^\dagger (V^{-1})^\dagger A_{j_{m+1}}^\dagger \quad (1.42)$$

$$\vdots$$

$$= \mathcal{N}_{V_{m+1} \dots V_2 V_1} \circ \dots \circ \mathcal{N}_{V_3 V_2 V_1} \circ \mathcal{N}_{V_2 V_1} \circ \mathcal{N}_{V_1}(\rho) \quad (1.43)$$

$$= \underbrace{\mathcal{N}_{V'_{m+1}} \circ \mathcal{N}_{V'_{m-1}} \circ \dots \circ \mathcal{N}_{V'_3} \circ \mathcal{N}_{V'_2} \circ \mathcal{N}_{V'_1}(\rho)}_{\text{Composition of } m \text{ independent channels}}, \quad (1.44)$$

where $\mathcal{N}_U = \sum_i U^\dagger A_i U \rho U^\dagger A_i^\dagger U$. Here $V'_j = V_j V'_{j-1}$ for $j \geq 2$ and $V'_1 = V_1$. Since each of the $\{V_i\}$ are chosen independently at random, the $\{V'_i\}$ are also independent. Thus the output of the circuit is a composition of m independent channels $\mathcal{N}_{V'_i}$. For a given value of m , K random instances of the circuit 1.36 are executed on noisy quantum hardware and for each output state $\rho_{\text{out}}^{(j)}$, its fidelity with respect to the input state $F(\rho_{\text{out}}^{(j)}, \rho)$ is estimated. Taking the average of these output state fidelities over the K random instances of the circuit gives the average sequence fidelity

$$F_{\text{seq}}(m, \rho) = \frac{1}{K} \sum_{j=1}^K F(\rho_{\text{out}}^{(j)}, \rho). \quad (1.45)$$

which is expected to scale with the noise channel's associated depolarizing channel parameter p as

$$F_{\text{seq}}(m, \rho) = Ap^m + B, \quad (1.46)$$

where A, B represent the state preparation and measurement (SPAM) parameters of the noise [21]. By applying this procedure to obtain $F_{\text{seq}}(m, \rho)$ for multiple values of m , the effective depolarizing parameter p of the noise in the system can be estimated by numerical fitting techniques. In a simplified way, Eq. 1.46 exploits the idea that averaging Eq. 1.44 over different random sequences $\{V_i\}$ for a given sequence length m approximates the composition of m depolarizing channels $\mathcal{D}_p$ (which together form a depolarizing channel $\mathcal{D}_{p^m}$ with parameter p^m) applied to the input state ρ . The number of instances K and the depth m of the circuits required to estimate the average behaviour of the noise channel has been heuristically found to scale linearly [22] in the number of qubits n , making RB a more scalable approach to characterizing noise in devices.

On the one hand, the QPT protocol permits a complete characterization of some unknown noise channel at the expense of $\mathcal{O}(16^n)$ measurements (uses) of the noise channel, while the Clifford RB protocol performs an average characterization of the channel using

significantly fewer measurements. The number of measurements required to learn an unknown noise channel is proportional to the level of detail that is desired in understanding the effects of the noise channel. However, as the size of a quantum system increases, the scalability of the noise characterization protocol becomes a more pressing concern and methods such as QPT that scale exponentially with the system size become practically infeasible. In recent years, there have been significant developments [23] in methods for learning arbitrary noise channels where a middle ground can be achieved between QPT and RB.

1.3 Error correcting codes

The construction of quantum error correcting codes (QECC) was one of the first approaches to managing noise in quantum systems. Peter Shor’s 9-qubit error correcting code is a notable example. It protects a single logical qubit of information by encoding it in a system of 9 physical qubits and can correct any single qubit error that affects the system. The construction of Shor’s code paved the way for the development of QECC by overcoming two major challenges in quantum systems. Firstly, it showed that despite the fact that measurements destroy coherence, it is possible to perform operations that extract classical information from a quantum system to check whether an error has occurred. This is accomplished by using *stabilizer* operators which are chosen so that the system is always an eigenstate of the operator, with the eigenvalue indicating whether the system has incurred an error. Secondly, it exploited the linearity of quantum mechanics so that a continuous set of errors (such as arbitrary rotations) could be corrected by correcting the discrete set of Pauli errors [13, 24]. These important ideas have since been cemented into the *stabilizer formalism* [25] for QECC which has allowed the field to borrow heavily from classical error correction. While, the idea of protecting information by encoding in a larger subspace carries over naturally and has led to the development of quantum versions of classical codes [7, 8, 24, 26], other ideas like decoding strategies [27–30] have also been adopted from the field of classical error correction.

1.3.1 The stabilizer formalism

The stabilizer formalism is a framework that views QECC as maps of logical qubits to physical qubits without focusing on the individual codewords, i.e., the state of the physical qubits. Rather, the approach here is to establish a subspace of the Hilbert space, known as the *codespace*, where error correction is possible, and analyze the various operations

(logical, and error-recovery) allowed within it. Codes within this formalism are also known as stabilizer codes, and will be the only class of QECC discussed in this thesis. The stabilizer formalism [25] is briefly summarized below:

Stabilizer codes

Definition 1. A stabilizer group $\mathcal{S}$ is an Abelian subgroup of the n -qubit Pauli group $\mathcal{P}^{\otimes n}$.

$$\mathcal{S} = \{S \in \mathcal{P}^{\otimes n} \mid S|\psi\rangle = |\psi\rangle\}$$

where $|\psi\rangle$ is a codeword of the QECC defined by $\mathcal{S}$. The stabilizer group $\mathcal{S}$ defines a non-trivial code if $-\mathbb{1} \notin \mathcal{S}$.

Thus, the simultaneous $+1$ eigenspace of the stabilizer group $\mathcal{S}$ defines the codespace of a QECC. For a group $\mathcal{S}$ that can be generated by a set of r independent generators, $\mathcal{S} = \langle S_1, S_2, \dots, S_r \rangle$, we say that the QECC having stabilizers $\mathcal{S}$ encodes $k = n - r$ logical qubits.

Definition 2. CSS stabilizer codes are codes whose stabilizer generators can be expressed as a set of only Pauli- X operators and only Pauli- Z operators i.e.

$$\mathcal{S}_{CSS} = \langle S_{X_1}, S_{X_2}, \dots, S_{Z_1}, S_{Z_2}, \dots \rangle$$

Logical operations of stabilizer codes

Definition 3. The centralizer $\mathcal{Z}(\mathcal{S})$ is the subgroup of $\mathcal{P}^{\otimes n}$ containing elements that commute with the stabilizer $\mathcal{S}$.

By definition, each element of $\mathcal{S}$ leaves the codeword unchanged and therefore performs the logical identity operation on the codespace. Since the elements of $\mathcal{Z}(\mathcal{S})$ commute with the elements of $\mathcal{S}$, they map the codespace onto itself. Thus elements in $\mathcal{Z}(\mathcal{S})$ are logical operations on the encoded qubit. In particular, the set $\mathcal{Z}(\mathcal{S}) \setminus \mathcal{S}$ contains the non-trivial logical operations of the code (i.e. not including logical identity). Note here that $\mathcal{S} \subseteq \mathcal{Z}(\mathcal{S})$, and that for any element $Z \in \mathcal{Z}(\mathcal{S})$ and $S \in \mathcal{S}$, Z and ZS have the same action on the codespace. Therefore the quotient set $\mathcal{Z}(\mathcal{S})/\mathcal{S}$ is the set of representatives of the logical operations on the codespace.

Error correction with stabilizer codes

It has been shown that for correcting the continuous set of all n -qubit errors, it is sufficient to correct [13] the discrete set of n -qubit Pauli errors. Thus, the correction of Pauli errors is the main focus of error correction with stabilizer codes. It is also generally assumed that the noise in a quantum system is local, and each qubit is affected by independent, identically distributed (i.i.d.) Pauli errors, although the latter assumption can be relaxed in some cases [31] to account for more realistic correlated noise in quantum systems.

Error-detection in a stabilizer code is performed by applying a *syndrome measurement* to obtain the eigenvalues of the stabilizer generators with respect to the state of the system. Syndrome measurements are implemented using a phase-estimation circuit [13]. As an example, for a code with stabilizer generators X_1X_2, Z_1Z_2 , the circuits to measure X_1X_2 (left) and Z_1Z_2 (right) operators are shown below—

$$\begin{array}{c}
 |0\rangle_A \text{ --- } [H] \text{ --- } \bullet \text{ --- } \bullet \text{ --- } [H] \text{ --- } \text{Measurement} \\
 \text{--- } \oplus \text{ --- } \text{---} \\
 \text{--- } \oplus \text{ --- } \text{---}
 \end{array}
 \quad
 \begin{array}{c}
 |0\rangle_A \text{ --- } \oplus \text{ --- } \oplus \text{ --- } \text{Measurement} \\
 \text{--- } \bullet \text{ --- } \text{---} \\
 \text{--- } \bullet \text{ --- } \text{---}
 \end{array}
 \quad (1.47)$$

Here, the qubit labelled A is the ancilla qubit from which the eigenvalue is retrieved after measurement.

When the state of the system is found to be in the -1 eigenspace of any stabilizer generator of $\mathcal{S}$, an error is said to be detected. Suppose a codeword $|\psi\rangle$ is affected by an error E that anticommutes with the i^{th} stabilizer generator,

$$|\psi\rangle \rightarrow E|\psi\rangle, \quad S_i E|\psi\rangle = -E|\psi\rangle. \quad (1.48)$$

A suitable error-recovery operator R , such that $RE \in \mathcal{S}$, can then be applied to the state to correct the error and return the system to the codespace. Syndrome measurements are thus similar to parity checks in classical error correction. Since syndrome measurements only provide classical information (eigenvalues) about the quantum system, the decoding strategy for stabilizer codes is essentially a classical problem involving a search for the appropriate recovery operation R for a given error E . Additionally, for CSS codes, the decoding problem for X and Z errors can be solved independently [25]. Note that this approach for CSS codes may be suboptimal, particularly when the noise in the system includes correlated X and Z errors.

A Pauli operator $E \in \mathcal{P}^{\otimes n}$ either commutes or anticommutes with each of the stabilizer generators. When a Pauli operator E of weight w (the number of non-Identity locations)

anticommutes with any of the generators of $\mathcal{S}$, the syndrome measurements corresponding to those generators results in the eigenvalue -1 . Such an operator E is called a detectable error of weight w . When an error E commutes with $\mathcal{S}$, it is an element of the centralizer $\mathcal{Z}(\mathcal{S})$ making it an undesired logical operation, i.e., a logical error. Since logical operations do not map codewords out of the codespace, logical errors are undetectable. Thus, if all the elements in the set of logical operations of the code have a minimum weight d , then this prevents arbitrary Pauli errors with weight $w \geq d$ from being detectable. This is the motivation for defining the distance of a stabilizer code.

Definition 4. *The distance of a stabilizer code ‘ d ’ is the minimum weight of elements in $\mathcal{Z}(\mathcal{S}) \setminus \mathcal{S}$. A stabilizer code is degenerate if $\mathcal{S}$ has an element (other than Identity) with weight less than d .*

A stabilizer QECC with k logical qubits encoded in n physical qubits with distance d is an $[[n, k, d]]$ code. An $[[n, k, d]]$ code can detect errors with weight up to $d - 1$ and correct those with weight up to $t = \lfloor \frac{d-1}{2} \rfloor$.

1.3.2 Shor’s 9-qubit code

Shor’s 9-qubit is an example of a stabilizer code. It is a $[[9, 1, 3]]$ CSS code whose X and Z stabilizer generators are given by

$$\mathcal{S}_X = \{X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9\} \quad (1.49)$$

$$\mathcal{S}_Z = \{Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9\}. \quad (1.50)$$

Shor’s code has distance $d = 3$, and thus it can be used to correct any single-qubit Pauli error on the physical qubits. Fig. 1.1 shows the circuit for encoding a state in the $[[9, 1, 3]]$ code, and syndrome measurement circuits for the stabilizer generators $X_1X_2X_3X_4X_5X_6$ and Z_8Z_9 . The logical states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are encoded as follows—

$$|\bar{0}\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \quad (1.51)$$

$$|\bar{1}\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle). \quad (1.52)$$

Representatives of logical Pauli operators $\bar{X}$ and $\bar{Z}$ are given by—

$$\bar{X} = Z_1Z_4Z_7, \quad \bar{Z} = X_1X_2X_3 \quad (1.53)$$

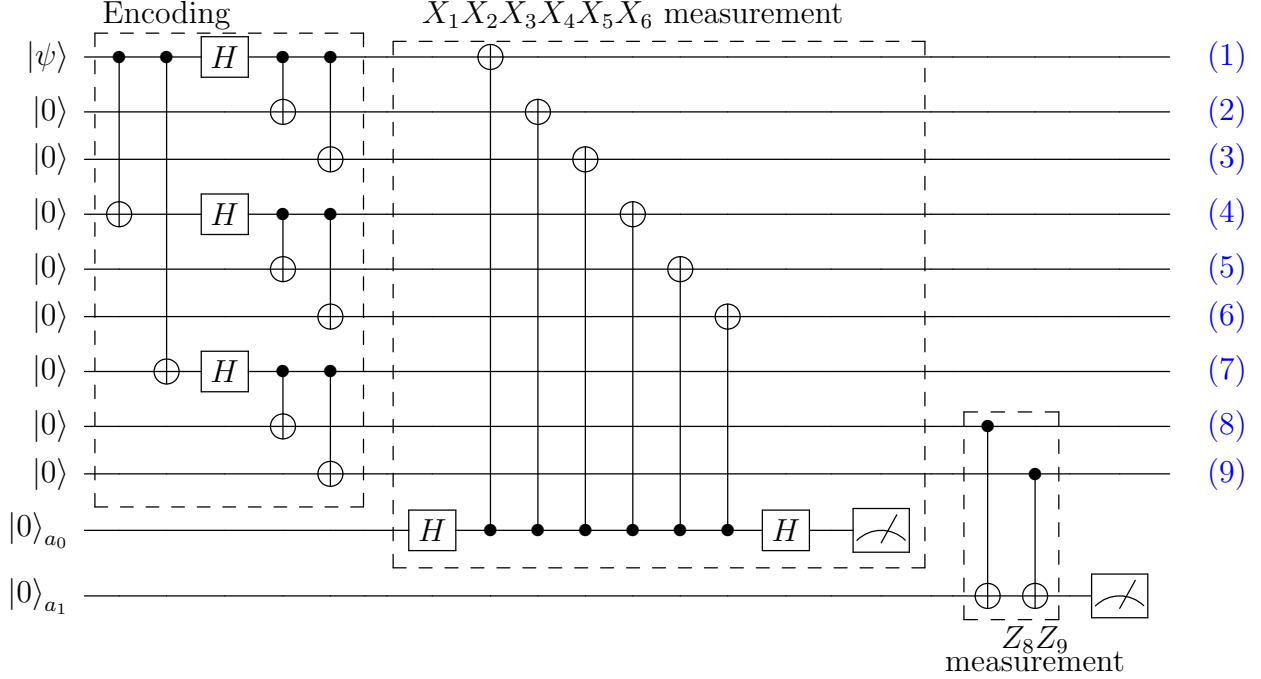


Figure 1.1: Encoding circuit for Shor's code (qubits indexed in blue on the right) to encode a state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ into the state $|\bar{\psi}\rangle = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle$ (see Eq. 1.51). The syndrome measurement circuits for stabilizer generators $X_1X_2X_3X_4X_5X_6$ and Z_8Z_9 are also shown. The measurement of ancilla qubits a_0, a_1 gives the eigenvalues of $X_1X_2X_3X_4X_5X_6$ and Z_8Z_9 respectively (see Eq. 1.47).

1.3.3 Topological quantum codes: The 2D Toric code reviewed

Topological quantum codes are a family of stabilizer codes associated with the spatial arrangement of qubits on a manifold with stabilizer generators that are geometrically local. The 2D Toric code [32, 33] is a popular example of a topological quantum code. It is a CSS code defined by $\mathcal{S} = \langle F_Z, V_X \rangle$ where F_Z is a *plaquette stabilizer* generator and V_X is a *vertex stabilizer* generator as shown in Fig 1.2. It can be visualized as a square lattice embedded on a torus. The lattice of dimension $L \times L$ has qubits residing on the edges of the lattice. F_Z corresponds to $Z^{\otimes 4}$ acting on the qubits on a square cell of the lattice and V_X corresponds to $X^{\otimes 4}$ acting on the qubits that meet at a vertex. The 2D Toric code can be laid out on a flat surface by considering a square lattice with periodic boundary conditions, i.e., the top and bottom of the lattice wrap around, as do the right and left

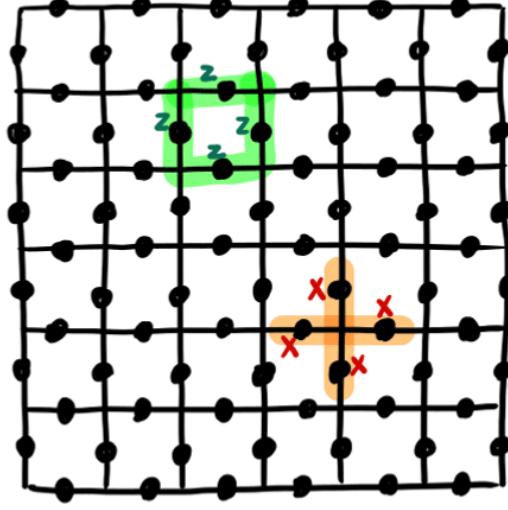


Figure 1.2: Plaquette F_Z and Vertex V_X stabilizer generators of the 2D Toric Code with qubits (black dots) on the edges of a square lattice with periodic boundary conditions.

edges. The number of physical qubits in this lattice is $n = 2L^2$, while there are $f = L^2$ plaquette stabilizers (number of faces) and $v = L^2$ vertex stabilizers (number of vertices). However, because of periodic boundary conditions, the Z -type (and X -type) stabilizers are not all independent. In fact, the Z -type (and X -type) stabilizers satisfy—

$$\prod_f F_z = \mathbb{1}, \quad \prod_v V_x = \mathbb{1} \quad (1.54)$$

Therefore, the number of independent stabilizer generators of the 2D Toric code is $r = 2L^2 - 2$. Thus the Toric code encodes $k = n - r = 2$ logical qubits.

The logical operators of the 2D Toric code are closed string operators, as shown in Fig. 1.3.

Note that the horizontal and vertical strings intersect at precisely one qubit location and thus they have the appropriate commutation relations to be the $\bar{X}_1$ and $\bar{Z}_1$. The logical operator corresponding to $\bar{X}_2$ ($\bar{Z}_2$) is the vertical (horizontal) closed string of X (Z) operators in the dual⁴ lattice (regular lattice). It is known that the distance of the 2D Toric code is L , the dimension of the lattice. Thus, the 2D Toric code is an $[[2L^2, 2, L]]$ code. The number of logical qubits encoded in the 2D Toric code is independent of the

⁴In this case, the dual lattice is the square lattice having the qubits on its vertices.

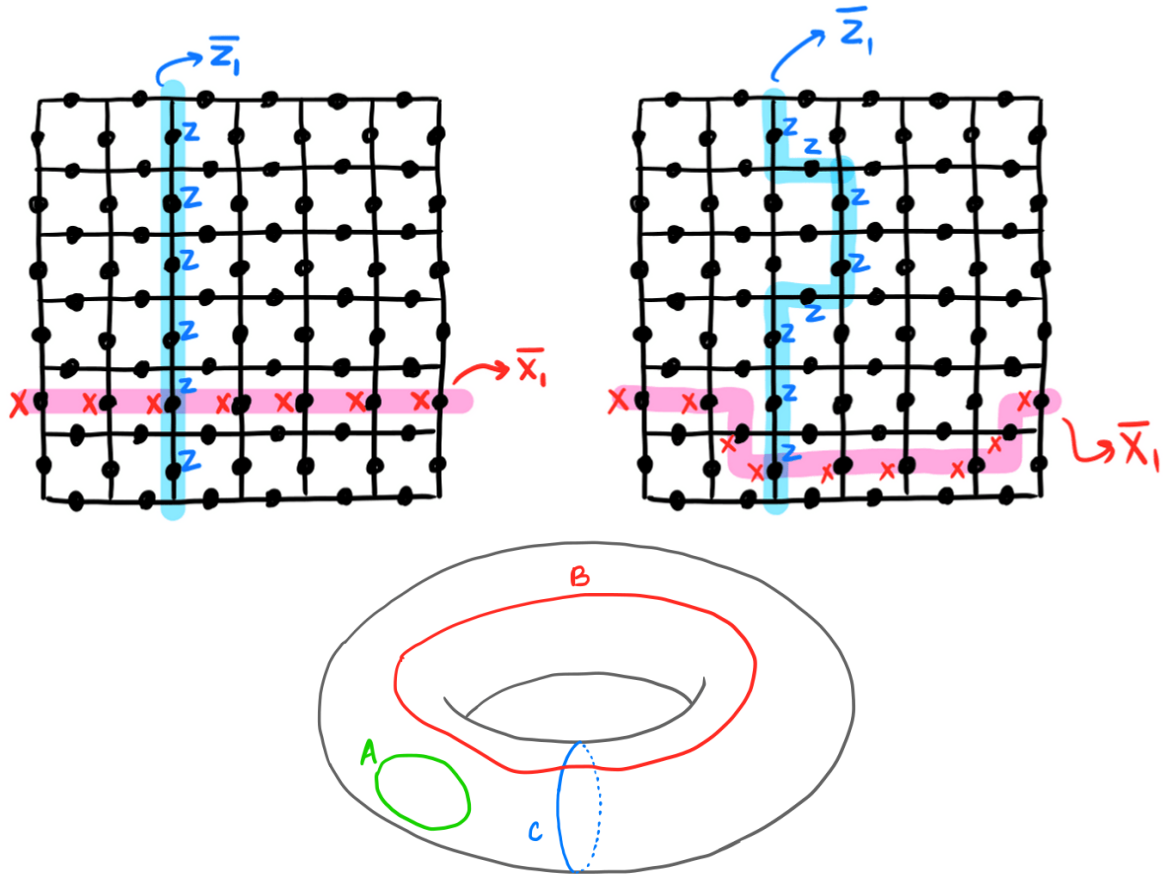


Figure 1.3: Logical operators of the 2D Toric code are closed strings on the regular (and dual) lattice. The operators $\bar{X}_1$ and $\bar{Z}_1$ are the $\bar{X}$ and $\bar{Z}$ logical operation on one of the two logical qubits encoded in the 2D Toric code, where $\bar{X}_1$ is a closed string on the dual lattice while $\bar{Z}_1$ is a closed string on the original lattice. The two blue string operators ($\bar{Z}_1$) are equivalent even though one of them is deformed, since either can be obtained from the other by multiplying by stabilizer generators. The non-trivial logical operators (loops B, C) form are non-trivial closed loops on the torus while the stabilizers (logical identity) form closed loops on the surface of the torus that are continuously deformable to a single point (loop A).

number of physical qubits and is related to the topology of the lattice arrangement (in this case, the torus).

It is common to represent topological quantum codes in terms of the Hamiltonian that

can be constructed from their stabilizer generators—

$$H = - \sum_{j=1}^r S_j \quad (1.55)$$

Since the codewords are $+1$ eigenvalues of the stabilizer generators, they are degenerate groundstates of H . The Hamiltonian corresponding to the Toric code is given by

$$H_{\text{toric}} = - \sum_f F_Z - \sum_v V_X \quad (1.56)$$

The geometry of the lattice ensures that any detectable Pauli error anticommutes with at least one pair of stabilizer generators. Every stabilizer generator it anticommutes with raises the energy of the system by one unit resulting in an *excitation*. Consider the single-qubit Z operator on a qubit on the lattice as shown in Fig 1.4. It anticommutes with the two vertex (X) stabilizers adjacent to it producing two X -type excitations (also called electric charges). Similarly X errors can produce Z -type excitations (also known as magnetic flux).

Since detectable errors in the 2D Toric code occur in pairs (like the two ends of a string), the most common strategy for correcting errors is to match an excitation to its most likely pair. This strategy is implemented using the classical algorithm known as minimum weight perfect matching (MWPM) [33, 34]. The 9-qubit code and the 2D toric code discussed above are two illustrative examples of QECC. Since the discovery of these codes, there have been other code constructions with different useful properties such as fault-tolerant logical operations, efficient encoding rate (i.e., the ratio of logical to physical qubits k/n), etc. There have also been many developments in efficient classical decoding strategies for these codes. Chapter 2 of this thesis involves a detailed discussion of a decoding strategy based on cellular automata that has been proven to be successful for many topological codes.

1.4 Fault-tolerant error correction

An $[[n, k, d]]$ error correcting code provides a way to correct errors of weight at most $t = \lfloor \frac{d-1}{2} \rfloor$ in a system of n physical qubits that encodes k logical qubits. Due to the sensitivity of qubits to their environment, errors in quantum systems can rapidly accumulate. If these errors are permitted to accumulate in an uncontrolled manner, the effective error in the system could have weight $w > t$ and potentially become an undetectable (and hence uncorrectable) logical error. Thus an error correcting code is only effective at protecting

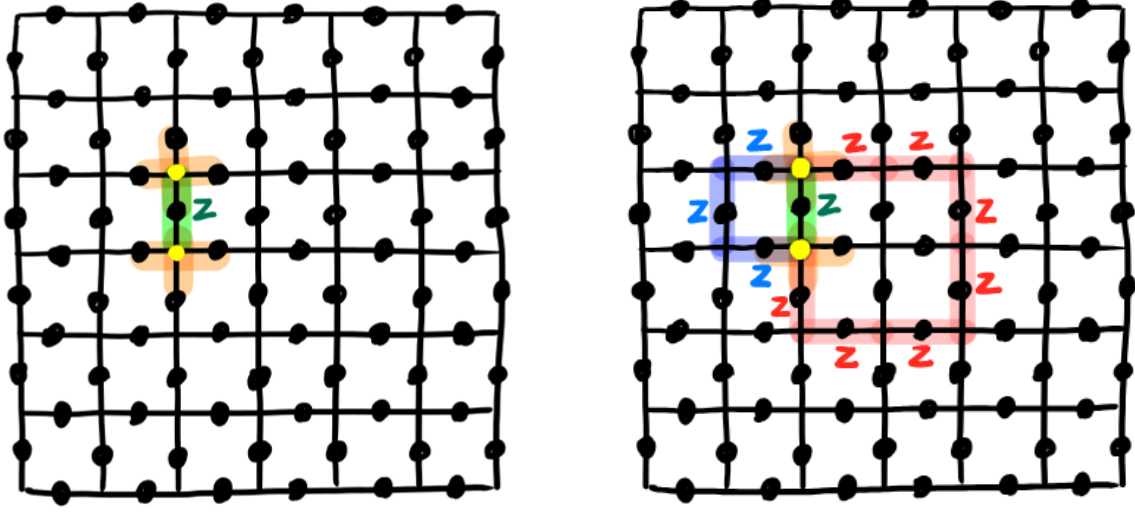


Figure 1.4: (Left) Excitations due to a single-qubit Z error are represented by yellow dots (in the dual lattice) and correspond to the two vertex stabilizers (X -type) which anticommute with this error. (Right) Consider the case when a 3-qubit Z error (shaded blue) occurs in the system. The three Z string operators having weight 1 (shaded green), weight 3 (shaded blue, i.e., the error itself) and weight 7 (shaded red) are all possible recovery operations because when multiplied by the error, they result in an element of the stabilizer group (and implement logical identity). The minimum weight perfect matching (MWPM) algorithm is a classical decoding strategy which matches each excitation to its most likely pair, and in this case would select the string having the lowest weight, i.e., single qubit Z operator as the suitable error-recovery operation. In fact, it would apply this same recovery operation when any of the three Z strings occur as errors in the system.

logical information if it can be guaranteed that the errors most likely to be acting on the system have weight $w \leq t$. The field of fault-tolerant error correction is concerned with protocols to control the propagation of errors. One natural strategy is to periodically perform error correction cycles (comprising syndrome measurements followed by recovery operations) so that errors are corrected without allowing them to accumulate over time. The frequency of the error correction cycles would depend on the expected time taken for errors of weight t to occur with high probability (which can be evaluated from the physical error rates). In most quantum devices, the 2-qubit entangling gates have an error rate that is at least one order of magnitude higher [35, 36] than the error rates of single qubit gates. Moreover, since entangling gates involve an interaction between multiple qubits, errors on one qubit can propagate to other qubits in the interaction. For instance, an X

error on the control qubit of a CNOT gate can propagate to the target qubit, causing a single qubit error to evolve into a 2-qubit error. Thus, another strategy for fault-tolerance is to minimize the number of 2-qubit entangling gates required for logical operations in an error correcting code. *Transversal logical gates* are one such strategy. A logical operation in a code that can be implemented using only single qubit gates on the physical qubits is an example of a transversal gate. Such a logical operation prevents the physical qubits in a single code block from interacting, thus controlling the propagation of error within that code block. Consider for example the $[[7, 1, 3]]$ Steane code [7, 25] having the following stabilizer generators—

$$\begin{aligned} Z_1 Z_2 Z_3 Z_4, \\ Z_1 Z_2 Z_5 Z_6, \\ Z_1 Z_3 Z_5 Z_7, \\ X_1 X_2 X_3 X_4, \\ X_1 X_2 X_5 X_6, \\ X_1 X_3 X_5 X_7. \end{aligned}$$

Some of the transversal logical operations available in the Steane code are—

$$\begin{aligned} \overline{X} &= X_1 X_2 X_3 X_4 X_5 X_6 X_7, \\ \overline{Z} &= Z_1 Z_2 Z_3 Z_4 Z_5 Z_6 Z_7, \\ \overline{H} &= H_1 H_2 H_3 H_4 H_5 H_6 H_7, \end{aligned}$$

where H is the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (1.57)$$

Additionally, the Steane code also has a transversal CNOT logical gate (shown in Fig. 1.5) where a logical CNOT can be applied to two logical qubits (in two different codeblocks) without applying entangling gates within a codeblock. Interestingly, it is known that all CSS codes have a transversal logical CNOT gate [10] that can be implemented between two code blocks as shown in Fig. 1.5. While transversal logical operations are a simple and popular approach to fault-tolerance, the Eastin-Knill theorem [37] proves that no QECC can have a universal set of transversal logical gates. Chapter 3 presents details of implementing fault-tolerant logical operations in the $[[8,3,2]]$ code, an important color code (which is a class of topological quantum codes, of which the Steane code is the simplest example).

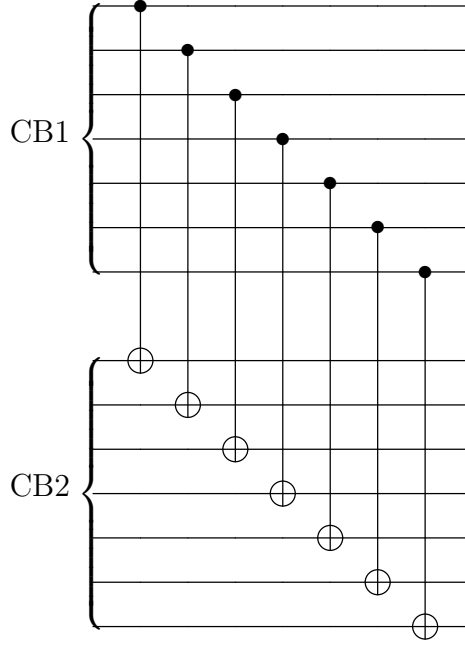


Figure 1.5: Transversal logical CNOT between two codeblocks (CB1 and CB2) of the $[[7, 1, 3]]$ Steane code. This is a fault-tolerant logical CNOT gate because no two qubits within the same codeblock interact, curtailing the uncontrolled spread of errors within the block.

1.5 Error mitigation protocols

The field of fault-tolerance has shown that for sufficiently low physical error rates, it is possible to encode qubits in error correcting codes and control the propagation of errors in the logical qubit to perform useful quantum computation. However, fault-tolerant error corrected architectures are resource intensive due to the large number of physical qubits required per logical qubit. For instance, the 2D surface code (which is derived from the 2D Toric code) is a popular QECC candidate for fault-tolerant error corrected architectures. As discussed in Sec. 1.3.3, these codes require $\mathcal{O}(d^2)$ physical qubits to encode (with distance d) a single qubit. Other topological codes (such as color codes) which are well-suited for fault-tolerant architectures also have similar qubit overheads. Current error corrected algorithmic primitives estimate the number of logical qubits required to perform useful tasks (quantum simulation, prime factorization) at approximately 10^4 qubits [38]. Significant engineering challenges need to be overcome in order to improve quantum control and reduce physical error rates to regimes in which large-scale fault-tolerant quantum

computers will be practical.

The field of error mitigation has developed as a way to employ the noisy intermediate-scale quantum (NISQ) devices available today to perform useful tasks without error correction. The spirit of error mitigation protocols is to repeat noisy circuits or use multiple instances of noisy quantum resources on a device and subsequently use classical post-processing techniques to piece together the ideal version of those operations. Since error correction mechanisms are not required, error mitigation strategies tend to have low (nearly constant) qubit overhead, generally not requiring more qubits than those in the actual quantum system of interest. The space cost (number of extra qubits that would be required for error correction) is traded-in for an increased (usually exponential) sampling cost on the device [39].

Since the focus of error mitigation is to use quantum computers without the machinery of error correction, the system is not guaranteed to be protected from noise and quantum information cannot remain coherent for long times. In order to preserve quantum resources such as entanglement and perform useful tasks with NISQ devices, the quantum circuits used in error mitigation are constrained to have low-depth [39], preventing them from performing prolonged computation on quantum states. Due to this, the goal in most error mitigation protocols is the estimation of (ideal) expectation values of observables without necessarily requiring a coherent preparation of the actual quantum state of interest.

1.5.1 Probabilistic error cancellation

Probabilistic error cancellation (PEC) is an example of an error mitigation technique that has been used for inverting noise in a quantum device [40]. Given a device whose (noisy) native gates are characterized, an ideal unitary U that is to be implemented can be decomposed into a linear combination (with real coefficients) of the (noisy) gates available on the NISQ device. This linear combination (see Eq. 1.58) is known as a quasiprobability decomposition (QPD) and has coefficients that sum up to 1. More specifically, for a complete set of basis operations $\{\mathcal{B}_i\}$ and the noise channel $\mathcal{N}$ on a noisy quantum device, there exists a quasi-probability decomposition (QPD) onto the noisy basis $\{\mathcal{E}_i = \mathcal{N} \circ \mathcal{B}_i\}$ with real coefficients q_i given by—

$$\mathcal{U}(\rho) = \sum_i^m q_i \mathcal{E}_i(\rho), \quad \sum_i^m q_i = 1, \quad (1.58)$$

$$\gamma = \sum_i^m |q_i|, \quad p_i = |q_i|/\gamma. \quad (1.59)$$

Here $\mathcal{U}(\rho) = U\rho U^\dagger$ and p_i is a true probability distribution associated with the QPD. Since the coefficients are real numbers which may be negative, the *negativity* $\gamma \geq 1$ of the QPD indicates how far the QPD is from a true probability distribution.

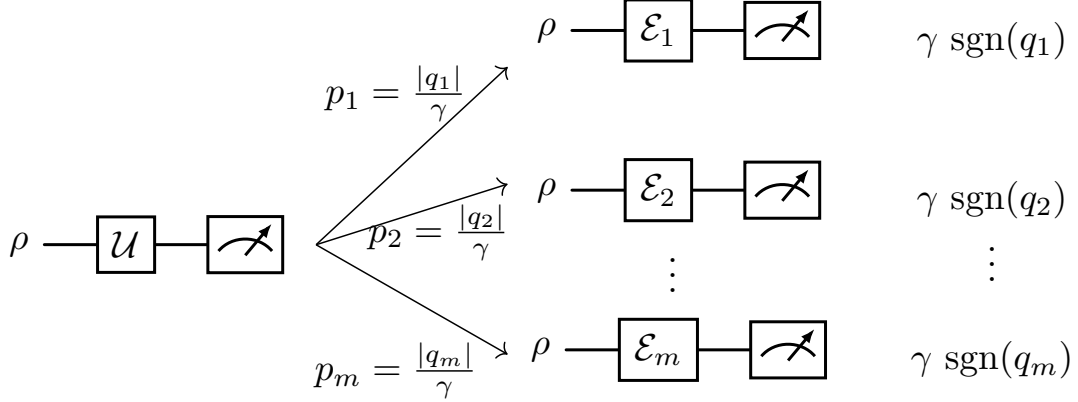


Figure 1.6: Probabilistic error cancellation via a quasiprobabilistic distribution of noisy basis operations $\{\mathcal{E}_i\}$ available on a NISQ device. The expectation value of the observable A can be estimated with respect to the ideal state $\mathcal{U}(\rho) = U\rho U^\dagger$ by sampling circuits from the basis according to the probability distribution $\{p_i\}$.

$$\langle A \rangle_{\mathcal{U}(\rho)} := \operatorname{Tr} [A \mathcal{U}(\rho)] \quad (1.60)$$

$$= \gamma \left(\sum_i^m \operatorname{sgn}(q_i) \cdot p_i \cdot \operatorname{Tr} [A \mathcal{E}_i(\rho)] \right) \quad (1.61)$$

$$= \gamma \left(\sum_i^m \operatorname{sgn}(q_i) \cdot p_i \cdot \langle A \rangle_{\mathcal{E}_i(\rho)} \right) \quad (1.62)$$

Given the QPD in Eq. 1.58, the expectation value of an observable A (having operator norm $\|A\| \leq 1$) with respect to the ideal state $\mathcal{U}(\rho)$ can be estimated (Eq. 1.62) using Algorithm 1. In particular, this is done by taking the weighted sum of the measurement outcomes of each (noisy) circuit that is randomly sampled from the possible basis operations $\mathcal{E}_i = \mathcal{N} \circ \mathcal{B}_i$ according to the probability distribution $\{p_i\}$ (see Fig. 1.6). The number

Algorithm 1 Estimate expectation value of A w.r.t. $\mathcal{U}(\rho)$

input: QPD of unitary gate U as $\mathcal{U}(\cdot) = \sum_i q_i \mathcal{E}_i(\cdot)$, initial state ρ

output: estimate of expectation value $\langle A \rangle_{\mathcal{U}(\rho)}$

- 1: **for** $j = 1$ to N **do**
 - 2: sample i from distribution $|q_i|/\gamma$
 - 3: apply $\mathcal{B}_i$ to ρ , implementing its noisy version $\mathcal{E}_i$ on the device
 - 4: measure the observable A and store its outcome M_j
 - 5: store the weighted measurement outcome $W_j = \gamma \cdot \text{sgn}(q_i) \cdot M_j$
 - 6: **return** empirical estimate of $\text{Tr}[A\mathcal{U}(\rho)] = \sum_{j=1}^N W_j$
-

of samples N required in Algorithm 1 to estimate $\langle A \rangle_{\mathcal{U}(\rho)}$ up to additive error ϵ with probability $1 - \delta$, where $0 \leq \delta, \epsilon < 1$, is given by [40–42]—

$$N = \frac{2\gamma^2}{\epsilon^2} \log \left(\frac{1}{\delta} \right). \quad (1.63)$$

The budding field of error mitigation has brought to the forefront many classical techniques for quantum simulation which may benefit from acceleration with quantum computers. Notably, PEC is an error mitigation strategy that was inspired by a classical technique for quantum simulation based on quasiprobability decompositions [40, 43]. In chapter 4 of this thesis, a similarly-inspired quasiprobabilistic method for implementing imaginary time evolution on quantum computers is discussed.

Chapter 2

Sweep rule: A cellular automaton decoder for topological codes

Topological codes are a promising candidate for fault-tolerant error correction architectures. The 2D Toric code discussed in Sec. 1.3 is an ubiquitous example of this class of codes. As mentioned in Sec. 1.3.3, errors in the 2D Toric code manifest as pairs of X -type or Z -type excitations. A popular decoding strategy for these errors is minimum weight perfect matching (MWPM), a classical algorithm that corrects errors by systematically matching each excitation with its most likely pair (see Fig. 1.4.) Extensions of the Toric code to $D \geq 3$ dimensions permit transversal non-Clifford logical operations [44] such as the CCZ gate, which are appealing for quantum computation because they are not efficiently classically simulable [45]. However, errors in the 3D and 4D Toric code are slightly different than those in the 2D Toric code, and therefore require different decoding algorithms. *Sweep rule* is a cellular automaton (CA) which has inspired decoders for topological quantum codes such as the 3D Toric code and will be the subject of study in this chapter. In order to understand the motivation behind sweep rule, it is important to first consider as an example the 3D Toric code and the manifestation of errors in this code.

2.1 The 3D Toric code

The 3D Toric code is a simple extension of the 2D Toric code on a three-dimensional lattice [44]. It is a CSS code encoding 3 logical qubits in an $L \times L \times L$ cubic lattice with $3L^3$ physical qubits arranged on the edges and having distance L . The stabilizer group of the 3D Toric code is generated by $\mathcal{S} = \langle F_Z, V_X \rangle$ where F_Z are Z -operators of weight

12 acting on the 12 qubits on the edges of each cubic cell of the lattice, while V_X are the X -operators of weight 6 acting on the qubits on all the edges incident on each vertex (see Fig. 2.1).

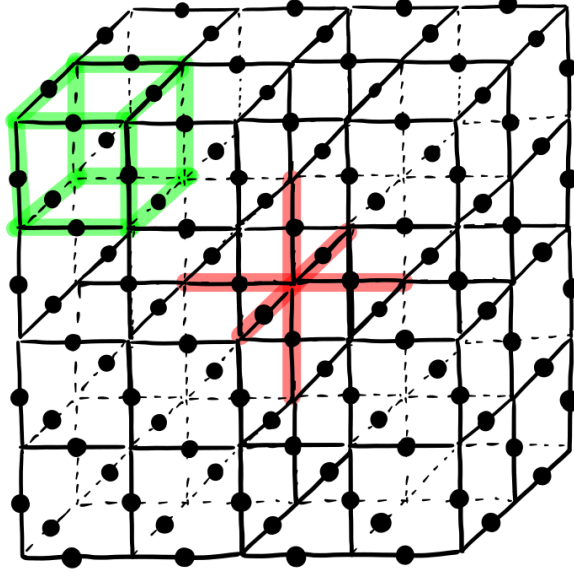


Figure 2.1: The 3D Toric code is an $[[3L^3, 3, L]]$ code with physical qubits (black dots) on the edges of an $L \times L \times L$ cubic lattice. The weight-12 Z stabilizer generators (green) act on each qubit on the edges of a cubic cell while the weight-6 X stabilizer generators (red) act on all the qubits on the edges incident on a vertex of the lattice.

Just as in the 2D Toric code, Z errors in the 3D Toric code produce pairs of X -type excitations. However, X errors in the 3D Toric code produce loop-like excitations in the dual lattice (see Fig. 2.2). Therefore, although Z errors can be decoded using an algorithm similar to MWPM, the correction of X errors requires a fundamentally different strategy. Like the 3D Toric code, many topological codes in $D \geq 3$ dimensions tend to have at least one type of errors (X or Z) that produces loop-like excitations (or excitations that form k -simplices for $k \geq 2$) in the volume of the lattice. Classical CA are a prime candidate for the development of a generalized approach to correcting such errors since they are data structures that evolve using geometrically-local update rules and whose implementation can be parallelized. The sweep rule is a classical CA that inspired the sweep decoder [28] for topological color codes [46–49]. At its crux, the sweep decoder corrects errors by using sweep rule to systematically shrink loop-like excitations generated within the lattice. Moreover, unlike other similar strategies to shrink loop-like excitations

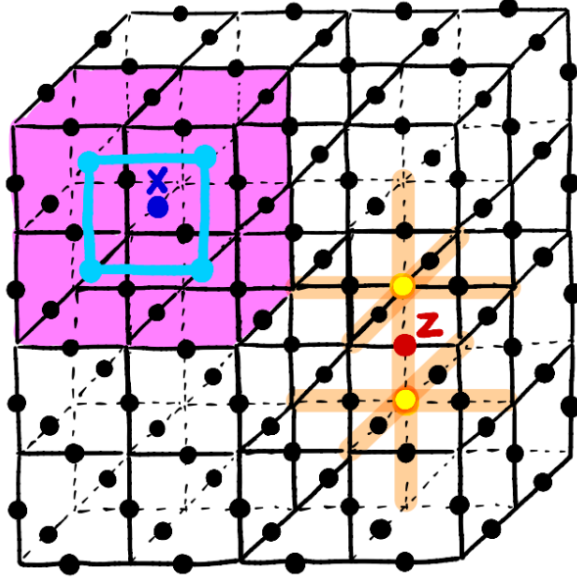


Figure 2.2: Errors in the 3D Toric code can produce excitations of two types. A single-qubit Z error (red) in the code anticommutes with two V_X stabilizers producing a pair of X -type excitations (yellow). A single-qubit X error (blue) anticommutes with four F_Z stabilizers producing four Z -type excitations (light blue) that form a closed loop in the dual lattice of the code. In general, Z -errors produce string-like excitations, as in the 2D Toric code. But unlike the 2D case, X -errors in the 3D Toric produce loop-like excitations in the dual lattice.

(such as the Glauber CA based on the 2D Ising model [50]), the sweep rule is robust to bias ($h \neq 0$) in syndrome measurement errors, i.e., incorrectly reported outcomes of syndrome measurements. The remainder of this chapter involves an investigation of the error correction properties of sweep rule via the study of a classical memory model based on sweep rule [1]. In Sec. 2.2, we describe the classical memory model based on the sweep rule. In Sec. 2.3, we numerically analyze the memory lifetime of the model. To corroborate our results for $h = 0$, we identify the phase transition of the sweep rule using statistical-mechanical techniques in Sec. 2.4 and discuss a simple analytical model that captures the memory lifetime at high temperatures in Sec. 2.5. We conclude by discussing the impact of our results in Sec. 2.6.

2.2 Error correction with sweep rule

We introduce the sweep rule by first presenting a simple model of classical memory. This model consists of a 2D system of $N = 2L^2$ classical two-state spins arranged on the faces of a triangular lattice of linear size L with periodic boundary conditions. The state of the i^{th} spin is denoted by $s_i^{(t)} \in \{+1, -1\}$ for $i \in \{1, 2, \dots, N\}$ at discrete half-integer time steps $t = 0, 0.5, 1, \dots$. At $t = 0$, we *encode* a logical bit in this system in one of two completely aligned configurations, either $\forall i : s_i^{(0)} = +1$ or $\forall i : s_i^{(0)} = -1$, corresponding to logical state 1 or 0, respectively.

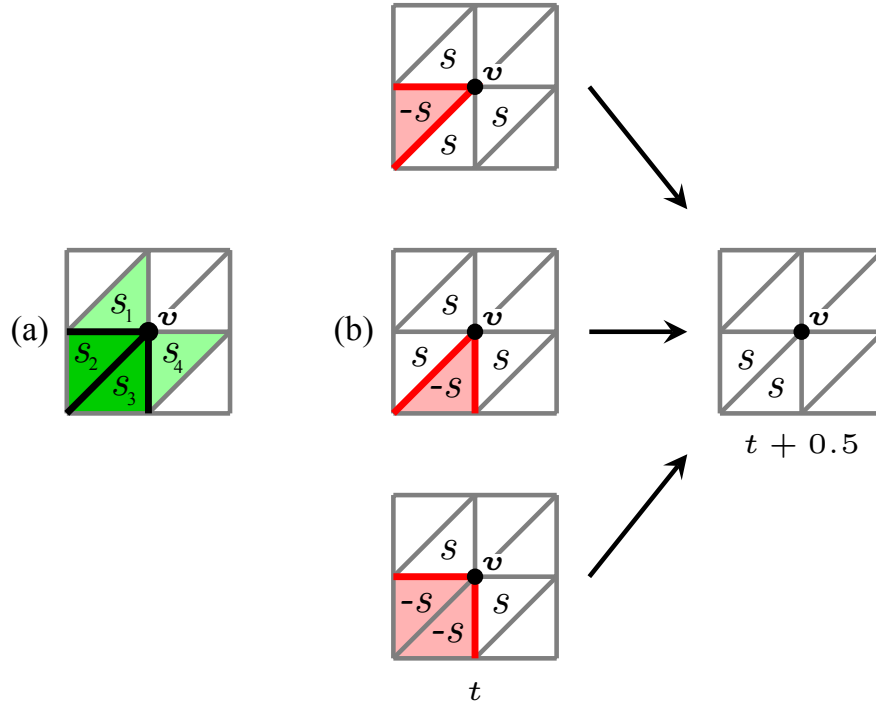


Figure 2.3: (a) Neighborhood of the vertex v comprises the highlighted spins s_1, s_2, s_3, s_4 situated on the (light and dark green) triangular faces. The deterministic rule updates s_2, s_3 (dark green) based on the restriction of the domain wall on the thick black edges. (b) Different configurations of the restricted domain wall (thick red edges) at time t and the resulting updated spins at $t + 0.5$.

The system evolves under the sweep rule applied at each vertex of the lattice simultaneously (*synchronous update*). The sweep rule is a probabilistic cellular automaton (PCA) [51] comprising a deterministic and a probabilistic update rule. A single time step

$t \rightarrow t + 1$ of this PCA can be divided into two half-steps. In the first half-step $t \rightarrow t + 0.5$, the system is updated by the deterministic rule and in the second half-step $t + 0.5 \rightarrow t + 1$, the system is updated by the probabilistic rule. In a single time step, the sweep rule can only update the spins s_2, s_3 in the neighborhood of the vertex v (see Fig. 2.3). The deterministic rule updates s_2, s_3 based on the local configuration of the domain wall, which is defined as the set of edges of the lattice that separate oppositely aligned spins. In particular, if the spins s_1 and s_4 are equal, then the deterministic rule updates spins s_2, s_3 to match them, i.e., if $s_1^{(t)} = s_4^{(t)}$, then set $s_2^{(t+0.5)} = s_3^{(t+0.5)} = s_1^{(t)} = s_4^{(t)}$. The probabilistic rule then updates the spins s_2 and s_3 as follows

$$\Pr \left(s_i^{(t+1)} = +1 \middle| s_i^{(t+0.5)} = -1 \right) = p, \quad (2.1)$$

$$\Pr \left(s_i^{(t+1)} = -1 \middle| s_i^{(t+0.5)} = +1 \right) = q, \quad (2.2)$$

where $p, q \in [0, 1]$ are the two parameters of the model. A single time step in the evolution of the system comprises synchronous update of all N spins with the deterministic rule followed by the probabilistic rule. In Appendix A.3, we also report properties of a variant of this PCA, the *asynchronous* sweep rule, where in a single time step the sweep rule is applied sequentially at N vertices selected via simple random sampling with replacement. Since the sweep rule updates the state of the system based only on its current state, its evolution is a Markov process.

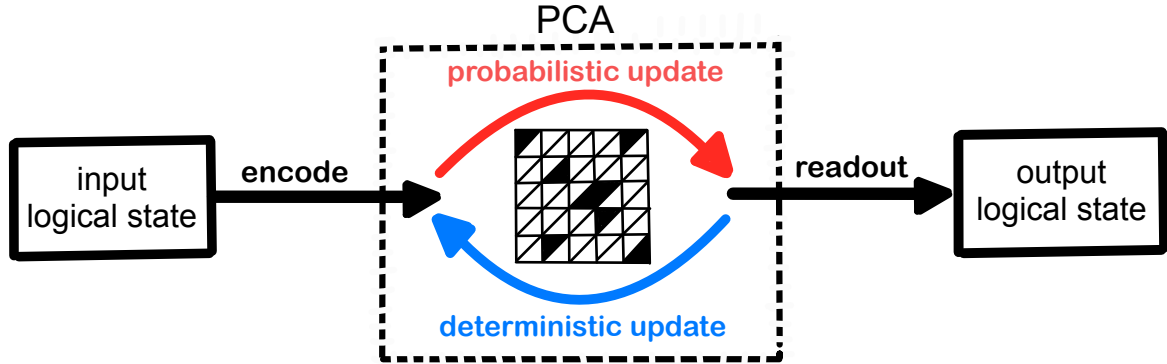


Figure 2.4: Schematic model of classical memory that protects logical information via a probabilistic cellular automaton. The deterministic update removes noise (random spin-flips) added to the system in the probabilistic update.

At time t , the instantaneous magnetization $M(t)$ of the system is defined as

$$M(t) = \frac{1}{N} \sum_{i=1}^N s_i^{(t)}. \quad (2.3)$$

We can infer the state of the logical bit from the instantaneous magnetization $M(t)$ or from the coarse-grained magnetization, i.e., a simple moving average of the instantaneous magnetization; a positive value indicates logical state 1, whereas a negative value indicates logical state 0. In Sec. 2.2.1, we discuss the evolution of the instantaneous magnetization and how coarse-graining can remove fluctuations in the instantaneous magnetization, making the coarse-grained readout a more reliable method of inferring the logical bit.

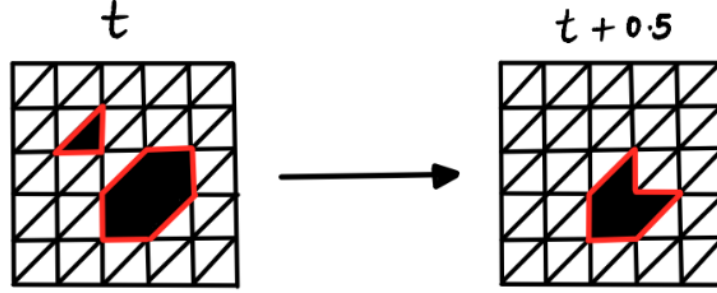


Figure 2.5: An illustration of the synchronous deterministic update which contracts the domain wall (red edges) in the sweep direction.

The evolution of this PCA simulates error correction in the system (see Fig. 2.4), where the deterministic rule may be viewed as a protocol that attempts to remove noise, i.e., random spin-flips introduced by the probabilistic rule, by contracting the domain wall in the sweep direction (see Fig. 2.5). From this perspective,

$$T = p + q, \quad h = \frac{p - q}{p + q} \quad (2.4)$$

are analogous to temperature and magnetic field [52]; however, for clarity, we will refer to h as bias. By definition, $T \in [0, 2]$, where $T = 1$ is analogous to the infinite temperature limit in thermodynamics; $T \in [0, 1)$ and $T \in (1, 2]$ are analogous to the positive and negative temperatures, respectively. By definition, $h \in [-1, 1]$. The non-zero bias h indicates whether one of the two possible spin-flips described in Eqs. (2.1)-(2.2) is more likely than the other. In particular, at $h = -1$ the probabilistic update only affects $+1$ spins, whereas only -1 spins are affected at $h = +1$.

We define the *memory lifetime* of the PCA as the time it takes on average for a logical bit-flip in the system with dynamics described by the PCA. The memory lifetime $\tau_{L,T}$ of the system of linear lattice size L at temperature T can be estimated as follows

$$\tau_{L,T} = \frac{1}{n} \sum_{j=0}^n t_j, \quad (2.5)$$

where t_j is the time it takes for the first logical bit-flip to occur in the j^{th} simulation of the PCA for $j \in \{1, 2, \dots, n\}$. Our goal is to identify a region in the (T, h) plane, where the memory lifetime of the system increases exponentially with system size, i.e., the system serves as a good memory. The boundary of this region tells us the maximum temperature (which we refer to as the critical temperature T_c^h) that can be tolerated at a given value of bias h .

In the context of QECC, it is particularly useful to consider the action of the update rules on the domain wall (shrinking domain walls along a sweep direction) rather than on the individual spins. This shift in perspective from the individual spins in the lattice to the domain wall is important because syndrome measurements in topological codes only provide access to the excitations, i.e., the local domain wall configuration, produced by errors without revealing the exact location and weight of the errors causing them. The memory of the sweep rule PCA is directly related to the sweep rule's ability as a QECC decoder to successfully apply corrections that shrink loop-like excitations without incurring a logical error in the encoded qubit. The temperature T is an indicator of the average number and size of loop-like excitations in the lattice, while the bias h indicates whether syndrome measurement errors that misreport measurement outcome $+1$ as -1 are more likely than the reverse. The region in the (T, h) plane where the PCA behaves like a good classical memory demarcates the regime where the sweep rule can be a good decoder for topological codes. Most notably, in the following sections we show that the sweep rule PCA exhibits memory even when $h \neq 0$ making it robust to bias in syndrome measurement errors. These important connections make the aforementioned classical memory model a simple and interesting system to investigate the mechanisms that make sweep rule applicable to QEC.

2.2.1 Simulating the sweep rule

In this section, we study how a logical bit encoded in the PCA evolves to estimate the memory lifetime of the system. Since we infer the state of the logical bit from the instantaneous magnetization $M(t)$, we numerically analyze the behavior of $M(t)$ and its moving average over time.

Evolution of the instantaneous magnetization

First, we study the system with no bias to identify the critical temperature T_c^0 . In this case, the spin-flip noise is symmetric with $p = q = T/2$. Thus, without loss of generality, we focus on the evolution of the logical bit 1 encoded in the initial spin configuration $\forall i : s_i^{(0)} = +1$.

Over the course of simulation, we observe the magnetization of the system and detect three different kinds of behavior depending on the temperature T and its relation to the critical temperature T_c^0 (see Fig. 2.6).

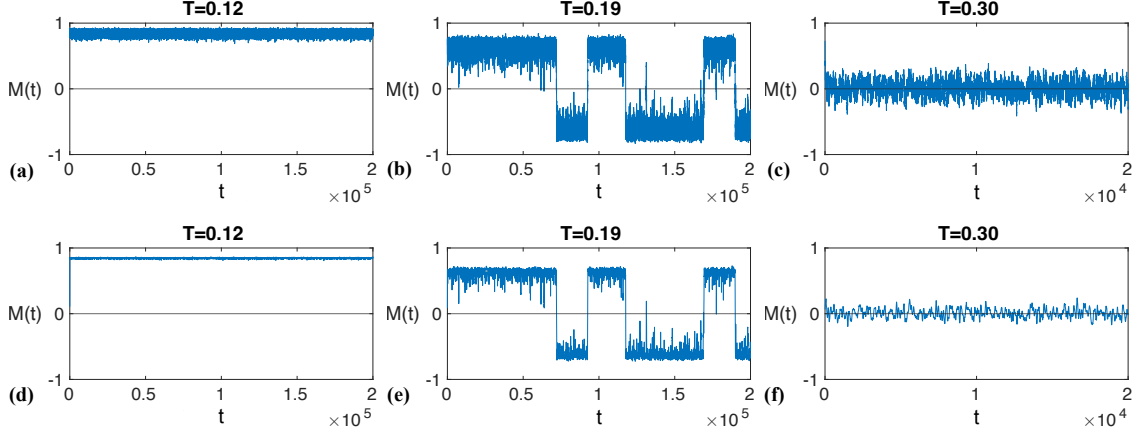


Figure 2.6: Evolution of magnetization of the PCA with $N = 512$ spins on a 16×16 triangular lattice for $h = 0$ at: (a)(d) low, (b)(e) moderate and (c)(f) high temperatures. The plots in (d)-(f) depict magnetization after it is smoothened (see Appendix A.1 for details). Logical bit-flips are noted as occurring at time steps when the value of smoothened magnetization is zero.

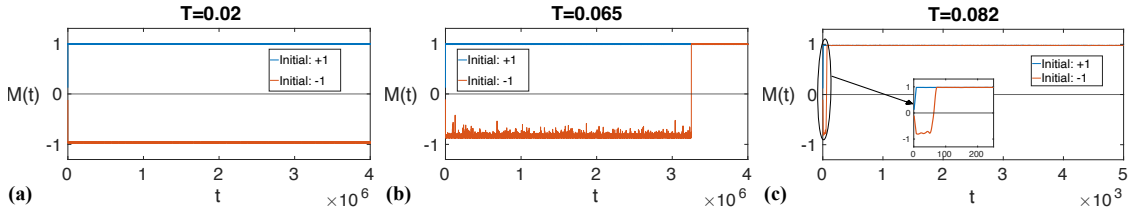


Figure 2.7: Evolution of smoothened magnetization of the PCA with $N = 512$ spins on a 16×16 triangular lattice for $h = 0.75$ at: (a) low, (b) moderate and (c) high temperatures. Since $h > 0$, the memory lifetime is limited by the lifetime of the logical state 0 (orange lower curve) that was initially encoded with -1 spins; the logical state 1 (blue upper curve) does not undergo logical bit-flips.

At low temperatures, i.e., $T < T_c^0$, the magnetization of the system exhibits fluctuations around a value $M_{\text{low}} \approx 1 - T$ (see Fig. 2.6(a)) with standard deviation scaling linearly with T . These fluctuations can be attributed to the addition of random spin-flips at each time step. At low temperatures, the probabilistic update sparsely introduces flipped spins

(see Fig. 2.8(a)) that are removed by the deterministic rule in the next few time steps ensuring that the value of magnetization remains close to M_{low} for the entire duration of the simulation. Thus, at low temperatures, we observe that for a long time we are able to correctly infer the logical bit-state.

At moderate temperatures, i.e., $T \lesssim T_c^0$, at the beginning, the magnetization of the system fluctuates around some positive value $M_{\text{mod}} < 1 - T$. The variance of the magnetization is greater than in the low temperature regime and diverges at the critical temperature T_c as indicated by the plot of magnetic susceptibility χ (defined in Eq. (A.5)) in Fig. 2.11(b). In this temperature range, at each time step, the probabilistic update creates clusters of flipped spins that are likely to be larger than those in the low-temperature regime (see Fig. 2.8(b)). The larger the cluster of flipped spins, the more time steps (and applications of the deterministic rule implementing error correction) are required to completely remove it from the system. Large clusters have the potential to eventually coalesce and cause the majority of the spins to flip leading to a logical bit-flip. Thus, unlike in the low temperature range, where the value of magnetization remains close to M_{low} over the whole duration of the simulation, in this case, the magnetization changes from a positive value M_{mod} to a negative value $-M_{\text{mod}}$ (see Fig. 2.6(b)), indicating a logical bit-flip. Over the course of the simulation, the magnetization continues to flip between M_{mid} and $-M_{\text{mid}}$.

At high temperatures, i.e., $T > T_c^0$, the magnetization fluctuates about $M_{\text{high}} = 0$ over the duration of the entire simulation. The variance of the magnetization in this temperature range is smaller than in the moderate temperature regime (see Fig. 2.11(b)). The system appears to lose the initially encoded state of the logical bit within a very short time (compared to the system size N).

The plots in Fig. 2.6(a)-(c) show that fluctuations in magnetization occur at all three temperature regimes due to the random physical spin-flips introduced by the probabilistic update. At moderate and high temperatures, these fluctuations are likely to build up within a short time interval (proportional to the system size N) and cause logical bit-flips, while at low temperatures these fluctuations disappear and do not affect the logical bit state. We thus choose to smoothen the plots of instantaneous magnetization (see Fig. 2.6(d)-(f)) by first coarse-graining it and then calculating its moving average (see Appendix A.1 for details).

For the PCA with non-zero bias, the spin-flip noise is not symmetric. In particular, for $h > 0$ the memory lifetime of the system is limited by the memory lifetime of the logical state 0. As in the case of $h = 0$, we smoothen the magnetization (see Fig. 2.7) to remove fluctuations that do not result in logical bit-flips.

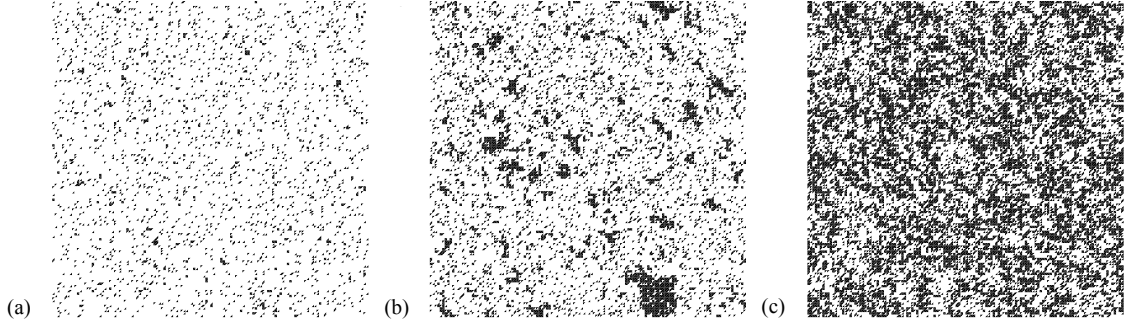


Figure 2.8: Snapshots of spin configurations on a 128×128 triangular lattice with $N = 32768$ spins at $h = 0$. (a) At low temperature $T = 0.12$, small clusters of -1 spins (black) are isolated from each other. (b) At $T = 0.19$, large clusters of -1 spins of arbitrary size appear, indicating critical behavior. As the system evolves, these clusters coalesce and cause logical bit-flips. (c) At high temperature $T = 0.30$, the spin configuration is disordered, with roughly equal numbers of $+1$ and -1 spins.

2.3 Numerical analysis of memory lifetime of sweep rule

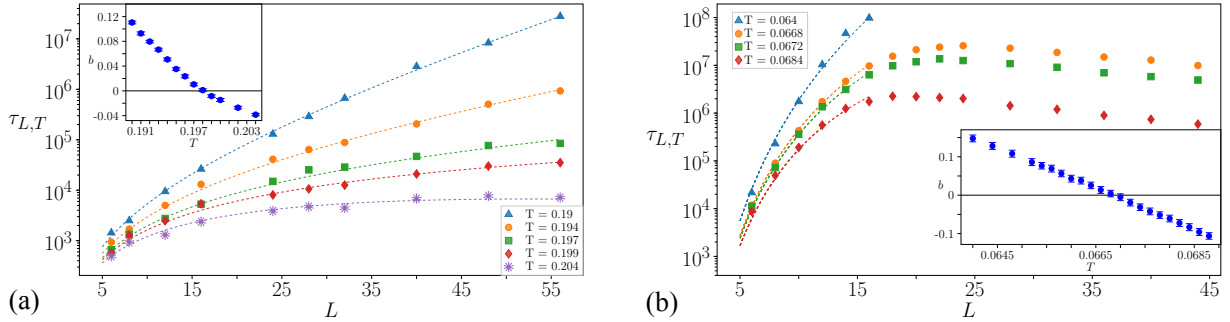


Figure 2.9: Plots of memory lifetime $\tau_{L,T}$ of the sweep rule as a function of the linear lattice size L at various temperatures T . This data is fit into the ansatz in Eq. (2.6). (a) At $h = 0$, we identify the critical temperature $T_c^0 = 0.197 \pm 0.002$ as the point where $b = 0$ (inset). The system behaves as a good memory below T_c^0 . (b) At $h = 0.75$, we similarly identify the critical temperature $T_c^{0.75} = 0.0670 \pm 0.0004$. Additionally, for $T > T_c^{0.75}$ we observe that $\tau_{L,T}$ increases with L as long as $L < L_T^*$, where L_T^* is the linear lattice size maximizing $\tau_{L,T}$; further increase of L decreases $\tau_{L,T}$.

We estimate the memory lifetime at various values of temperature T and bias h (see

Appendix A.1 for detailed data) and analyze how it scales with the linear lattice size L . We plot $\tau_{L,T}$ as a function of L for various T (see Fig. 2.9). We fit this data in the numerical ansatz

$$\log \tau_{L,T} = a \log L + bL + c, \quad (2.6)$$

where a , b and c are fitting parameters. The parameter b captures the exponential scaling of memory lifetime with L . For $T \approx T_c^h$, we observe that the parameters a, c do not exhibit a strong dependence on temperature. Therefore, we constrain a, c to be fixed over the temperature range (see Fig. A.1 in Appendix A.1) and report the behavior of b as a function of T (see Fig. A.3 in Appendix A.1).

For $h = 0$, we observe that at low temperatures, the memory lifetime scales exponentially with linear lattice size L as indicated by the parameter b being positive (see inset of Fig. 2.9(a)). At $T \approx 0.197$, this exponential scaling of memory lifetime with L is lost ($b = 0$) and the memory lifetime $\tau_{L,T}$ scales polynomially in L , i.e., $\tau_{L,T} \propto L^a$ with $a \approx 2$. Thus, we estimate the critical temperature to be $T_c^0 = 0.197 \pm 0.002$. Similarly, we perform the numerical analysis of memory lifetime of the PCA for multiple values of h . For instance, when $h = 0.75$, the parameter b (see inset of Fig. 2.9(b)) intersects the x -axis at $T_c^{0.75} = 0.0670 \pm 0.0004$ and the memory lifetime $\tau_{L,T}$ scales as L^a , where $a \approx 7$, indicating a phase transition. Subsequently, we obtain the phase transition points T_c^h in each case (see Figs. A.2 and A.3 in Appendix A.1), thus recovering the phase diagram shown in Fig. 2.10.

2.4 Statistical-mechanical analysis of memory phase transition of sweep rule

In this section, we use standard statistical-mechanical techniques (see Fig. 2.11) to identify a phase transition in the sweep rule with no bias. In particular, we study the scaling of the (ensemble) averages of the system's magnetization M , magnetic susceptibility χ , and two-point correlation function ξ (see Appendix A.2 for details). In the neighborhood of T_c^0 , this phase transition is characterized by critical exponents ν , β and γ corresponding to the following scaling relations [53]

$$\widetilde{\xi}(L^{\frac{1}{\nu}}|\varepsilon|) = L^{-1}\langle\xi\rangle, \quad (2.7)$$

$$\widetilde{\chi}(L^{\frac{1}{\nu}}|\varepsilon|) = L^{-\frac{\gamma}{\nu}}\langle\chi\rangle, \quad (2.8)$$

$$\widetilde{M}(L^{\frac{1}{\nu}}|\varepsilon|) = L^{\frac{\beta}{\nu}}\langle M\rangle, \quad (2.9)$$

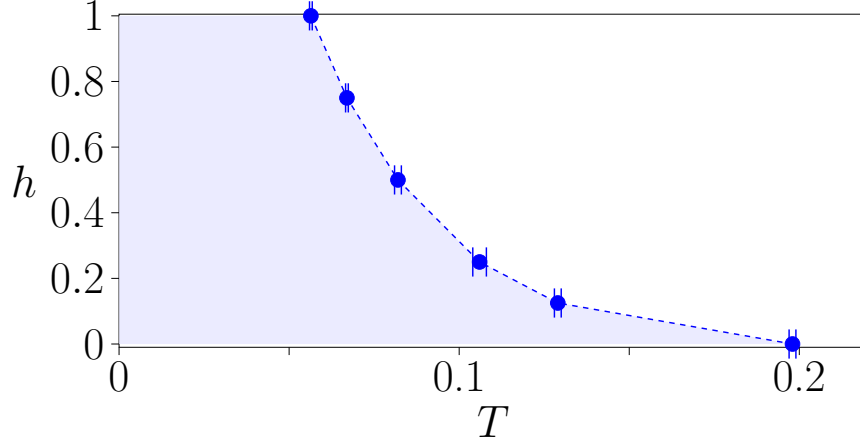


Figure 2.10: Phase diagram of the sweep rule with noise parameters $p, q \in [0, 1]$ in the (T, h) plane, where $T = p + q$ and $h = (p - q)/(p + q)$ can be interpreted as temperature and bias. The shaded region corresponds to a region, where two stable phases coexist and the sweep rule can work as memory. The diagram is symmetric with respect to the $h = 0$ axis.

where $\varepsilon = (T - T_c^0)/T_c^0$ is the reduced temperature of the system. The numerical data for different linear lattice sizes L is collapsed onto scaling functions $\tilde{\xi}$, $\tilde{M}$, and $\tilde{\chi}$, which are independent of the lattice size L (see Fig. 2.11(d)-(f)). This method yields the critical exponents $\beta = 0.122 \pm 0.007$, $\gamma = 1.63 \pm 0.02$ and $\nu = 0.93 \pm 0.01$ with $T_c = 0.197 \pm 0.001$. These values of critical exponents are also found to satisfy the hyperscaling relation [50, 53]

$$2\beta + \gamma = \nu d, \quad (2.10)$$

where $d = 2$ since we are studying 2D lattices. Moreover, the ratios $\gamma/\nu = 1.75 \pm 0.04$ and $\beta/\nu = 0.13 \pm 0.01$ obtained are equal to the corresponding ratios for the 2D Ising model indicating that the sweep rule belongs to the *weak universality class* [54] of the Ising model. This is also in agreement with the universality class of Toom's rule [52, 55, 56] (which the sweep rule generalizes) reported in Ref. [50]. The statistical-mechanical analysis of the sweep rule independently identifies a second-order phase transition at $T_c^0 = 0.1973 \pm 0.0001$ from ferromagnetic behavior at lower temperatures to paramagnetic behavior at higher temperatures (see Fig. 2.11(a)-(c)). In the absence of bias, i.e., $h = 0$, the ferromagnetic behavior of a system permits the stability of two different phases corresponding to average magnetization $+M$ and $-M$, which can be used to encode the two states of a logical bit. Such ferromagnetic behavior is an indicator that the system behaves as a good memory as in the case of the 2D Ising model at $h = 0$. The use of two independent techniques,

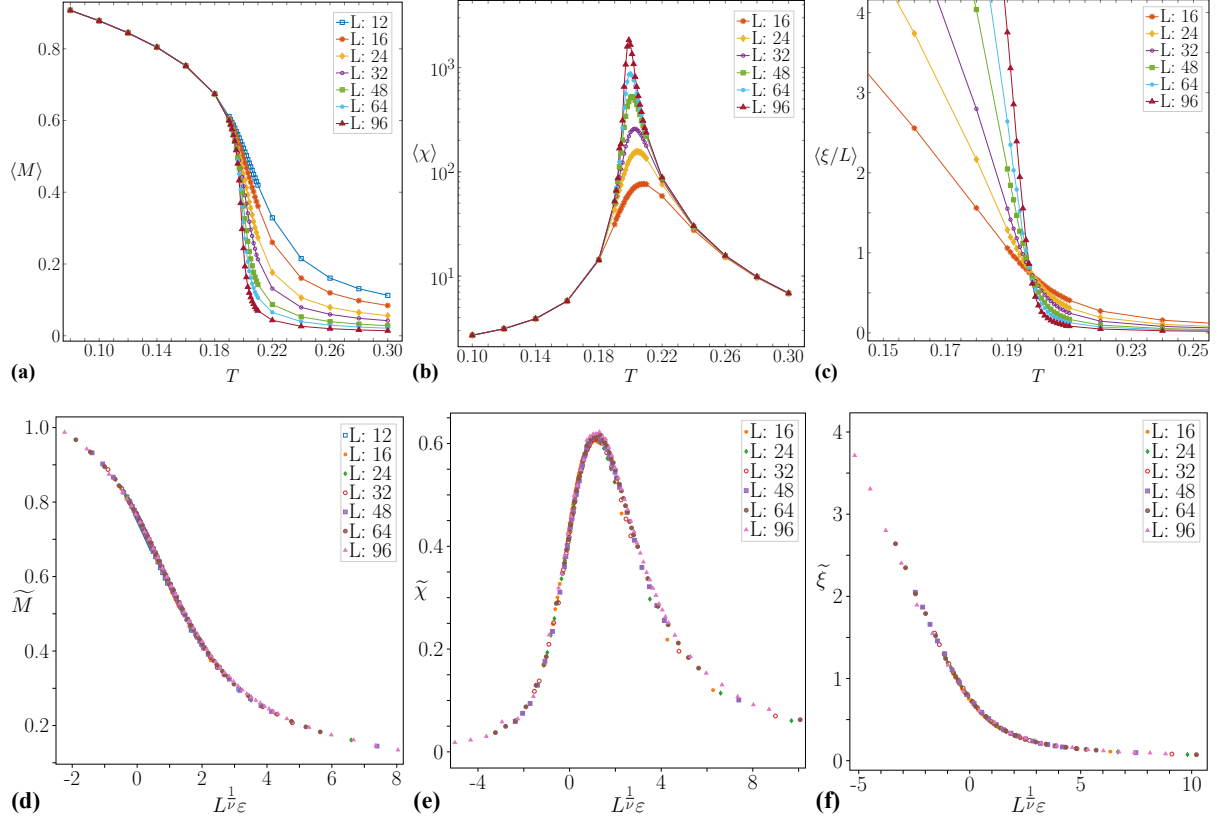


Figure 2.11: Critical behavior of the sweep rule at $h = 0$. (a) The ensemble average of magnetization $\langle M \rangle$ drops continuously from a non-zero value to zero at the critical temperature, indicating a second-order phase transition. (b) The ensemble average of the magnetic susceptibility $\langle \chi \rangle$ captures the variance of the magnetization, which diverges at the critical temperature. (c) We can identify the critical temperature from the intersection of the plots of the ensemble average of two-point correlation functions $\langle \xi/L \rangle$ for different L . (d)-(f) The data collapse of $\langle M \rangle$, $\langle \chi \rangle$ and $\langle \xi/L \rangle$ into L -independent functions $\tilde{M}$, $\tilde{\chi}$ and $\tilde{\xi}$ respectively, is used to obtain the critical exponents β , γ , ν (see Table A.1) with critical temperature $T_c^0 = 0.1973 \pm 0.0001$.

numerical analysis of memory lifetime scaling and statistical mechanical analysis of the sweep rule spin model, to identify the same phase transition suggests that the scaling of memory lifetime with linear lattice size is a good indicator of the sweep rule's phase transition.

2.5 Analytical model of sweep rule

In this section, following Ref. [57] we discuss a simple analytical model of the PCA at $h = 0$. We note that in the infinite temperature limit, i.e., $T = 1$, the probabilistic update flips each spin with probability $p = q = 0.5$, i.e., each spin has a 50% chance of being flipped irrespective of the outcome of the deterministic update. Thus, we can treat the system as a collection of N independent spins with a mean flipping rate $\lambda = 0.5$ per time step. At $T \lesssim 1$, the probability of spin-flips in one time step is less than 0.5 and depends on the outcome of the deterministic rule, which in turn depends on the state of the neighboring spins. As temperature decreases, the effects of the deterministic update compete with those of the probabilistic update. Nevertheless, for sufficiently high temperatures, the system is still predominantly affected by random spin-flips and the system can be modelled by a smaller collection of $N_{\text{eff}} < N$ effective spins that independently flip at some mean flipping rate $\lambda < 0.5$ per time step. Since we only take into account the effects of random spin-flips in this simple model, we can expect that the model will break down for temperatures close to the critical temperature, where the effect of the deterministic updates cannot be neglected.

Using the same reasoning as presented in Ref. [57], the fidelity $F(t)$ of the logical bit encoded in the PCA at time step t can be written as

$$F(t) = \frac{1}{2} \left(1 + \text{erf} \left(\frac{N_{\text{eff}} - 2\mu}{2\sqrt{2}\sigma} \right) \right), \quad (2.11)$$

$$\mu = \frac{1}{2} N_{\text{eff}} (1 - e^{-2\lambda t}), \quad (2.12)$$

$$\sigma = \frac{1}{2} [N_{\text{eff}} (1 - e^{-2\lambda t}) (1 + e^{-2\lambda t})]^{1/2}. \quad (2.13)$$

Here, $F(t)$ is the probability that at time step t a logical bit-flip does not occur in the system, i.e., fewer than half of the spins in the system have flipped an odd number of times since initialization, $\text{erf}(\cdot)$ denotes the error function, and N_{eff}, λ are the two parameters of the model that vary with temperature. As per this model, $F(t)$ approximates a binomial distribution with mean μ and standard deviation σ . By definition, the fidelity at initializa-

tion is $F(0) = 1$. As time progresses, the fidelity is expected to decrease from this initial value and $\lim_{t \rightarrow \infty} F(t) = 0.5$.

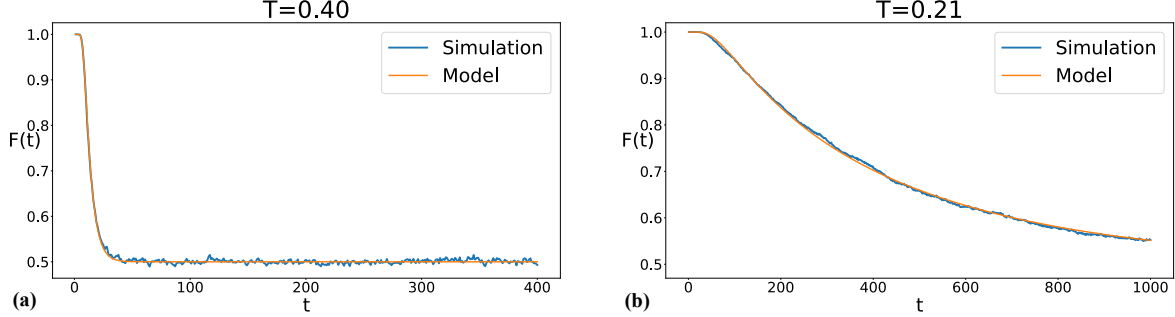


Figure 2.12: The plots of fidelity $F(t)$ of the logical bit encoded in the PCA at $h = 0$. The width of the peak in the $F(t)$ plot captures the memory lifetime of the PCA at that temperature. (a) At $T = 0.40$, the fidelity estimated from simulation agrees very well with the proposed model in Eq. (2.11). For a system with $N = 200$ spins, we obtain parameters $N_{\text{eff}} = 38.7 \pm 1.2$, $\lambda = 0.0884 \pm 0.0001$ from the model, with goodness of fit parameters $\chi^2 = 335$ and degrees of freedom $\text{DOF} = 398$. (b) For the same system at a lower temperature $T = 0.21$, we obtain the parameters to be $N_{\text{eff}} = 1.373 \pm 0.006$, $\lambda = (1.105 \pm 0.003) \times 10^{-3}$ with $\chi^2 = 1181$, $\text{DOF} = 998$. The ratio χ^2/DOF is found to increase as T is lowered, indicating that the model best captures the behavior of the system at high temperatures [57].

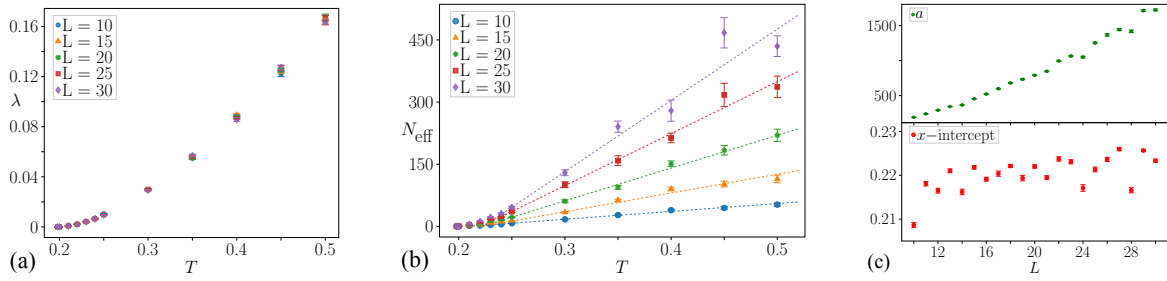


Figure 2.13: (a) The plot of the mean flipping rate λ as a function of T . λ does not change noticeably with L . (b) The plot of the effective number of spins N_{eff} as a function of T . In the range $T \in [0.25, 0.5]$, this parameter scales linearly with temperature, as captured by Eq. (2.14). (c) We plot the slope a and the x -intercept of this linear fit as a function of lattice size L . The x -intercept gives the temperature at which $N_{\text{eff}} = 0$, subsequently indicating the breakdown of the model.

We test the validity of the model in Eq. (2.11) by numerically estimating the fidelity

of the system as a function of time and checking the goodness of fit of the model. We simulate the PCA initialized with all spins in state +1 for $n = 10^4$ trials and at each time step t , we estimate $F(t)$ as the fraction of the trials for which $M(t) > 0$. We observe that in the high temperature range, i.e., $T \in [0.25, 1]$, the model is a good description of the system. For instance at $T = 0.4$ (see Fig. 2.12(a)), the goodness of fit ratio indicates agreement between numerical data and Eq. (2.11). However, as the temperature is lowered, the quality of this fit diminishes. In particular, at $T = 0.21$, the goodness of fit ratio (see Fig. 2.12(b)) suggests that the model starts to break down.

In Fig. 2.13, we plot the parameters λ and N_{eff} against temperature T for different lattice sizes L . We observe that λ is predominantly dependent on T and does not change noticeably with L . In the range $T \in [0.25, 0.5]$, the parameter N_{eff} can be approximated with the following ansatz

$$N_{\text{eff}} = aT + b, \quad (2.14)$$

where a, b are fitting parameters. The slope a scales linearly with L and by finding the x -intercept of the numerical ansatz in Eq. (2.14) we can identify that our simple analytical model breaks down at $T \approx 0.22$, as N_{eff} (that corresponds to the effective number of independent spins) becomes zero (see Fig. 2.13(c)). This is indicative of a phase transition in the system and corroborates our numerical findings at $h = 0$.

2.6 Discussion

In this chapter, we studied a classical spin system evolving under the sweep rule, and used numerical simulations to find the region in the (T, h) plane, where its memory lifetime scales exponentially with linear lattice size. For $h = 0$, we supported our numerical analysis by an additional statistical-mechanical study of the sweep rule and a simple analytical model that captures the memory lifetime at high temperatures. We also found that the sweep rule belongs to the weak 2D Ising universality class. Most notably, we found that unlike the Glauber CA [50] (Ising model [58–61]) that also removes noise in a system by contracting domain walls, the sweep rule behaves as a good memory even in the presence of bias.

In the main text, we focused on the synchronous sweep rule on triangular lattices. However, our results do not change qualitatively if we consider square lattices instead (see Appendix A.1.4). In fact, the sweep rule on square lattices is identical to Toom’s rule. From this perspective, our results recover and are consistent with the previous works on Toom’s rule [50, 52, 55]. We also remark that our findings are largely the same for the asynchronous sweep rule, with the main difference being that it belongs to the Ising universality class (see Appendix A.3 for details).

PCAs such as the sweep rule and Toom’s rule are at the heart of many decoding algorithms [28, 33, 46, 62–65] for quantum error correcting codes. This is due to their ability to protect encoded information in the presence of bias, which guarantees robustness to measurement errors that are unavoidable in quantum error correction protocols. Studying classical spin models and their phase transitions gives insights into the performance of quantum error correcting codes and their optimal decoding algorithms [31, 33, 66–69]. Such studies of PCAs are motivated by the hope that they will bring us closer to realizing (possibly self-correcting) quantum memories [70, 71].

Chapter 3

Fault-tolerant logical operations in QECC

Quantum error correcting codes (QECC) can be used to encode k logical qubits into a system of $n > k$ physical qubits so that the probability of errors on the logical qubit is lower than that on the physical qubits. In this way, QECCs offer a way to prolong the coherence of quantum information. Another crucial aspect of QECCs is the ability to perform quantum computation with the encoded information using encoded logical operations (see Sec. 1.3). As discussed in Sec. 1.4, QEC is only feasible over the course of quantum computation if errors in the system are prevented from propagating in an uncontrolled fashion. While transversal logical operations are a simple and popular approach to fault-tolerance, the Eastin-Knill theorem [37] proves that no QECC can have a universal set of transversal logical gates. In many stabilizer codes, Clifford group operations can be easy to implement transversally. But non-Clifford operations, a crucial component of universal gate sets, are generally more challenging to implement fault-tolerantly in QECCs, often requiring more complex procedures such as magic state distillation [72, 73].

There exists a family of codes known as triorthogonal codes [74] with transversal non-Clifford gates, implemented by tensor products of single-qubit $T = \text{diag}(1, \exp(i\pi/4))$ gates. The $[[8,3,2]]$ code [75, 76] is the smallest known code in this family having a transversal entangling non-Clifford gate—the $\text{CCZ} = \text{diag}(1, 1, 1, 1, 1, 1, 1, -1)$ gate. From a fault-tolerance perspective, it is particularly desirable to implement complex entangling gates transversally using single-qubit gates, as single-qubit gates on most hardware platforms [35, 36, 77–80] are often less noisy than entangling gates by at least an order of magnitude. Using small codes to demonstrate fault-tolerant Clifford and non-Clifford operations has previously been suggested [81] and implemented in NMR [82, 83], trapped-ion [84–87], and

superconducting hardware [88–90].

In this chapter, we investigate the performance of the encoded gates of the $[[8,3,2]]$ code on superconducting and trapped-ion hardware platforms. We compare the performance of the encoded gates with the same gates executed with no encoding, finding that the encoded gates perform better than their bare (non-encoded) counterparts in every case where the encoded gate is non-Clifford, even though the encoded circuits contain more entangling gates than the bare circuits. Notably, we observe improved performance for the CCZ gate, which is the dominant gate in circuits such as adders [91, 92] and the modular exponentiation used in Shor’s algorithm [93, 94]. In Sec. 3.1, we review the definition of the $[[8,3,2]]$ code and its transversal logical gates. In Sec. 3.2, we give fault-tolerant circuits for preparing encoded states of the $[[8,3,2]]$ code and for logical measurements. In Sec. 3.3, we describe our demonstrations on quantum hardware and their results, and we conclude with Sec. 3.4.

3.1 The $[[8,3,2]]$ color code

The $[[8,3,2]]$ color code is a stabilizer code [95], encoding 3 logical qubits into 8 physical qubits with distance 2 (meaning that it can detect any single-qubit error). It is convenient to define the code using a geometric representation, where the physical qubits reside at the vertices of a cube, as shown in Fig. 3.1. The stabilizer group is generated by an X -type operator acting on all the qubits, and by Z -type operators associated with the faces of the cube. Concretely, using the qubit indices in Fig. 3.1, the stabilizer group is

$$\mathcal{S} = \langle X^{\otimes 8}, Z_0 Z_1 Z_2 Z_3, Z_4 Z_5 Z_6 Z_7, \\ Z_0 Z_1 Z_4 Z_5, Z_0 Z_2 Z_4 Z_6 \rangle, \quad (3.1)$$

where Z_i denotes a Pauli Z operator acting on qubit i etc. We note that the stabilizer generators in Fig. 3.1 are either X -type or Z -type, meaning that the $[[8,3,2]]$ code is a CSS code [8, 24].

The logical operators of the $[[8,3,2]]$ code also have a geometric interpretation. Logical X operators are associated with the faces of the cube, and logical Z operators with the edges of the cube. We can choose the following basis of logical Pauli operators

$$\begin{aligned} \overline{X}_1 &= X_0 X_1 X_2 X_3, & \overline{Z}_1 &= Z_0 Z_4, \\ \overline{X}_2 &= X_0 X_1 X_4 X_5, & \overline{Z}_2 &= Z_0 Z_2, \\ \overline{X}_3 &= X_0 X_2 X_4 X_6, & \overline{Z}_3 &= Z_0 Z_1, \end{aligned} \quad (3.2)$$

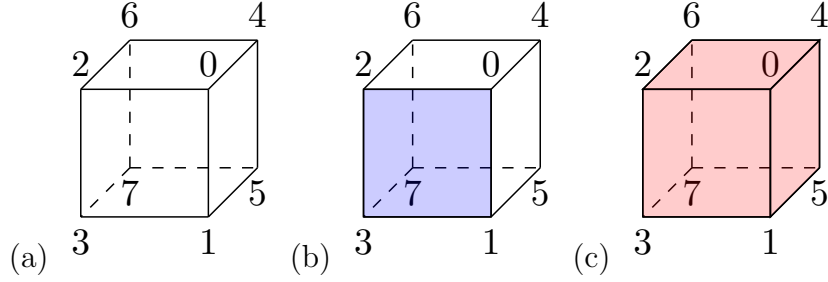


Figure 3.1: Geometric representation of the $[[8, 3, 2]]$ code. (a) The physical qubits reside at the vertices of the cube. (b) Z -type stabilizers are associated with faces, for example the blue face has an associated stabilizer $Z_0Z_1Z_2Z_3$. (c) The X -type stabilizer acts on all the qubits.

where we use overlines to distinguish operators acting on the logical qubits from operators acting on the physical qubits.

The $[[8, 3, 2]]$ code is notable for having a non-Clifford transversal gate, CCZ implemented by T and $T^\dagger$ gates. Specifically,

$$\overline{\text{CCZ}} = T_0T_1^\dagger T_2^\dagger T_3^\dagger T_4^\dagger T_5T_6T_7^\dagger. \quad (3.3)$$

This gate again has a geometric interpretation: vertices and edges of the cube form a bipartite graph and CCZ is implemented by applying T to (the qubits on) one set of the vertices and $T^\dagger$ to the other. The transversality of CCZ and Pauli X imply that the $[[8, 3, 2]]$ code also has transversal $\text{CZ} = \text{diag}(1, 1, 1, -1)$ gates, as follows

$$\begin{aligned} \overline{\text{CZ}}_{12} &= S_0S_2^\dagger S_4^\dagger S_6, \\ \overline{\text{CZ}}_{13} &= S_0S_1^\dagger S_4^\dagger S_5, \\ \overline{\text{CZ}}_{23} &= S_0S_1^\dagger S_2^\dagger S_3, \end{aligned} \quad (3.4)$$

where $S = T^2$ and CZ_{ij} acts on logical qubits i and j .

3.2 Fault-tolerant circuits

For an error-detecting code such as the $[[8, 3, 2]]$ code, we say that a circuit is fault-tolerant if any single-qubit error on the input state or an error at any single location in the circuit can at worst lead to a detectable error on the output state. A circuit location can be a

state preparation, gate, or measurement. We need only consider Pauli errors due to error discretization [96]. And we note that as the $[[8,3,2]]$ code is a CSS code, it is sufficient to analyse X and Z errors independently. We remark that the logical CCZ and CZ gates discussed in Sec. 3.1 are transversal and are therefore trivially fault-tolerant. We also need fault-tolerant circuits for logical measurement and logical state preparation, and we now discuss each of these in turn.

As the $[[8,3,2]]$ code is a CSS code, we can do a fault-tolerant measurement of the logical qubits in the X or Z basis by measuring all of the physical qubits in the X or Z basis, respectively, and processing the classical outcomes [96]. In the case of an error-detecting code such as the $[[8,3,2]]$ code, the classical processing is especially simple: we simply discard any measurement result that corresponds to a state that is not a $+1$ eigenvalue of the stabilizers. For example, when measuring in the X basis we accept any result whose parity is even, i.e., a $+1$ eigenstate of $X^{\otimes 8}$. This is fault-tolerant because single-qubit errors before the measurements are detectable by definition, and any single measurement error is equivalent to a single-qubit error before the measurement.

3.2.1 GHZ state preparation

First we consider a fault-tolerant circuit for preparing the logical GHZ state, $|\text{GHZ}\rangle = (|000\rangle + |111\rangle)/\sqrt{2}$. Our circuit (shown in Fig. 3.2) factorizes into two independent and identical sub-circuits acting on qubits 0, 3, 5, 6 and qubits 1, 2, 4, 7 (the two bipartite sets discussed in Sec. 3.1). The $[[8,3,2]]$ code can detect any weight ≤ 3 X error and so we only need to consider the four-qubit errors $X_0X_3X_5X_6$ and $X_1X_2X_4X_7$. However, each of these errors is in fact a logical $\bar{X}_1\bar{X}_2\bar{X}_3$ operator and so leaves the target $|\text{GHZ}\rangle$ state invariant. The only possible Z errors are weight one (detectable) and weight two (non-detectable). However, one can verify that all the non-detectable errors have trivial action on the target $|\text{GHZ}\rangle$ state. For example, the first CNOT could fail giving a Z_1Z_2 error, but this implements a logical $\bar{Z}_2\bar{Z}_3$ operator (see Eq. 3.2) and hence leaves the target $|\text{GHZ}\rangle$ state invariant.

3.2.2 $|+++ \rangle$ state preparation

Next, we provide a fault-tolerant circuit for preparing the $|+++ \rangle$ state, shown in Fig. 3.3. In this circuit, the potentially problematic errors are those that can propagate through the CNOT gates. Consider, for example, the CNOT gates with qubit 0 as the control. The

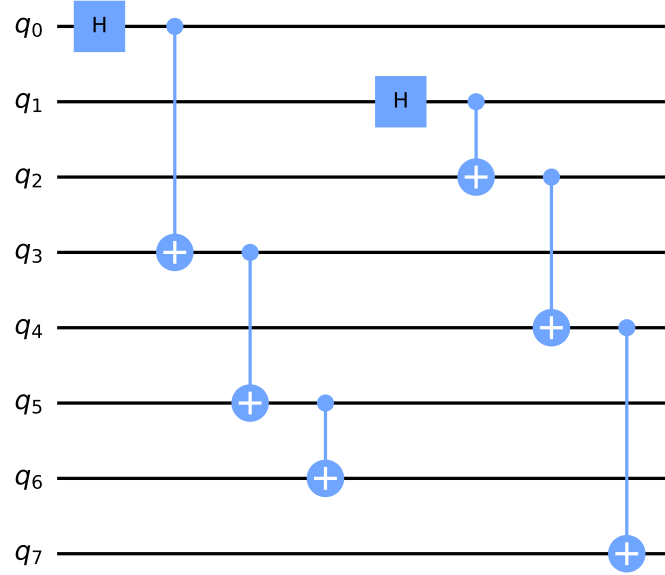


Figure 3.2: Fault-tolerant circuit for preparing the $|\text{GHZ}\rangle$ state in the $[[8,3,2]]$ code.

possible multi-qubit X errors that can arise from these gates are

$$\begin{aligned}
 X_0 X_3 & \text{ (detectable),} \\
 X_0 X_2 X_3 & \text{ (detectable),} \\
 X_0 X_1 X_2 X_3 & (\bar{X}_1),
 \end{aligned} \tag{3.5}$$

where the only non-detectable error has trivial action on the target encoded state. The same is true for the other groups of CNOT gates with the same target. Certain Z errors can also propagate through CNOT gates. For example, consider the CNOT gates with qubit 1 as the target. The possible multi-qubit Z errors that can arise from these gates are

$$\begin{aligned}
 Z_1 Z_{a_0} & \text{ (detectable),} \\
 Z_1 Z_7 Z_{a_0} & \text{ (detectable),} \\
 Z_1 Z_6 Z_7 & \text{ (detectable),} \\
 Z_1 Z_6 Z_7 Z_{a_0} & \text{ (detectable),} \\
 Z_0 Z_1 Z_6 Z_7 & \text{ (stabilizer).}
 \end{aligned} \tag{3.6}$$

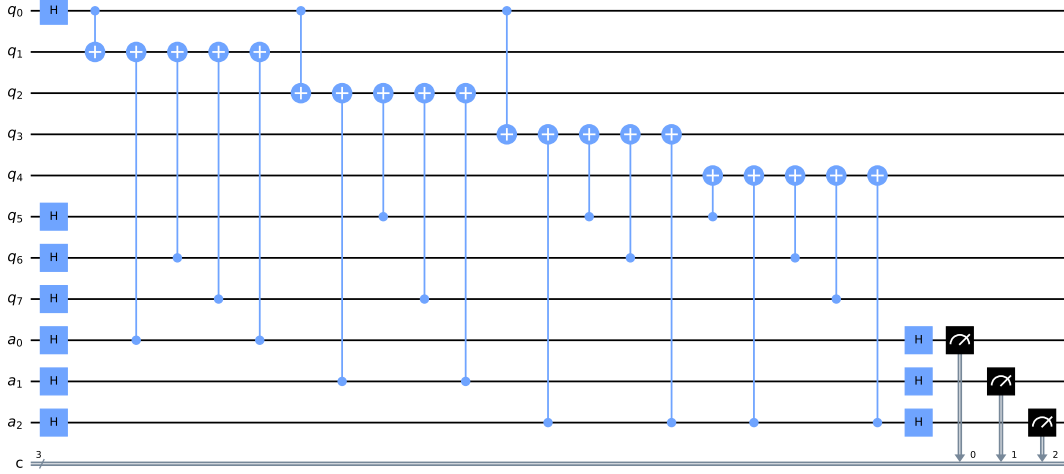


Figure 3.3: Fault-tolerant circuit for preparing the state $|+++ \rangle$ in the $[[8,3,2]]$ code. The qubits a_0 , a_1 and a_2 are flag qubits whose purpose is to detect certain Z errors that could cause logical errors. If we measure the three flag qubits (into the classical register c indicated by a double line) to be in the $|0\rangle$ state then we accept the output.

The purpose of the flag qubit [97], a_0 , is to make the error $Z_1 Z_7 = \bar{Z}_1 \bar{Z}_2$ detectable. Similarly, the flag qubits a_1 and a_2 catch the errors $Z_2 Z_7$, $Z_3 Z_6$ and $Z_4 Z_6$.

3.3 Demonstration results

We investigate the performance of circuits comprised of three parts: state preparation, a transversal logical gate, and logical measurement.

For the state preparation part, we consider either $|\text{GHZ}\rangle$ or $|+++ \rangle$ state preparation, using the circuits described in Sec. 3.2. For the logical gate part, we consider each of the distinct products of the transversal logical CCZ, CZ₁₂, CZ₀₂ and CZ₀₁ gates available in the $[[8,3,2]]$ code along with the logical identity gate implemented as a ‘no operation’. For the logical measurement part, we consider transversal Z basis and X basis measurements. In the encoded case, the fault-tolerant measurement involves post-selection and we provide the post-selection rates for each of the demonstrations in Appendix B.

We test these circuits on two quantum computers: `ibm_mumbai`, a 27-qubit device developed by IBM [98], and `ionq-11q`, an 11-qubit device developed by IonQ [35]. The

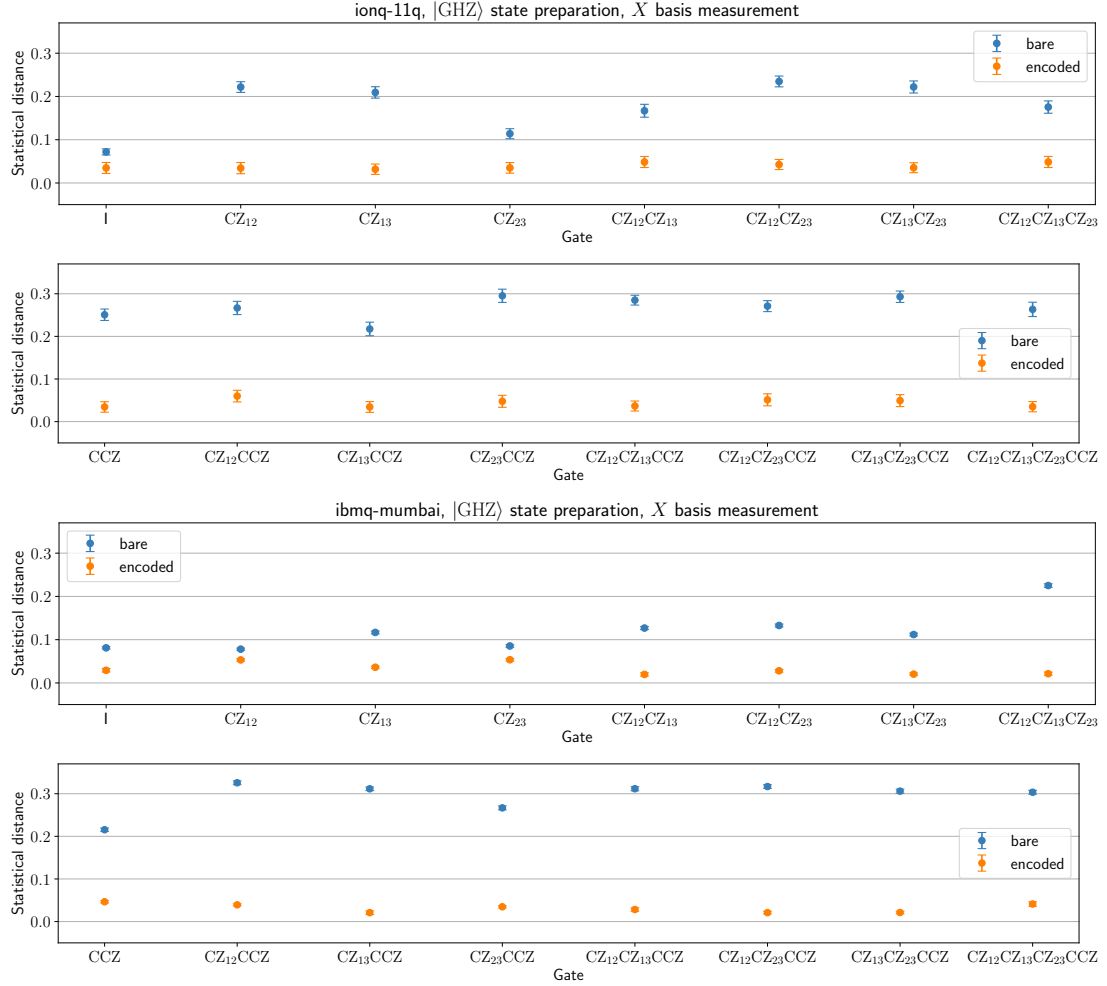


Figure 3.4: Performance of bare (non-encoded) and encoded versions of circuits for preparing states of the form $g|\text{GHZ}\rangle$, where g is a transversal gate of the $[[8,3,2]]$ code. In each case, we measure the qubits in the X basis and we plot the statistical distance of the observed measurement distribution from the ideal distribution. The upper two plots show the data for `ionq-11q`, where we ran 1024 shots for each circuit, and the lower two plots show the data for `ibmq-mumbai` where we ran 10,000 shots for each circuit. In both cases, the error bars are calculated using bootstrap resampling.

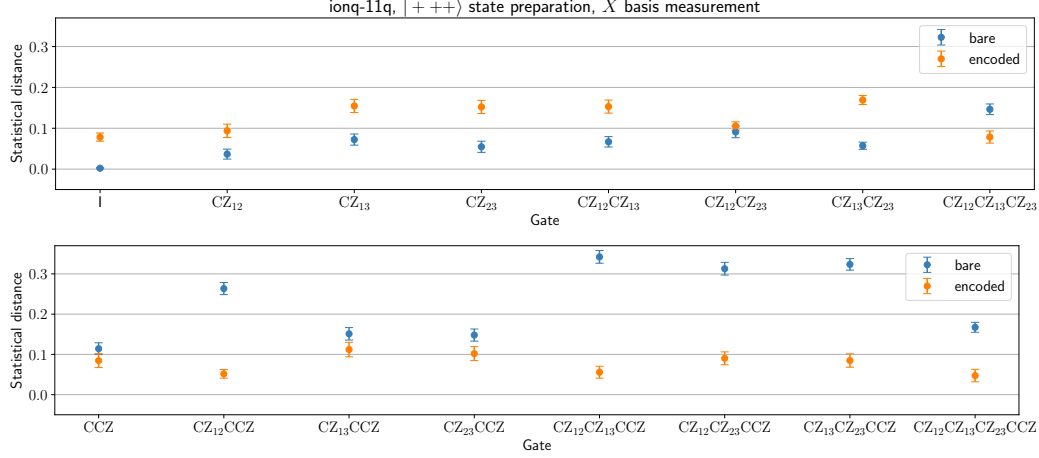


Figure 3.5: Performance of bare (non-encoded) and encoded versions of circuits for preparing states of the form $g|+++ \rangle$, where g is a transversal gate of the $[[8,3,2]]$ code. In each case, we measure the qubits in the X basis and we plot the statistical distance of the observed measurement distribution from the ideal distribution. Each data point represents 1024 shots of the circuit performed on **ionq-11q**, and we use bootstrap resampling to calculate the error bars.

IonQ device has all-to-all qubit connectivity, whereas the IBM device has “heavy-hexagon” qubit connectivity [99]. We only consider $|\text{GHZ}\rangle$ state preparation on the IBM device, as our circuit for preparing logical $|+++ \rangle$ states (Fig. 3.3) does not respect the connectivity constraints of the IBM device and would therefore require SWAP gates to implement, meaning that our error analysis is no longer valid. We leave open the possibility of finding a fault-tolerant circuit for preparing logical $|+++ \rangle$ states on the IBM device. We compare the performance of the encoded circuits against the performance of the bare (no encoding) circuits, using the statistical distance of the output distribution from the ideal output distribution as our metric. The data and analysis code is available at [100].

We show the results for $|\text{GHZ}\rangle$ state preparation and X basis measurement in Fig. 3.4. For both devices and for every transversal gate, we observe improved performance of the encoded version of the circuit.

We show the results for $|+++ \rangle$ state preparation and X basis measurement in Fig. 3.5. The bare version of the circuit performs better for transversal Clifford gates, whereas the encoded version performs better for transversal non-Clifford gates. Notably, we observe lower statistical distances for the preparation of the encoded magic state $\text{CCZ}|+++ \rangle$. We can attribute the difference between the results for Clifford and non-Clifford gates to the

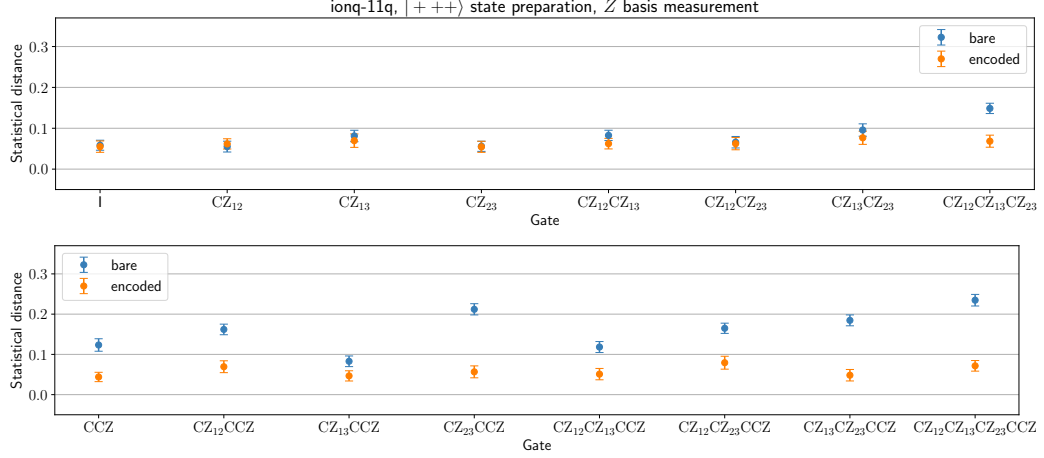


Figure 3.6: Performance of bare (non-encoded) and encoded versions of circuits for preparing states of the form $g|+++ \rangle$, where g is a transversal gate of the $[[8,3,2]]$ code. In each case, we measure the qubits in the Z basis and we plot the statistical distance of the observed measurement distribution from the ideal distribution. Each data point represents 1024 shots of the circuit performed on `ionq-11q`, and we use bootstrap resampling to calculate the error bars.

compilation of the three-qubit CCZ gate into a circuit involving multiple two-qubit gates on the IonQ device [101]. And the discrepancy between the results for $|+++ \rangle$ and $|\text{GHZ}\rangle$ state preparation is expected, given that the bare circuit for preparing the former requires only single-qubit gates and the latter requires two entangling gates.

The results for Z basis measurement (Figs. 3.6, 3.7) are qualitatively similar to that of the X basis measurement. For the demonstrations with $|+++ \rangle$ state preparation and Z basis measurement, we observe improved performance for the encoded circuits containing a transversal non-Clifford gate; see Fig. 3.6. And for the demonstrations with $|\text{GHZ}\rangle$ state preparation and Z basis measurement, we observe improved performance for all of the encoded circuits; see Fig. 3.7. We note that in these cases the gates have no effect on the expected output measurement distribution, as the gates commute with the measurements.

3.4 Discussion

We have shown that using the $[[8,3,2]]$ code allows us to prepare certain (encoded) states more accurately (as measured by the statistical distance) than using the native gates to

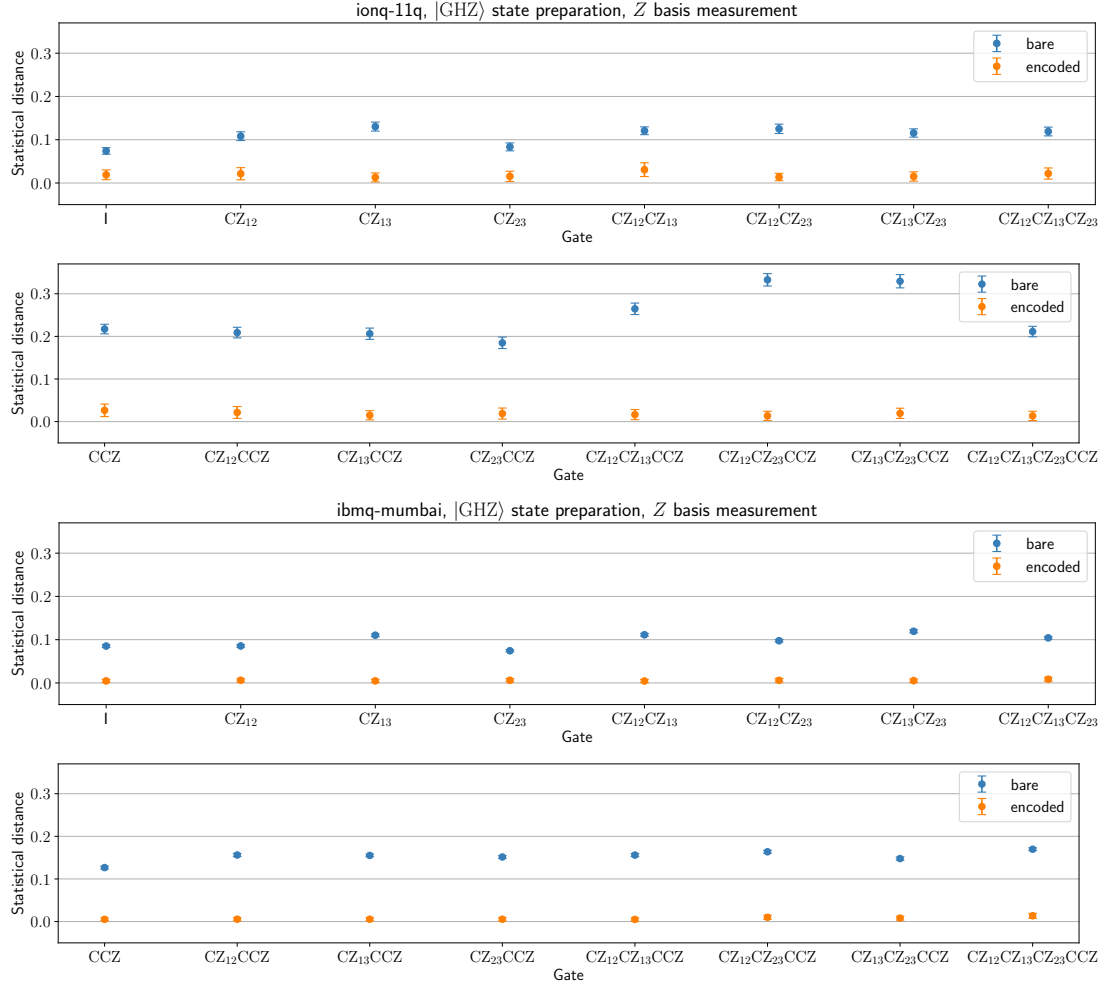


Figure 3.7: Performance of bare (non-encoded) and encoded versions of circuits for preparing states of the form $g|\text{GHZ}\rangle$, where g is a transversal gate of the $[[8,3,2]]$ code. In each case, we measure the qubits in the Z basis and we plot the statistical distance of the observed measurement distribution from the ideal distribution. The upper two plots show the data for `ionq-11q`, where we ran 1024 shots for each circuit, and the lower two plots show the data for `ibmq-mumbai` where we ran 10,000 shots for each circuit. In both cases, the error bars are calculated using bootstrap resampling.

prepare the same (non-encoded) states. We observe this advantage across a range of circuits on two different hardware platforms: IBM’s superconducting qubits and IonQ’s trapped-ion qubits. The all-to-all connectivity of the IonQ device that we used enabled us to run more circuits fault-tolerantly than we could on the IBM device. In particular, we were able to interrogate the performance of the $[[8,3,2]]$ code for preparing magic states of the form $g|+++ \rangle$, where $g \in \text{CCZ} \times \{I, \text{CZ}_{12}, \text{CZ}_{13}, \text{CZ}_{23}\}$. We observe an improved performance for the encoded version of circuits for preparing these states, illustrating the utility of codes like the $[[8,3,2]]$ code, where multi-qubit non-Clifford gates can be applied using single-qubit operations.

The $[[8,3,2]]$ belongs to the family of codes, known as generalized triorthogonal codes [102–104], with transversal multi-qubit Z rotations implemented by single-qubit gates. In future it would be interesting to test the performance of larger codes in this family with higher distance. For example, Ref. [104] gives a $[[64,6,4]]$ code with a transversal $\text{CCZ}^{\otimes 2}$ gate and it is possible that smaller examples could be found using the techniques of [105–107]. In addition, 3D color codes [108–110] are also generalized triorthogonal codes and therefore our approach could be extended to the error-correcting regime by exchanging the $[[8,3,2]]$ code for a color code with larger distance (for concrete examples see [111, 112]).

As with any stabilizer code, the transversal gates of the $[[8,3,2]]$ code do not form a universal set of gates. Therefore, in order to use the $[[8,3,2]]$ code or a similar code to implement an actual quantum algorithm, we would need to supplement the transversal gates with additional fault-tolerant gates in order to obtain a universal gate set. One possibility worth considering would be to explore the implementation of logical gates via permutations of the physical qubits [113, 114], which can be fault-tolerant if implemented by qubit relabelling or physically moving the qubits.

Chapter 4

An algorithm for imaginary time evolution of quantum systems

The primary objective of error correction and mitigation strategies is to minimize noise in quantum systems to enable effective quantum computation. Originally, quantum computers were conceived to efficiently simulate quantum systems [3, 4], a task crucial for advancing fields such as chemistry and materials design. Quantum simulation, for a system described by a Hamiltonian, broadly encompasses two main objectives: understanding static properties such as ground and thermal states of the Hamiltonian, and studying dynamic evolution under the Schrödinger equation (for closed systems) or general master equations (for open systems) over time. Numerous classical and quantum methods have emerged over the years to simulate the static and dynamical properties of quantum systems [115–118]. This chapter explores a technique known as imaginary time evolution for estimating static properties of Hamiltonians.

4.1 Imaginary time evolution (ITE)

For a closed quantum system given by a Hamiltonian H , the state $|\Psi\rangle$ of the system over time is determined by the time-dependent Schrödinger equation,

$$i\hbar \frac{\partial |\Psi\rangle}{\partial t} = H |\Psi\rangle. \quad (4.1)$$

For a time-independent Hamiltonian H with energy eigenstates $\{|e_k\rangle\}$ and corresponding energy values $\{E_k\}$ (enumerated in ascending order), the initial state $|\Psi_0\rangle$ of the system

can be expressed as

$$|\Psi_0\rangle = \sum_k c_k |e_k\rangle. \quad (4.2)$$

For this system, the solution to the Schrödinger equation prescribes the time-dependent state of the system to be

$$|\Psi_t\rangle = \sum_k c_k e^{-i\frac{E_k}{\hbar}t} |e_k\rangle. \quad (4.3)$$

A transformation of the variable $t \rightarrow \tau = it$, (also known as Wick rotation [119]), gives the *imaginary time* Schrödinger equation:

$$-\hbar \frac{\partial |\Psi\rangle}{\partial \tau} = H |\Psi\rangle. \quad (4.4)$$

The state of the system at imaginary time τ is then given by

$$|\Psi_\tau\rangle = \sum_k c_k e^{-\frac{E_k}{\hbar}\tau} |e_k\rangle. \quad (4.5)$$

The state in Eq. 4.5 shows that as the system evolves in imaginary time, the initial coefficients of each eigenstate undergo exponential decay, with the higher energy states decaying faster than the lower energy states. In the limit $\tau \rightarrow \infty$, the state of the system is $|\Psi_\tau\rangle \approx |e_0\rangle$ provided that $c_0 \neq 0$, where $|e_0\rangle$ corresponds to the ground state of the Hamiltonian. This simple transformation from real time to imaginary time evolution (ITE) can thus be used to prepare the ground state of the system.

The ITE operator is also related to the partition function $\mathcal{Z}_\beta$ of the quantum system which is given by

$$\mathcal{Z}_\beta = \text{Tr}(e^{-\beta H}), \quad (4.6)$$

and links imaginary time τ to the inverse temperature β associated with the partition function. This suggests that ITE may also be used to estimate thermal properties of the system [115, 117, 120, 121].

Observe that unlike the real time evolution operator $U = e^{-i\frac{H}{\hbar}t}$ which is unitary, the ITE operator $V = e^{-\frac{H}{\hbar}\tau}$ is non-unitary, making it difficult to directly implement on a quantum computer. Since the ITE operation is *non-trace-preserving* (NTP), applying it to a system requires the resulting state to be normalized. A number of classical and quantum algorithms have been developed for ITE, including recent proposals [115, 122–128] that

are aimed at intermediate to near-term quantum computers. Nevertheless, implementing these algorithms on quantum hardware is likely to require extensive error-mitigation.

In the remainder of this chapter, we propose and analyze a new quantum algorithm to compute imaginary-time evolved expectation values which incorporates ideas from quantum error-mitigation [40]. Specifically, we use a technique inspired by probabilistic error cancellation (PEC) (See discussion in Sec. 1.5) to implement trotterized ITE as a probabilistic linear combination of quantum circuits followed by post-processing. Our algorithm for ITE requires no ancillary qubits, and is also inherently noise-resilient since we can directly implement it as a probabilistic linear combination of native noisy gates. In Sec. 4.2, we first present our algorithm in the context of more general NTP maps and then in Sec. 4.3, we consider its application to ITE, specifically the ITE of the 1D Heisenberg model.

4.2 Quasiprobability decompositions of linear maps

Let $\{\mathcal{B}_i\}$ be a set of ideal basis operations that can be implemented on a quantum device. Note that the elements¹ of $\{\mathcal{B}_i\}$ are *completely positive, and trace preserving (CPTP)*. For an n -qubit device, the set $\{\mathcal{B}_i\}$ is said to be *complete* if it contains 2^{4n} linearly independent elements. Suppose $\mathcal{T}$ is a *hermiticity preserving* linear map from the set of operators that maps a 2^n -dimensional Hilbert space to itself. There exists a decomposition of $\mathcal{T}$ into a linear combination of the elements of a complete basis set $\{\mathcal{B}_i\}$ given by

$$\mathcal{T}(\rho) = \sum_i q_i \mathcal{B}_i(\rho), \quad (4.7)$$

such that the q_i s are real numbers [41] and ρ is the initial state of the system to which the map $\mathcal{T}$ is applied. We can then define

$$\gamma = \sum_i^m |q_i|, \quad p_i = |q_i|/\gamma, \quad (4.8)$$

where $\{p_i\}$ is a probability distribution over the basis elements $\{\mathcal{B}_i\}$. Note that Eq. 4.7 is similar to Eq. 1.58, with the constraint $\sum_i q_i = 1$ being relaxed since we allow $\mathcal{T}$ to be NTP. In literature, a linear combination such as that in Eq. 4.7 is called a quasiprobability distribution (QPD) only when the constraint $\sum_i q_i = 1$ is satisfied. In the remainder of

¹with the exception of some projective measurements which are completely positive (CP), but not necessarily trace preserving (TP).

this chapter, we abuse this terminology and continue to refer to the decomposition as a QPD even though the q_i s do not necessarily sum to 1.

Given a basis $\{\mathcal{B}_i\}$, the QPD of a map $\mathcal{T}$ can be obtained by solving a linear program. The size of this optimization problem scales polynomially with Hilbert-space dimension (2^n) and is generally considered practically feasible only for a few ($\mathcal{O}(1)$) qubits[40, 41]. We also remark that when $\mathcal{T}$ is NTP, the basis set $\{\mathcal{B}_i\}$ must necessarily contain at least one NTP element in order to be able to obtain a QPD. The single-qubit basis [129] shown in Table 4.1 is an example of a complete basis consisting of NTP elements.

4.2.1 An algorithm for estimating expectation values for states evolving under non-trace-preserving maps

For an observable A (having operator norm $\|A\| \leq 1$), we can use Eq. 4.7 to write

$$\text{Tr}[A\mathcal{T}(\rho)] = \sum_i \text{sgn}(q_i) \cdot |q_i| \cdot \text{Tr}[A\mathcal{B}_i(\rho)] \quad (4.9)$$

$$= \gamma \sum_i \text{sgn}(q_i) p_i \cdot \text{Tr}[A\mathcal{B}_i(\rho)]. \quad (4.10)$$

This immediately suggests a method to compute quantities $\text{Tr}[A\mathcal{T}(\rho)]$ given a quantum state ρ , in a manner similar to traditional probabilistic error cancellation (discussed in Sec. 1.5 and Algorithm 1). Note, however, that since $\mathcal{T}$ is not assumed to be unitary or even trace-preserving (TP), $\mathcal{T}(\rho)$ is not necessarily a quantum state. Therefore, to estimate the expectation value of the observable A with respect to the quantum state resulting from applying $\mathcal{T}$, we must rescale Eq. 4.10 by $\text{Tr}[\mathcal{T}(\rho)]$ as follows—

$$\langle A \rangle_{\mathcal{T}} := \frac{\text{Tr}[A\mathcal{T}(\rho)]}{\text{Tr}[\mathcal{T}(\rho)]}. \quad (4.11)$$

The quantity $\text{Tr}[\mathcal{T}(\rho)]$ can itself be estimated from Eq. 4.9 by replacing the observable A with the Identity—

$$\text{Tr}[\mathcal{T}(\rho)] = \gamma \sum_i \text{sgn}(q_i) p_i \cdot \text{Tr}[\mathcal{B}_i(\rho)] \quad (4.12)$$

$$= \gamma \left(\sum_{i, \mathcal{B}_i \in \mathcal{B}_{\text{TP}}} \text{sgn}(q_i) p_i + \sum_{i, \mathcal{B}_i \in \mathcal{B}_{\text{NTP}}} \text{sgn}(q_i) p_i \cdot \text{Tr}[\mathcal{B}_i(\rho)] \right), \quad (4.13)$$

where $\mathcal{B}_{\text{TP}}$ and $\mathcal{B}_{\text{NTP}}$ correspond to the trace-preserving and non-trace-preserving operations in $\{\mathcal{B}_i\}$ respectively. Thus, in order to estimate the rescaling factor $\text{Tr}[\mathcal{T}(\rho)]$, we

need to determine the quantities $\text{Tr}[\mathcal{B}_i(\rho)]$ for $\mathcal{B}_i \in \mathcal{B}_{\text{NTP}}$. In the rest of this work, we consider choices of basis where the NTP operations are projections onto certain states, as is the case for the Endo Basis [129] (refer to the last six elements of Table 4.1). Then, for the NTP basis elements, $\text{Tr}[\mathcal{B}_i(\rho)]$ is simply the rate of successful projections to the state corresponding to the element $\mathcal{B}_i$.

$[1]$		
$[\sigma_X]$		$= [H][S]^2[H]$
$[\sigma_Y]$		$= [H][S]^2[H][S]^2$
$[\sigma_Z]$		$= [S]^2$
$[R_X]$	$= \left[\frac{1+i\sigma^X}{\sqrt{2}}\right]$	$= [H][S]^3[H]$
$[R_Y]$	$= \left[\frac{1+i\sigma^Y}{\sqrt{2}}\right]$	$= [S][H][S]^3[H][S]^3$
$[R_Z]$	$= \left[\frac{1+i\sigma^Z}{\sqrt{2}}\right]$	$= [S]^3$
$[R_{YZ}]$	$= \left[\frac{\sigma^Y + \sigma^Z}{\sqrt{2}}\right]$	$= [H][S]^3[H][S]^2$
$[R_{ZX}]$	$= \left[\frac{\sigma^Z + \sigma^X}{\sqrt{2}}\right]$	$= [S]^3[H][S]^3[H][S]^3$
$[R_{XY}]$	$= \left[\frac{\sigma^X + \sigma^Y}{\sqrt{2}}\right]$	$= [H][S]^2[H][S]^3$
$[\pi_X]$	$= \left[\frac{1+\sigma^X}{2}\right]$	$= [S][H][S][H][P_0][H][S]^3[H][S]^3$
$[\pi_Y]$	$= \left[\frac{1+\sigma^Y}{2}\right]$	$= [H][S]^3[H][P_0][H][S][H]$
$[\pi_Z]$	$= \left[\frac{1+\sigma^Z}{2}\right]$	$= [P_0]$
$[\pi_{YZ}]$	$= \left[\frac{\sigma^Y + i\sigma^Z}{2}\right]$	$= [S][H][S][H][P_0][H][S][H][S]^3$
$[\pi_{ZX}]$	$= \left[\frac{\sigma^Z + i\sigma^X}{2}\right]$	$= [H][S]^3[H][P_0][H][S][H][S]^2$
$[\pi_{XY}]$	$= \left[\frac{\sigma^X + i\sigma^Y}{2}\right]$	$= [P_0][H][S]^2[H]$

Table 4.1: **Endo Basis** : The set $\{\mathcal{B}_i\}$ of 16 single-qubit operations shown in this table is a complete basis for general single-qubit maps. The map $[P_0]$ refers to projection (via measurement) onto state $|0\rangle$ and is a non-trace-preserving, post-selection operation. This set contains 10 trace-preserving elements and 6 non-trace-preserving elements. These operations were introduced in [129] using the notation $[U](\cdot) = U(\cdot)U^\dagger$ to represent the action of maps. All these operations can be realized using Hadamard gates H , phase gates S and postselection operations P_0 .

We provide in Algorithm 2 a protocol to estimate the rescaled expectation values of observable A as shown in Eq. 4.11. The strategy is to use Eqs. 4.9 and 4.12 to separately estimate the numerator and denominator on the RHS of Eq. 4.11. The proof of Lemma 1 in Appendix C.1 shows that the number of circuits N that need to be sampled from the distribution $\{p_i\}$ is proportional to γ^2 . Related results on the use of linear decompositions to implement NTP operations [130] have shown that the optimal value of γ for a general

linear map $\mathcal{T}$ is given by its diamond norm.

We remark that while in Eq. 4.7 we consider the basis operations $\{\mathcal{B}_i\}$ to be ideal (noiseless), in the presence of noise it may also be possible to compose the basis elements with a noise channel $\mathcal{N}$ and use the noisy basis set $\{\mathcal{E}_i = \mathcal{N} \circ \mathcal{B}_i\}$ to obtain a QPD just as in traditional probabilistic error cancellation [129] (PEC) approaches (see discussion on the basis set in Sec. 1.5). Thus, our algorithm can be made inherently noise-resilient (as corroborated by the experimental demonstration in Fig. 4.1). Moreover, in contrast to methods such as block encoding [131, 132] which require ancillae to implement non-unitary operations, our algorithm requires no ancillae to estimate expectation values with respect to states evolving under non-unitary maps.

Algorithm 2 Estimate re-scaled expectations

input: QPD of map $\mathcal{T}$ as $\mathcal{T}(\cdot) = \sum_i q_i \mathcal{B}_i(\cdot)$, initial state ρ
output: estimate of re-scaled expectation value $\langle A \rangle_{\mathcal{T}}$

- 1: **for** $j = 1$ to N **do**
- 2: sample i from distribution $|q_i|/\gamma$
- 3: **if** $\mathcal{B}_i$ is TP **then**
- 4: apply $\mathcal{B}_i$ directly to ρ .
- 5: **else**
- 6: measure in basis corresponding to $\mathcal{B}_i$ and store outcome.
- 7: measure A
- 8: estimate $\text{Tr}[\mathcal{T}(\rho)]$ from the rate of post-selection (of mid-circuit measurement) of each circuit.
- 9: estimate $\text{Tr}[A\mathcal{T}(\rho)]$ from the measurement outcomes of A for the post-selected circuits.
- 10: **return** empirical estimate of $\text{Tr}[A\mathcal{T}(\rho)]/\text{Tr}[\mathcal{T}(\rho)]$

Lemma 1 (Algorithm 2 cost). *Given the QPD of a linear map $\mathcal{T}$ as in Eq. 4.7, with γ given by Eq. 4.8, and an observable A with $\|A\| \leq 1$, Algorithm 2 returns an estimate of the re-scaled expectation value $\langle A \rangle_{\mathcal{T}}$ within error $\epsilon \cdot (\text{Tr}[\mathcal{T}(\rho)])^{-1}$ using $O(\gamma^2 \epsilon^{-2})$ randomly chosen operations sampled from $\{\mathcal{B}_i\}$ according to the probability distribution $\{p_i = |q_i|/\gamma\}$.*

4.3 Estimating expectation values for ITE states via QPD

In this section, we consider the specific case where the map $\mathcal{T}$ in Eq. 4.7 is chosen to be the ITE operator for a Hamiltonian H

$$H = \sum_{l=1}^L H_l \quad (4.14)$$

presented as a sum of $L = O(\text{poly}(n))$ local terms H_l acting non-trivially on at most k qubits. We refer to $T = e^{-\beta H}$ as the ITE operation with β denoting the inverse temperature. Since the ITE operation is HP, we can use Algorithm 2 to estimate expectation values of an observable A with respect to the normalized quantum state $(e^{-\beta H} \rho e^{-\beta H} / \text{Tr}[e^{-\beta H} \rho e^{-\beta H}])$ that results from applying ITE to a system's initial state ρ .

As an example, we consider the ITE operation for the simple case of the 2-qubit 1D Heisenberg Hamiltonian—

$$H = -XX - YY - ZZ \quad (4.15)$$

Since the basis $\{\mathcal{B}_i\}$ in Table 4.1 is known to be complete for all linear maps on single-qubits, we use the 256-element Cartesian product set $\{\mathcal{B}_i\} \times \{\mathcal{B}_j\}$ as the basis for decomposing the 2-qubit ITE operator $e^{-\beta H}$. In Fig. 4.1, we use the QPD obtained and demonstrate the implementation of Algorithm 2 for estimating the energy of the 2-qubit Heisenberg model when it is initialized in the state $|00\rangle$ and undergoes ITE for increasing values of β .

The basis used to obtain the QPD of a map is not unique. In fact, different choices of basis can result in QPDs with different coefficients q_i , and hence different values of γ . In Lemma 1, we show that γ is related to the sampling complexity of Algorithm 2, thus a lower value of γ is preferable. It is known that the optimal value of γ for a map is lower bounded by its diamond norm [130]. In our example, we additionally consider the 2-qubit Takagi basis proposed in Table II. of [133] (also shown in Table 4.2) and find that the ITE operator for the 2-qubit Heisenberg Hamiltonian can be decomposed into it with a lower value of γ , which is closer to the lower bound (see Fig. 4.2). This reduced value of γ can be attributed to the presence of entangling operations in the Takagi basis. Since the Endo basis for 2-qubits is simply the (uncorrelated) Cartesian product of two single-qubit basis sets, it has a high sampling overhead for simulating entangling operations in expectation. This is apparent in the case of the CNOT operation [133, 134], which has $\gamma_{\text{CNOT}} = 9$ for the 2-qubit Endo basis but $\gamma_{\text{CNOT}} = 1$ when decomposed into the Takagi basis.

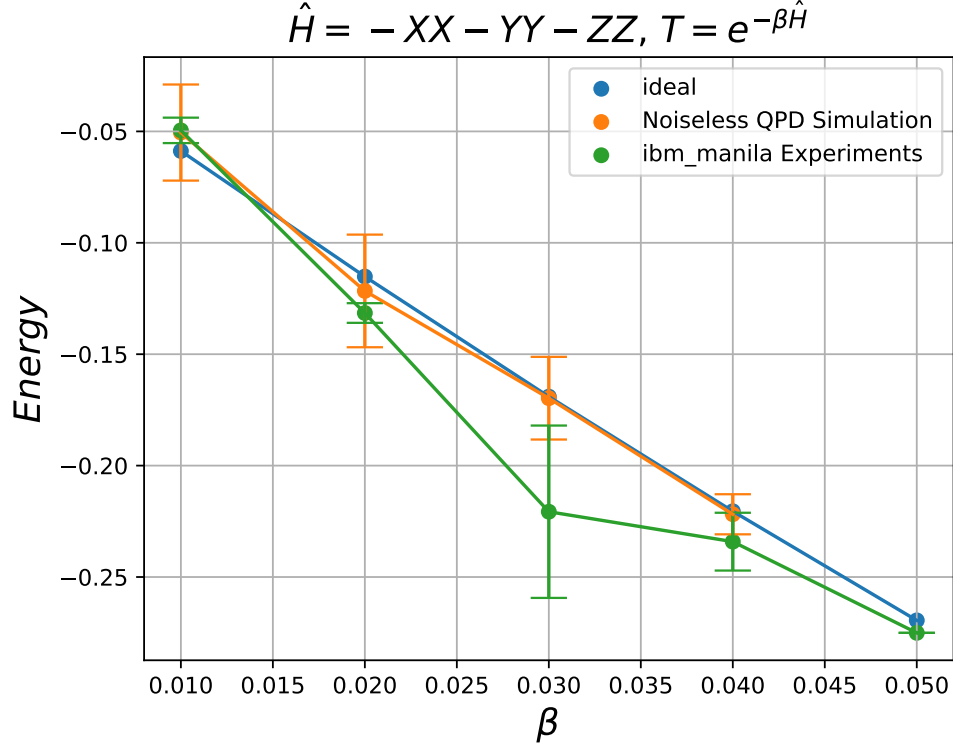


Figure 4.1: Estimation of energy for the 2-qubit Heisenberg model using ITE. For the Hamiltonian $H = -XX - YY - ZZ$, we implement the ITE operation $\mathcal{T}(\cdot) = e^{-\beta H}(\cdot)e^{-\beta H}$ for $\beta = \{0.01, 0.02, 0.03, 0.04, 0.05\}$ on the initial state $|00\rangle$. The blue data points represent the ideal expected energy. The orange points indicate the energies calculated by classically simulating our ITE algorithm. The green points are obtained by running our ITE algorithm with the noisy gates available on the 7-qubit superconducting quantum device **ibm_manila** with no additional error mitigation. The agreement between the ideal values and the quantum experimental data demonstrates the noise-resilience of our ITE algorithm. Note that the data point for the experimental and numerical simulation for $\beta = 0.05$ is shown to have no error bar because it was obtained from a single experiment while those for the other values of β were obtained from the mean of 10 experiments each.

We remark that although the Takagi basis permits a decomposition of the ITE operator in this case, it is not a complete basis for general linear maps on 2-qubits since it has only 241 elements instead of the 256 elements required. The problem of obtaining a QPD for a given map requires careful consideration of two important factors—the choice of basis set used for the decomposition and the value of γ obtained from the QPD. It has been

$\mathcal{B}_1\text{--}\mathcal{B}_{169}$	$\{\mathcal{B}_i\}_{i=1}^{13} \otimes \{\mathcal{B}_i\}_{i=1}^{13}$
$\mathcal{B}_{170}\text{--}\mathcal{B}_{178}$	$\mathcal{CX} + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{179}\text{--}\mathcal{B}_{187}$	$\mathcal{X}_1 \circ \mathcal{CX} \circ \mathcal{X}_1 + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{188}\text{--}\mathcal{B}_{196}$	$\mathcal{CS} + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{197}\text{--}\mathcal{B}_{205}$	$\mathcal{CH} + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{206}\text{--}\mathcal{B}_{214}$	$\mathcal{C}_H\mathcal{X} + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{215}\text{--}\mathcal{B}_{223}$	$\mathcal{CX} \circ \mathcal{H}_1 + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$
$\mathcal{B}_{224}\text{--}\mathcal{B}_{226}$	$\mathcal{S}_W + \text{conjugation with } \mathcal{K}_2, \mathcal{K}_2^\dagger$
$\mathcal{B}_{227}\text{--}\mathcal{B}_{232}$	$i\mathcal{S}_W + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_2^\dagger$
$\mathcal{B}_{233}\text{--}\mathcal{B}_{241}$	$\mathcal{S}_W \circ \mathcal{H}_1 + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$

Table 4.2: **Takagi Basis** [133] : A universal basis for CPTP maps acting on two-qubit systems. $\{\mathcal{B}_i\}_{i=1}^{13}$ refers to the first 13 elements of the Endo Basis. $\mathcal{CX}$, $\mathcal{CS}$, $\mathcal{CH}$, $\mathcal{C}_H\mathcal{X}$ are channel versions of CNOT, controlled-phase, controlled-Hadamard, and NOT controlled with ± 1 eigenstates of the Hadamard gate. $\mathcal{K}$ is the channel version of the gate $K = SH$. $\mathcal{S}_W$ and $i\mathcal{S}_W$ are channel versions of SWAP and iSWAP gates. The subscripts refer to the subsystems that the operations act on. “ $\mathcal{U} + \text{conjugation with } \mathcal{V}$ ” refers to $\mathcal{V}^\dagger \circ \mathcal{U} \circ \mathcal{V}$. Then, “ $\mathcal{U} + \text{conjugation with } \mathcal{K}_{1,2}, \mathcal{K}_{1,2}^\dagger$ ” collects all of the nine possible conjugations with $\mathbb{1}_{12}$, $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_1^\dagger$, $\mathcal{K}_2^\dagger$, $\mathcal{K}_1 \otimes \mathcal{K}_2$, $\mathcal{K}_1 \otimes \mathcal{K}_2^\dagger$, $\mathcal{K}_1^\dagger \otimes \mathcal{K}_2$, $\mathcal{K}_1^\dagger \otimes \mathcal{K}_2^\dagger$ for $\mathcal{U}$.

established that the Endo basis is complete for all linear maps [134]. The Takagi basis is known to be complete [133] for the set of all 2-qubit CPTP maps. However, these bases are not guaranteed to produce the optimal value of γ when decomposing a specific map of interest. In general, the QPD corresponding to the optimal value of γ , i.e., its diamond norm, could be a linear combination of operations that may be expensive (in gate depth) to implement directly on a quantum device. This tradeoff between the generality of the basis and the optimality of the decomposition [133–135] has been well studied for the specific case of QPDs for CPTP maps, with several results on optimizing [41, 136] the choice of basis for different usecases. However, the solution to this problem for general HP maps, including NTP maps such as ITE which commonly occur in physics, is yet (to the best of our knowledge) unknown. Additionally, the problem of finding a basis for decomposing maps acting on systems larger than 2-qubits quickly becomes intractable (since the size of the associated optimization problem grows polynomially with the dimension of the Hilbert space) and remains a challenge even for the case of decomposing CPTP maps.

Thus, when applying our method to the general n -qubit Hamiltonian in Eq. 4.14, we circumvent the problem of obtaining a direct decomposition of the n -qubit operation $e^{-\beta H}$

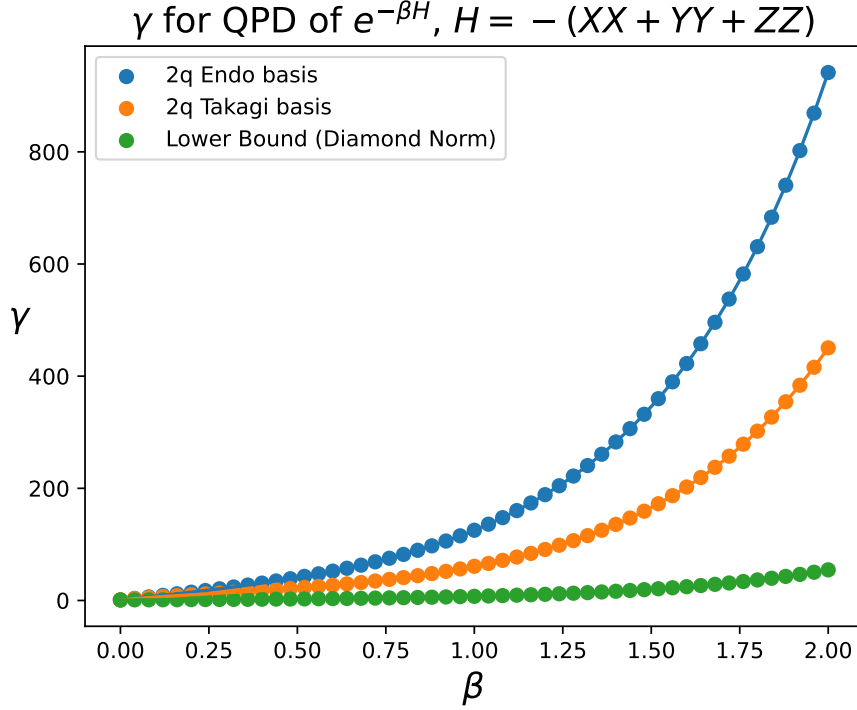


Figure 4.2: We plot the parameter γ , which governs the sampling complexity of Algorithm 2, obtained from the decomposition of the ITE operation $e^{-\beta H}$ for the 2-qubit Heisenberg Hamiltonian for different values of inverse temperature β , and using two different basis sets. The lower bound [130] of γ is obtained from the diamond norm of the map.

by considering its 1st-order Lie-Trotter product formula approximation—

$$e^{-\beta H} \approx e^{-\beta H_1} e^{-\beta H_2} \dots e^{-\beta H_L}. \quad (4.16)$$

Since each H_l is k -local for some constant k , we obtain the QPD of each local ITE operation $e^{-\Delta\tau H_l}$ separately and implement a trotterized ITE (see Fig. 4.3) using Algorithm 3 (See Lemmas 2, 4).

Lemma 2 (First-order Trotter approximation error). *The error in estimating the normalized expectation value $\langle A \rangle_{\mathcal{T}}$ of an observable A for $\mathcal{T}(\rho) = e^{-\beta H} \rho e^{-\beta H}$ using the 1st order Trotter approximation in Eq. 4.16 is $\epsilon_{\text{trot}} \cdot (\text{Tr}[\mathcal{T}(\rho)])^{-1}$ where*

$$\epsilon_{\text{trot}} = \mathcal{O}(c^2 L^2 \beta^2 e^{2cL\beta}) \quad (4.17)$$

and $\|H_l\| \leq c$ for each k -local term H_l in Eq. 4.14.

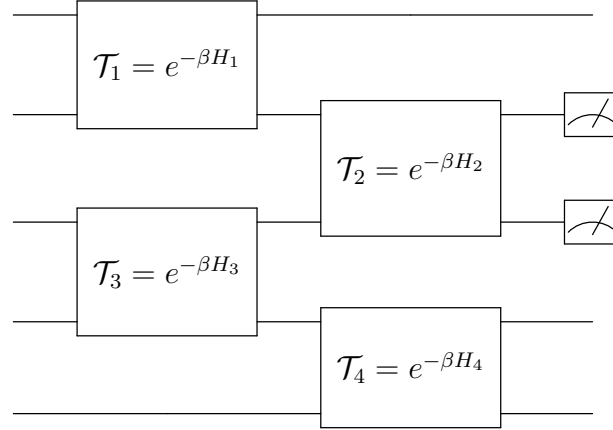


Figure 4.3: A schematic of the trotterized imaginary time evolution of a Hamiltonian $H = H_1 + H_2 + H_3 + H_4$. The QPD of each 2-qubit ITE operation $\mathcal{T}_i = e^{-\beta H_i}$ is used to approximately implement $e^{-\beta H}$ in expectation using Algorithm 3.

In Algorithm 3, we present the protocol for the application of a sequence $(\mathcal{T}_1, \dots, \mathcal{T}_L)$ of maps and in Lemma 3 we analyze its sampling cost (See proof in Appendix C.3). For the n -qubit 1D Heisenberg Hamiltonian, we implement the trotterized ITE operator via $n - 1$ applications of the ITE of the 2-qubit local Hamiltonian in Eq. 4.15.

Lemma 3 (Algorithm 3 cost). *Suppose we are given the QPDs of linear maps $\mathcal{T}_1, \dots, \mathcal{T}_L$, each acting non-trivially on a constant number of qubits, into a set of operations $\{\mathcal{B}_i\}$ as in Eq. 4.7 with parameters $\gamma_1, \dots, \gamma_L$ respectively (Eq. 4.8). Define the n -qubit map ξ as the composition of the operations $(\mathcal{T}_1, \dots, \mathcal{T}_L)$. Then, for a Hermitian observable A with $\|A\| \leq 1$, Algorithm 3 returns an estimate of the re-scaled expectation value $\langle A \rangle_\xi$ within error $\epsilon \cdot (\text{Tr}[\xi(\rho)])^{-1}$ using $\mathcal{O}(\gamma_1^2 \dots \gamma_L^2 \epsilon^{-2})$ random circuits.*

Lemma 4 (Cost of trotterized ITE via QPD). *Suppose we are given the QPDs of linear maps $\mathcal{T}_1, \dots, \mathcal{T}_L$, where each $\mathcal{T}_l(\cdot) = e^{-\beta H_l}(\cdot)e^{-\beta H_l}$ is the ITE operation for the k -local Hamiltonian H_l , into a set of operations $\{\mathcal{B}_i\}$ as in Eq. 4.7 with negativities $\gamma_1, \dots, \gamma_L$ respectively. Define the n -qubit map ξ as the composition of the operations $(\mathcal{T}_1, \dots, \mathcal{T}_L)$ which is the 1st order Trotter approximation of $\mathcal{T}(\cdot) = e^{-\beta H}(\cdot)e^{-\beta H}$ for the Hamiltonian in Eq. 4.16. Then, for a Hermitian observable A with $\|A\| \leq 1$, Algorithm 3 for implementing ξ returns an estimate of the re-scaled expectation value $\langle A \rangle_\mathcal{T}$ within error $\epsilon \cdot (\text{Tr}[\mathcal{T}(\rho)])^{-1}$ using $\mathcal{O}(\gamma_1^2 \dots \gamma_L^2 \epsilon^{-2})$ random circuits. Here $\epsilon = \epsilon_{\text{trot}} + \epsilon_{\text{qpd}}$ is the sum of the corresponding trotterization error (Lemma 2) and the error from Algorithm 3 (see Lemma 3).*

Algorithm 3 Estimate re-scaled expectations for repeated applications of $\mathcal{T}$

input: Operator $\xi = \mathcal{T}_R \circ \dots \circ \mathcal{T}_1$ implemented by the sequence of maps $(\mathcal{T}_1, \dots, \mathcal{T}_R)$, QPDs of the maps $\mathcal{T}_k$ as $\mathcal{T}_k(\cdot) = \sum_i q_{k,i} \mathcal{B}_i(\cdot)$, initial state ρ

output: estimate of re-scaled expectation value $\langle A \rangle_\xi$

- 1: **for** $j = 1$ to N **do**
- 2: sample $(i_1, \dots, i_R)$ from the joint distribution $\{(|q_{1,i_1}|/\gamma, \dots, |q_{R,i_R}|/\gamma)\}$ and implement the circuit consisting of the gate sequence $(\mathcal{B}_{i_1}, \dots, \mathcal{B}_{i_R})$
- 3: **for** $k = 1$ to R **do**
- 4: **if** $\mathcal{B}_{i_k}$ is TP **then**
- 5: apply $\mathcal{B}_{i_k}$ directly to the current state.
- 6: **else**
- 7: measure in basis corresponding to $\mathcal{B}_{i_k}$ and store outcome.
- 8: measure A
- 9: estimate $\text{Tr}[\xi(\rho)]$ from the rate of simultaneous post-selection (of all the mid-circuit measurements) of each of the N circuits.
- 10: estimate $\text{Tr}[A\xi(\rho)]$ from measurement outcomes of A for the post-selected circuits.
- 11: **return** empirical estimate of $\text{Tr}[A\xi(\rho)]/\text{Tr}[\xi(\rho)]$

4.3.1 Estimating thermal state properties via QPD

The thermal state of a system with Hamiltonian H at inverse temperature β is given by the mixed state

$$\rho_\beta = \frac{e^{-\beta H}}{\mathcal{Z}_\beta}, \quad (4.18)$$

where $\mathcal{Z}_\beta = \text{Tr}[e^{-\beta H}]$ is the partition function. The thermal expectation value of an observable O at inverse temperature β is given by

$$\langle O \rangle_\beta = \text{Tr}[O\rho_\beta] = \frac{\text{Tr}[Oe^{-\beta H}]}{\text{Tr}[e^{-\beta H}]} \quad (4.19)$$

It has been shown that we can estimate the thermal expectation values of an observable O by estimating the expectation value of O with respect to a *thermal pure quantum state* [120, 137, 138]. We follow the discussion in [120] where thermal pure quantum states (TPQ) are defined as the (normalized) pure quantum states $|\psi_\beta\rangle$ obtained by applying the ITE operation $e^{-\beta H/2}$ to a random n -qubit Clifford state—

$$|\psi_\beta\rangle = \frac{e^{-\beta H/2} U |0\rangle}{\sqrt{\text{Tr}[\langle 0| U^\dagger e^{-\beta H} U |0\rangle]}}, \quad U \in \mathcal{C}_n \quad (4.20)$$

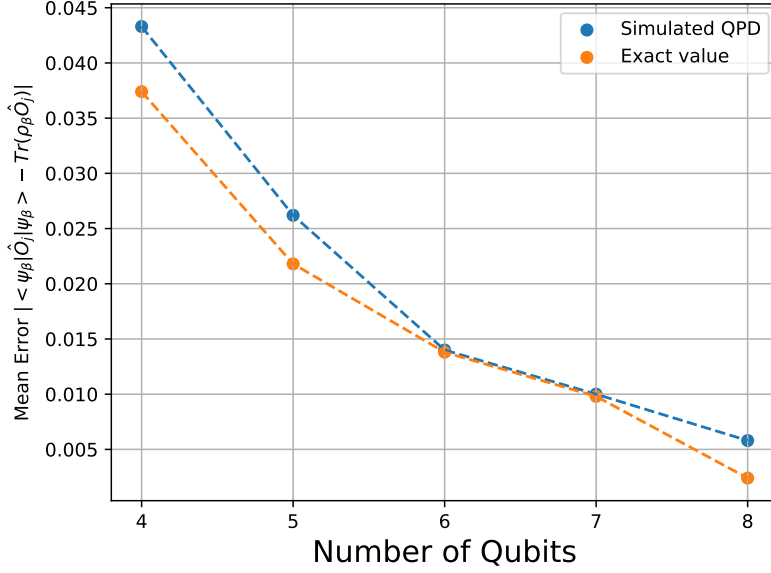


Figure 4.4: We plot the mean error in the estimate of thermal expectation values $\text{Tr}[\rho_\beta O_j]$ of 30 Pauli observables (O_j) by preparing 10 TPQ states $|\psi_\beta\rangle$, (see Eq. 4.20) for 1D Heisenberg models with $n = \{4, 5, 6, 7, 8\}$ qubits respectively. The TPQ states are prepared by applying the ITE operation $e^{-\beta H/2}$ using Algorithm 3 to a random Clifford state for $\beta = 0.04$. The estimation of expectation values of each TPQ state required $N = \{1024, 9400, 51200, 409600, 1638400\}$ for $n = \{4, 5, 6, 7, 8\}$ qubits respectively. The mean error in the estimate of thermal expectation values decays with increasing n , as expected from the inequality in Eq. 4.21. The agreement of the plots of the exact calculated values of TPQ expectations and those obtained via a QPD implementation for the mean error in the estimate of the thermal expectation values via TPQ states suggests the validity of our method.

Since sampling from the n -qubit Clifford group $\mathcal{C}_n$ yields a unitary 3-design [139], a TPQ state $|\psi_\beta\rangle$ satisfies

$$\Pr[|\langle\psi_\beta|O|\psi_\beta\rangle - \text{Tr}[\rho_\beta O]| \geq \epsilon] \leq C_\epsilon e^{-\alpha n}, \quad (4.21)$$

for all Hermitian observables O with at most polynomially large spectral norm $\|O\|$.

As an example, we estimate thermal state properties of the n -qubit 1D Heisenberg Hamiltonian (Eq. 4.22) by implementing the ITE operation $e^{-\beta H}$.

$$H = \sum_{i=1}^{n-1} -(X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}) \quad (4.22)$$

The Heisenberg model has $n - 1$ local terms acting on pairs of nearest neighbors. Each local term is given by the 2-qubit Hamiltonian $H_{\text{local}} = -(XX + YY + ZZ)$. Thus, we implement a trotterized ITE of the n -qubit Heisenberg model via $n - 1$ applications of the ITE operator $e^{-\beta H_{\text{local}}}$ using Algorithm 3.

For each system size n , we prepare 10 TPQ states as per Eq. 4.20 by implementing trotterized ITE as described. For each of the 10 TPQ states, we estimate the expectation values of a set of 30 randomly chosen Pauli operators $\{O_j\}$. In Fig. 4.4, we plot the mean (over 10 TPQ states and the 30 observables $\{O_j\}$) error in the estimate of the thermal expectation values $\langle O_j \rangle_\beta$. The plot reflects the 3-design property of the Clifford group and indicates that the mean error in estimates of thermal expectation values decays with increasing system size. Fig. 4.4 demonstrates the use of TPQ states for estimating thermal properties of the Heisenberg model. Our demonstration is similar to that in Fig. A2. of Ref. [120] where ITE is implemented using a block encoding of the Hamiltonian requiring at least $\log(n)$ ancillae, along with controlled operations between the ancillae and the system qubits.

Our method of simulating ITE in expectation via QPDs has two merits that make it amenable for implementation on NISQ devices—where qubits are few, and multi-qubit entangling gates are at least 10x noisier than single qubit gates. Firstly, our methods require no ancillae. Secondly, we avoid the problem of implementing multi-qubit-controlled operations between the ancillae and the data qubits which can require large SWAP networks for implementation on devices that only permit entangling operations between nearest neighbors (such as IBM’s superconducting devices). The cost of our technique lies predominantly in the number of random circuits $N = \mathcal{O}(\gamma^{\text{poly}(n)})$ required to estimate expectation values for an n -qubit system. We also remark that despite this exponential sampling cost, our method provides an explicit strategy for implementing ITE in terms of local operations implementable on a quantum device. Finding these local operations requires solving a classical optimization problem, but over a space of dimension $\mathcal{O}(2^k)$ for k -local Hamiltonians. We may contrast these with variational methods [128, 140] for implementing ITE which require solving optimization problems over a heuristic parameter space that generally scales with the system size n .

4.4 Discussion

In this chapter, we discussed imaginary time evolution (ITE) and presented a new quasiprobabilistic approach to estimating expectation values of imaginary time evolved quantum states. We showed how a modification of the traditional probabilistic error cancellation

(PEC) technique can allow us to estimate rescaled expectation values for systems evolving under non-unitary operations, while requiring no ancillary qubits. Moreover, we demonstrated that our quasiprobabilistic method is inherently noise-resilient, allowing us to simulate general linear maps (in expectation) as a probabilistic linear combination of native noisy gates. The cost of our method lies in the exponential sampling overhead, which we argue is on par with other methods of implementing non-unitary operations such as block encoding. Another key component of our protocol is the choice of basis for decomposing a given map. Identifying a basis that leads to a low value of the sampling parameter γ while still being easy to implement on a device (low native-gate depth) requires solving a linear program over the space of CP operations. However, in the case of ITE of local Hamiltonians, we can fix the search space of the basis by trotterizing the ITE operation. Thus, our method suggests a clear strategy for scaling up the size of the system being simulated, without relying on heuristic ansatze. As a proof of concept of our method, we also simulated our quasiprobabilistic approach to successfully estimate thermal properties of the 1D Heisenberg model. We remark that in its current form, the implementation of ITE via QPD (to generate thermal pure quantum states) on quantum hardware is challenging because of the n -qubit random Clifford states that need to be prepared. In future work, we could consider replacing the random Clifford unitaries by other t -designs sampled from simpler gates, for example matchgates, to obtain similar bounds on the error in estimating thermal properties, while also making the preparation of thermal states more amenable to implementation on NISQ hardware.

Chapter 5

Conclusion

Modern day quantum devices boast a large number of qubits ($\sim 10 - 100$) compared to the previous generations, but the noise in these devices continues to present a significant challenge to quantum computation. Nevertheless, the era of noisy-intermediate scale quantum devices has provided an exciting testbed for the development of techniques that will be useful for building large-scale fault tolerant quantum computers. They have also led to a transition from a purely theoretical, device-agnostic approach to quantum algorithms design to a more tailored approach that takes into account constraints such as the ratio of 2-qubit to 1-qubit gate errors, and qubit connectivity in the device architecture, in order to learn new ways to make quantum computers scalable for performing useful computation. In this thesis, I have demonstrated methods to reduce the effects of noise on NISQ devices along with new ways to improve standard quantum algorithms to make them more amenable to implementation on quantum devices. In chapter 2, we showed that the sweep rule is robust to quantum (syndrome) measurement errors when used as a decoder for topological quantum codes, making it a prime candidate for fast, local decoders for QECC. In chapter 3, we implemented logical operations in the $[[8, 3, 2]]$ quantum code on superconducting and trapped ion devices. We demonstrated a fault-tolerant implementation of the $[[8, 3, 2]]$ code's non-Clifford logical gates on quantum hardware and showed that the encoded gates have an improved performance over the unencoded version. The $[[8, 3, 2]]$ code permits a fault-tolerant CCZ gate which is crucial for the creation of magic states, which are resource states for many quantum algorithms such as Shor's algorithm. The demonstrated improvement of these encoded non-Clifford gates on NISQ devices serves as a simple, yet powerful proof-of-concept of the advantages of fault-tolerant quantum computation via quantum error correcting codes, while also underlining the utility of NISQ devices. In chapter 4, I presented an improved algorithm for quantum simulation, par-

ticularly the implementation of imaginary time evolution of quantum systems to estimate static properties of quantum systems. Our quasiprobabilistic approach for estimating properties of imaginary time-evolved states is based on a modification of similar QPD-based approaches to error mitigation and quantum simulation, and requires no ancillae or controlled operations between ancillae and the system of interest. We also showed that our protocol is inherently noise-resilient and thus applicable to NISQ devices.

In addition to presenting specific findings in each chapter, this thesis aims to underscore the diverse range of paradigms employed to address the challenges faced when using quantum devices. These include cellular automata, the use of flag qubits in QEC, quasiprobabilistic implementations of linear maps, and twirling. As we enter the era of pre-fault-tolerant quantum computation, we envision these paradigms, borrowed from maths, physics, and computer science, forming a valuable toolkit for overcoming the unique challenges on the path to constructing large-scale fault tolerant quantum computers.

References

- [1] Annie Ray, Raymond Laflamme, and Aleksander Kubica. “Protecting information via probabilistic cellular automata”. In: *Physical Review E* 109.4 (Apr. 2024), p. 044141.
- [2] Daniel Honciuc Menendez, Annie Ray, and Michael Vasmer. “Implementing fault-tolerant non-Clifford gates using the $[[8,3,2]]$ color code”. In: *Physical Review A* 109.6 (June 2024), p. 062438.
- [3] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488.
- [4] Seth Lloyd. “Universal Quantum Simulators”. In: *Science* 273.5278 (Aug. 1996), pp. 1073–1078.
- [5] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Nov. 1994, pp. 124–134.
- [6] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Physical Review A* 52.4 (Oct. 1995). Publisher: American Physical Society, R2493–R2496.
- [7] Andrew Steane. “Simple Quantum Error Correcting Codes”. In: *Physical Review A* 54.6 (Dec. 1996), pp. 4741–4751.
- [8] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (Aug. 1996). Publisher: American Physical Society, pp. 1098–1105.
- [9] Dorit Aharonov and Michael Ben-Or. “Fault-Tolerant Quantum Computation with Constant Error Rate”. In: *SIAM Journal on Computing* 38.4 (Jan. 2008). Publisher: Society for Industrial and Applied Mathematics, pp. 1207–1282.

- [10] P.W. Shor. “[Fault-tolerant quantum computation](#)”. In: *Proceedings of 37th Conference on Foundations of Computer Science*. ISSN: 0272-5428. Oct. 1996, pp. 56–65.
- [11] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. “[Resilient Quantum Computation](#)”. In: *Science* 279.5349 (Jan. 1998). Publisher: American Association for the Advancement of Science, pp. 342–345.
- [12] Daniel Greenbaum. *[Introduction to Quantum Gate Set Tomography](#)*. arXiv:1509.02921 [quant-ph]. Sept. 2015.
- [13] Michael A. Nielsen and Isaac L. Chuang. *[Quantum Computation and Quantum Information: 10th Anniversary Edition](#)*. ISBN: 9780511976667 Publisher: Cambridge University Press. Dec. 2010.
- [14] Mark M. Wilde. *[From Classical to Quantum Shannon Theory](#)*. arXiv:1106.1445. Nov. 2016.
- [15] Yongshan Ding et al. “[Systematic Crosstalk Mitigation for Superconducting Qubits via Frequency-Aware Compilation](#)”. In: *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. Oct. 2020, pp. 201–214.
- [16] Arnaud Carignan-Dugas, Joel J. Wallman, and Joseph Emerson. “[Bounding the average gate fidelity of composite channels using the unitarity](#)”. In: *New Journal of Physics* 21.5 (May 2019). Publisher: IOP Publishing, p. 053016.
- [17] Joseph Emerson et al. “[Symmetrized characterization of noisy quantum processes](#)”. eng. In: *Science (New York, N.Y.)* 317.5846 (Sept. 2007), pp. 1893–1896.
- [18] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. “[Characterizing quantum gates via randomized benchmarking](#)”. In: *Physical Review A* 85.4 (Apr. 2012), p. 042311.
- [19] Steven T. Flammia and Joel J. Wallman. “[Efficient Estimation of Pauli Channels](#)”. In: *ACM Transactions on Quantum Computing* 1.1 (Dec. 2020), 3:1–3:32.
- [20] Joseph Emerson, Robert Alicki, and Karol Życzkowski. “[Scalable noise estimation with random unitary operators](#)”. In: *Journal of Optics B: Quantum and Semiclassical Optics* 7.10 (Sept. 2005), S347.
- [21] Robin Harper et al. “[Statistical analysis of randomized benchmarking](#)”. In: *Physical Review A* 99.5 (May 2019), p. 052350.
- [22] Alexander Erhard et al. “[Characterizing large-scale quantum computers via cycle benchmarking](#)”. In: *Nature Communications* 10.1 (Nov. 2019), p. 5347.

- [23] Andreas Elben et al. “The randomized measurement toolbox”. In: *Nature Reviews Physics* 5.1 (Jan. 2023), pp. 9–24.
- [24] Andrew Steane. “Multiple Particle Interference and Quantum Error Correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (Nov. 1996), pp. 2551–2577.
- [25] Daniel Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation*. arXiv:0904.2557 [quant-ph]. Apr. 2009.
- [26] A. M. Steane. “Error Correcting Codes in Quantum Theory”. In: *Physical Review Letters* 77.5 (July 1996), pp. 793–797.
- [27] Michael Herold et al. “Cellular-automaton decoders for topological quantum memories”. In: *npj Quantum Information* 1.1 (Oct. 2015). Publisher: Nature Publishing Group, pp. 1–8.
- [28] Aleksander Kubica and John Preskill. “Cellular-Automaton Decoders with Provable Thresholds for Topological Codes”. In: *Physical Review Letters* 123.2 (July 2019). Publisher: American Physical Society, p. 020501.
- [29] Joschka Roffe et al. “Decoding across the quantum low-density parity-check code landscape”. In: *Physical Review Research* 2.4 (Dec. 2020), p. 043423.
- [30] Pavel Panteleev and Gleb Kalachev. “Degenerate Quantum LDPC Codes With Good Finite Length Performance”. In: *Quantum* 5 (Nov. 2021), p. 585.
- [31] Christopher T. Chubb and Steven T. Flammia. “Statistical Mechanical Models for Quantum Codes with Correlated Noise”. In: *Annales de l’Institut Henri Poincaré D* 8 (2021), pp. 269–321.
- [32] A.Yu Kitaev. “Fault-Tolerant Quantum Computation by Anyons”. In: *Annals of Physics* 303 (2003), pp. 2–30.
- [33] Eric Dennis et al. “Topological Quantum Memory”. In: *Journal of Mathematical Physics* 43 (2002), pp. 4452–4505.
- [34] Austin G. Fowler. *Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time*. arXiv:1307.1740 [quant-ph]. Oct. 2014.
- [35] K. Wright et al. “Benchmarking an 11-Qubit Quantum Computer”. In: *Nat. Commun.* 10.1 (Nov. 2019), p. 5464.
- [36] Petar Jurcevic et al. “Demonstration of Quantum Volume 64 on a Superconducting Quantum Computing System”. In: *Quantum Sci. Technol.* 6.2 (Apr. 2021), p. 025020.

- [37] Bryan Eastin and Emanuel Knill. “Restrictions on Transversal Encoded Quantum Gate Sets”. In: *Phys. Rev. Lett.* 102.11 (Mar. 2009), p. 110502.
- [38] Isaac H. Kim et al. “Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules”. In: *Physical Review Research* 4.2 (Apr. 2022), p. 023019.
- [39] Yihui Quek et al. *Exponentially tighter bounds on limitations of quantum error mitigation*. arXiv:2210.11505. Mar. 2024.
- [40] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error Mitigation for Short-Depth Quantum Circuits”. In: *Physical Review Letters* 119.18 (Nov. 2017). Publisher: American Physical Society, p. 180509.
- [41] Christophe Piveteau. *Advanced Methods for Quasiprobabilistic Quantum Error Mitigation*. Master Thesis, ETH Zurich. 2020.
- [42] Christophe Piveteau, David Sutter, and Stefan Woerner. “Quasiprobability decompositions with reduced sampling overhead”. In: *npj Quantum Information* 8.1 (Feb. 2022). Publisher: Nature Publishing Group, pp. 1–9.
- [43] Hakop Pashayan, Joel J. Wallman, and Stephen D. Bartlett. “Estimating Outcome Probabilities of Quantum Circuits Using Quasiprobabilities”. In: *Physical Review Letters* 115.7 (Aug. 2015), p. 070501.
- [44] Tomas Jochym-O’Connor and Theodore J. Yoder. “Four-dimensional toric code with non-Clifford transversal gates”. In: *Physical Review Research* 3.1 (Feb. 2021), p. 013118.
- [45] Daniel Gottesman. *The Heisenberg Representation of Quantum Computers*. arXiv: 9807006.
- [46] Michael Vasmer, Dan E. Browne, and Aleksander Kubica. “Cellular Automaton Decoders for Topological Quantum Codes with Noisy Measurements and Beyond”. In: *Scientific Reports* 11 (2021), p. 2027.
- [47] H. Bombin and M. A. Martin-Delgado. “Topological Quantum Distillation”. In: *Physical Review Letters* 97 (2006), p. 180501.
- [48] H. Bombin and M. A. Martin-Delgado. “Optimal Resources for Topological Two-Dimensional Stabilizer Codes: Comparative Study”. In: *Physical Review A* 76 (2007), p. 012305.
- [49] Aleksander Marek Kubica. *The ABCs of the Color Code: A Study of Topological Quantum Codes as Toy Models for Fault-Tolerant Quantum Computation and Quantum Phases Of Matter*. PhD Thesis, California Institute of Technology. 2018.

- [50] Danuta Makowiec and Piotr Gnaciński. “Universality Class of Probabilistic Cellular Automata”. In: *Cellular Automata*. Vol. 2493. Springer Berlin Heidelberg, 2002, pp. 104–113.
- [51] Stephen Wolfram. “Statistical Mechanics of Cellular Automata”. In: *Reviews of Modern Physics* 55 (1983), pp. 601–644.
- [52] Charles H. Bennett and G. Grinstein. “Role of Irreversibility in Stabilizing Complex and Nonergodic Behavior in Locally Interacting Discrete Systems”. In: *Physical Review Letters* 55 (1985), pp. 657–660.
- [53] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford : New York: Clarendon Press ; Oxford University Press, 1999.
- [54] M. Suzuki. “New Universality of Critical Exponents”. In: *Progress of Theoretical Physics* 51 (1974), pp. 1992–1993.
- [55] Andrei L Toom. “Stable and attractive trajectories in multicomponent systems”. In: *Advances in Probability* 6.1 (1980), pp. 549–575.
- [56] G. Grinstein. “Can Complex Structures Be Generically Stable in a Noisy World?”. In: *IBM Journal of Research and Development* 48 (2004), pp. 5–12.
- [57] Francesca Valery Day and Sean Barrett. *The Ising Ferromagnet as a Self-Correcting Physical Memory: A Monte-Carlo Study*. arXiv:1201.0390. Jan. 2012.
- [58] R. Peierls. “On Ising’s Model of Ferromagnetism”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 32 (1936), pp. 477–481.
- [59] Lars Onsager. “Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition”. In: *Physical Review* 65 (1944), pp. 117–149.
- [60] W Lenz. “Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern”. In: *Z. Phys.* 21 (1920), pp. 613–615.
- [61] Barry M. McCoy and Tai Tsun Wu. *The Two-Dimensional Ising Model*. Harvard University Press, 1973.
- [62] Nikolas P. Breuckmann et al. *Local Decoders for the 2D and 4D Toric Code*. arXiv:1609.00510. Sept. 2016.
- [63] Aleksander Kubica and Nicolas Delfosse. “Efficient Color Code Decoders in $d \geq 2$ Dimensions from Toric Code Decoders”. In: *Quantum* 7 (2023), p. 929.
- [64] Nikolas P. Breuckmann. *PhD Thesis: Homological Quantum Codes Beyond the Toric Code*. arXiv:1802.01520. Feb. 2018.
- [65] Fernando Pastawski, Lucas Clemente, and Juan Ignacio Cirac. “Quantum Memories Based on Engineered Dissipation”. In: *Physical Review A* 83 (2011), p. 012304.

- [66] H. Bombin. “Topological Subsystem Codes”. In: *Physical Review A* 81 (2010), p. 032301.
- [67] H Bombin, Guillaume Duclos-Cianci, and David Poulin. “Universal Topological Phase of Two-Dimensional Stabilizer Codes”. In: *New Journal of Physics* 14 (2012), p. 073048.
- [68] Aleksander Kubica et al. “Three-Dimensional Color Code Thresholds via Statistical-Mechanical Mapping”. In: *Physical Review Letters* 120 (2018), p. 180501.
- [69] Arpit Dua et al. “Clifford-Deformed Surface Codes”. In: *PRX Quantum* 5 (1 2024), p. 010347.
- [70] Benjamin J. Brown et al. “Quantum Memories at Finite Temperature”. In: *Reviews of Modern Physics* 88 (2016), p. 045005.
- [71] Barbara M. Terhal. “Quantum Error Correction for Quantum Memories”. In: *Reviews of Modern Physics* 87 (2015), pp. 307–346.
- [72] Sergey Bravyi and Alexei Kitaev. “Universal Quantum Computation with Ideal Clifford Gates and Noisy Ancillas”. In: *Phys. Rev. A* 71.2 (Feb. 2005), p. 022316.
- [73] E. Knill. *Fault-Tolerant Postselected Quantum Computation: Schemes*. arXiv:quant-ph/0402171. Feb. 2004.
- [74] Sergey Bravyi and Jeongwan Haah. “Magic-State Distillation with Low Overhead”. In: *Phys. Rev. A* 86.5 (Nov. 2012), p. 052329.
- [75] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. “Unfolding the Color Code”. In: *New J. Phys.* 17.8 (Aug. 2015), p. 083026.
- [76] Earl Campbell. *The smallest interesting colour code*. <https://earlrcampbell.com/2016/09/26/the-smallest-interesting-colour-code/>. 2016.
- [77] C. J. Ballance et al. “High-Fidelity Quantum Logic Gates Using Trapped-Ion Hyperfine Qubits”. In: *Phys. Rev. Lett.* 117 (6 Aug. 2016), p. 060504.
- [78] Yulin Wu et al. “Strong Quantum Computational Advantage Using a Superconducting Quantum Processor”. In: *Phys. Rev. Lett.* 127 (18 Oct. 2021), p. 180501.
- [79] W. Huang et al. “Fidelity Benchmarks for Two-Qubit Gates in Silicon”. In: *Nature* 569.7757 (May 2019), pp. 532–536.
- [80] S. A. Moses et al. “A Race-Track Trapped-Ion Quantum Processor”. In: *Physical Review X* 13.4 (Dec. 2023), p. 041052.
- [81] Daniel Gottesman. *Quantum fault tolerance in small experiments*. arXiv:1610.03507. Oct. 2022.

- [82] Alexandre M. Souza et al. “Experimental Magic State Distillation for Fault-Tolerant Quantum Computing”. In: *Nat. Commun.* 2.1 (Jan. 2011), p. 169.
- [83] Jingfu Zhang, Raymond Laflamme, and Dieter Suter. “Experimental Implementation of Encoded Logical Qubit Operations in a Perfect Quantum Error Correcting Code”. In: *Phys. Rev. Lett.* 109 (10 Sept. 2012), p. 100503.
- [84] Daniel Nigg et al. “Experimental Quantum Computations on a Topologically Encoded Qubit”. In: *Science* 345.6194 (July 2014), pp. 302–305.
- [85] Laird Egan et al. *Fault-Tolerant Operation of a Quantum Error-Correction Code*. arXiv:2009.11482. Jan. 2021.
- [86] Lukas Postler et al. “Demonstration of Fault-Tolerant Universal Quantum Gate Operations”. In: *Nature* 605.7911 (May 2022), pp. 675–680.
- [87] C. Ryan-Anderson et al. *Implementing Fault-tolerant Entangling Gates on the Five-qubit Code and the Color Code*. 2022.
- [88] Christophe Vuillot. “Is Error Detection Helpful on IBM 5Q Chips ?” In: *Quantum Inf. Comput.* 18.11&12 (2018), pp. 0949–0964.
- [89] Robin Harper and Steven T. Flammia. “Fault-Tolerant Logical Gates in the IBM Quantum Experience”. In: *Phys. Rev. Lett.* 122.8 (Feb. 2019), p. 080504.
- [90] Riddhi S. Gupta et al. *Encoding a Magic State with beyond Break-Even Fidelity*. arXiv:2305.13581. May 2023.
- [91] Vlatko Vedral, Adriano Barenco, and Artur Ekert. “Quantum networks for elementary arithmetic operations”. In: *Phys. Rev. A* 54 (1 July 1996), pp. 147–153.
- [92] Craig Gidney. “Halving the cost of quantum addition”. In: *Quantum* 2 (June 2018), p. 74.
- [93] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1484–1509.
- [94] Craig Gidney and Martin Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. In: *Quantum* 5 (Apr. 2021), p. 433.
- [95] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD Thesis, Caltech, arXiv:quant-ph/9705052. 1997.
- [96] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [97] Rui Chao and Ben W. Reichardt. “Quantum error correction with only two extra qubits”. In: *Phys. Rev. Lett.* 121.5 (Aug. 2018), p. 050502.

- [98] *IBM Quantum*. <https://quantum-computing.ibm.com/>.
- [99] Jared B. Hertzberg et al. “Laser-Annealing Josephson Junctions for Yielding Scaled-up Superconducting Quantum Processors”. In: *npj Quantum Inf.* 7.1 (Aug. 2021), p. 129.
- [100] Michael Vasmer. *Data analysis for "Implementing fault-tolerant non-Clifford gates using the $[[8,3,2]]$ color code"*. <https://github.com/MikeVasmer/832-code>.
- [101] *Getting started with Native Gates*. <https://ionq.com/docs/getting-started-with-native-gates>.
- [102] Earl T. Campbell and Mark Howard. “Unifying Gate Synthesis and Magic State Distillation”. In: *Phys. Rev. Lett.* 118 (6 Feb. 2017), p. 060501.
- [103] Earl T. Campbell and Mark Howard. “Unified Framework for Magic State Distillation and Multiqubit Gate Synthesis with Reduced Resource Cost”. In: *Phys. Rev. A* 95.2 (Feb. 2017), p. 022316.
- [104] Jeongwan Haah and Matthew B. Hastings. “Codes and Protocols for Distilling T , Controlled- S , and Toffoli Gates”. In: *Quantum* 2 (June 2018), p. 71.
- [105] Sepehr Nezami and Jeongwan Haah. “Classification of Small Triorthogonal Codes”. In: *Phys. Rev. A* 106.1 (July 2022), p. 012437.
- [106] Jingzhen Hu, Qingzhong Liang, and Robert Calderbank. “Designing the Quantum Channels Induced by Diagonal Gates”. In: *Quantum* 6 (Sept. 2022), p. 802.
- [107] Mark A. Webster, Armanda O. Quintavalle, and Stephen D. Bartlett. *Transversal Diagonal Logical Operators for Stabiliser Codes*. 2023.
- [108] H. Bombin and M. A. Martin-Delgado. “Topological Computation without Braiding”. In: *Phys. Rev. Lett.* 98 (16 Apr. 2007), p. 160502.
- [109] Héctor Bombín. “Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes”. In: *New Journal of Physics* 17.8 (Aug. 2015), p. 083002.
- [110] Aleksander Kubica and Michael E. Beverland. “Universal transversal gates with color codes: A simplified approach”. In: *Phys. Rev. A* 91 (3 Mar. 2015), p. 032330.
- [111] Michael Vasmer and Dan E. Browne. “Three-dimensional surface codes: Transversal gates and fault-tolerant architectures”. In: *Phys. Rev. A* 100 (1 July 2019), p. 012312.
- [112] Michael Vasmer and Aleksander Kubica. “Morphing Quantum Codes”. In: *PRX Quantum* 3 (3 Aug. 2022), p. 030319.

- [113] Markus Grassl and Martin Roetteler. “[Leveraging automorphisms of quantum codes for fault-tolerant quantum computation](#)”. In: *2013 IEEE International Symposium on Information Theory*. 2013, pp. 534–538.
- [114] Rui Chao and Ben W. Reichardt. “[Fault-Tolerant Quantum Computation with Few Qubits](#)”. In: *npj Quantum Inf.* 4.1 (Dec. 2018), p. 42.
- [115] Mario Motta et al. “[Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution](#)”. In: *Nature Physics* 16.2 (Feb. 2020). Publisher: Nature Publishing Group, pp. 205–210.
- [116] Jules Tilly et al. “[The Variational Quantum Eigensolver: A review of methods and best practices](#)”. In: *Physics Reports*. The Variational Quantum Eigensolver: a review of methods and best practices 986 (Nov. 2022), pp. 1–128.
- [117] E. M. Stoudenmire and Steven R. White. “[Minimally entangled typical thermal state algorithms](#)”. In: *New Journal of Physics* 12.5 (May 2010), p. 055026.
- [118] Ryan Babbush et al. “[Quantum simulation of exact electron dynamics can be more efficient than classical mean-field methods](#)”. In: *Nature Communications* 14.1 (July 2023), p. 4058.
- [119] Alexander Miessen et al. “[Quantum algorithms for quantum dynamics](#)”. In: *Nature Computational Science* 3.1 (Jan. 2023), pp. 25–37.
- [120] Luuk Coopmans, Yuta Kikuchi, and Marcello Benedetti. “[Predicting Gibbs-State Expectation Values with Pure Thermal Shadows](#)”. In: *PRX Quantum* 4.1 (Jan. 2023), p. 010305.
- [121] Steven R. White. “[Minimally Entangled Typical Quantum States at Finite Temperature](#)”. In: *Physical Review Letters* 102.19 (May 2009), p. 190601.
- [122] David Poulin and Pawel Wocjan. “[Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer](#)”. In: *Phys. Rev. Lett.* 103 (22 Nov. 2009), p. 220502.
- [123] Kristan Temme et al. “[Quantum metropolis sampling](#)”. In: *Nature* 471.7336 (2011), pp. 87–90.
- [124] Anirban Narayan Chowdhury and Rolando D. Somma. “[Quantum Algorithms for Gibbs Sampling and Hitting-Time Estimation](#)”. In: *Quant. Inf. Comp.* 17.1–2 (2017), pp. 41–64.
- [125] Joran van Apeldoorn et al. “[Quantum SDP-Solvers: Better upper and lower bounds](#)”. In: *Quantum* 4 (Feb. 2020), p. 230.

- [126] Zoe Holmes et al. “Quantum algorithms from fluctuation theorems: Thermal-state preparation”. In: *Quantum* 6 (2022), p. 825.
- [127] Chi-Fang Chen et al. “Quantum Thermal State Preparation”. In: (). arXiv:2303.18224.
- [128] Shi-Ning Sun et al. “Quantum Computation of Finite-Temperature Static and Dynamical Properties of Spin Systems Using Quantum Imaginary Time Evolution”. In: *PRX Quantum* 2.1 (Feb. 2021), p. 010317.
- [129] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical Quantum Error Mitigation for Near-Future Applications”. In: *Phys. Rev. X* 8.3 (July 2018), p. 031027.
- [130] Xuanqiang Zhao et al. *Power of quantum measurement in simulating unphysical operations*. arXiv:2309.09963.
- [131] Daan Camps et al. “Explicit Quantum Circuits for Block Encodings of Certain Sparse Matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 45.1 (Mar. 2024), pp. 801–827.
- [132] Daan Camps and Roel Van Beeumen. “FABLE: Fast Approximate Quantum Circuits for Block-Encodings”. In: *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Sept. 2022, pp. 104–113.
- [133] Ryuji Takagi. “Optimal resource cost for error mitigation”. In: *Physical Review Research* 3.3 (Aug. 2021). Publisher: American Physical Society, p. 033178.
- [134] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical Quantum Error Mitigation for Near-Future Applications”. In: *Physical Review X* 8.3 (July 2018), p. 031027.
- [135] Suguru Endo et al. “Mitigating algorithmic errors in a Hamiltonian simulation”. In: *Physical Review A* 99.1 (Jan. 2019), p. 012334.
- [136] Christophe Piveteau, David Sutter, and Stefan Woerner. “Quasiprobability Decompositions with Reduced Sampling Overhead”. In: *npj Quantum Inf* 8.1 (Feb. 2022), p. 12.
- [137] Sho Sugiura and Akira Shimizu. “Canonical Thermal Pure Quantum State”. In: *Physical Review Letters* 111.1 (July 2013), p. 010401.
- [138] Sho Sugiura and Akira Shimizu. “Thermal Pure Quantum States at Finite Temperature”. In: *Physical Review Letters* 108.24 (June 2012), p. 240401.
- [139] Zak Webb. *The Clifford group forms a unitary 3-design*. arXiv:1510.02769 [quant-ph]. Nov. 2016.
- [140] Sam McArdle et al. “Variational ansatz-based quantum simulation of imaginary time evolution”. In: *npj Quantum Information* 5.1 (Sept. 2019), pp. 1–6.

- [141] Ruben S. Andrist. *Understanding Topological Quantum Error-Correction Codes Using Classical Spin Models*. PhD Thesis, ETH Zurich. 2012.
- [142] Andrew M. Childs. *Lecture notes on quantum algorithms*. <https://www.cs.umd.edu/~amchilds/qa/qa.pdf>. 2022.

APPENDICES

Appendix A

Simulation of sweep rule

A.1 Details of numerical analysis of memory lifetime

A.1.1 Obtaining data

For parameters T, h and linear lattice size L , we simulate the sweep rule $n = 1000$ times, each simulation starting with all spins aligned. For each simulation, we smoothen the magnetization $M(t)$ as follows.

- Coarse graining the magnetization $M(t)$ by a factor of $L/2$ to obtain the coarse grained magnetization

$$M_{\text{CG}}(t') = \frac{2}{L} \sum_{t=t'+1}^{t'+L/2} M(t), \quad (\text{A.1})$$

where t' is a multiplicity of $L/2$.

- Convoluting $M_{\text{CG}}(t')$ by a rectangular filter (moving average filter) of width $w = L$ (at low temperatures) or $w = L/2$ (at high temperatures) to obtain the smoothened magnetization

$$M_{\text{smooth}}(t') = \frac{1}{w} \sum_{i=0}^{w-1} M_{\text{CG}}\left(t' + \frac{1}{2}iL\right). \quad (\text{A.2})$$

For the i^{th} simulation, where $i \in \{1, 2, \dots, n\}$, we record the smallest value of t_i for which $M_{\text{smooth}}(t_i) = 0$. This is noted as the time of the first logical bit-flip in the i^{th} simulation. The memory lifetime $\tau_{L,T}$ of the system of lattice size L at temperature T is then estimated by the mean of the recorded values t_i .

We remark that in our numerical analysis we also compared the rectangular filter in Eq. (A.2) with Gaussian filters of the same width. We observed that the smoothened magnetization M_{smooth} obtained with the rectangular filter enabled a clearer identification of logical bit-flips than that obtained with the Gaussian filter.

A.1.2 Identifying phase transitions

For a given value of h , we fit the $\tau_{L,T}$ data obtained for various L and T into the fitting ansatz in Eq. (2.6). We use the bootstrapping method to estimate the variance of the fitting parameters (a, b, c) , i.e., for each L, T , we resample (with replacement) from the data set $t_i, i \in \{1, 2, \dots, n\}$ to obtain data sets $t_i^1, t_i^2, \dots, t_i^B$, where $B = 50$. For each of the B data sets we calculate $\tau_{L,T}$ and fit into the ansatz to obtain the value of the fit parameters a, b, c . Each parameter is then estimated by its mean and variance over the B samples (see Fig. A.1).

We identify the phase transition as the temperature where the parameter b becomes zero and exponential scaling of memory lifetime with L is lost. In our analysis, we observed that in the neighbourhood of the phase transition, the parameters a, c do not strongly depend on temperature T . So, in this neighbourhood, we fit the resampled memory lifetime data into the ansatz while constraining parameters a, c to be constant over the range of temperatures. Constraining parameters a, c in this manner (see Fig. A.1(b)) gives a smooth plot of b as a function of T allowing us to clearly identify the point where $b = 0$.

A.1.3 Analysis of the sweep rule on triangular lattices

We use the aforementioned techniques and identify the critical temperatures T_c^h of the sweep rule at different values of bias $h = \{0, 0.125, 0.25, 0.5, 0.75, 1\}$. In Fig. A.2 we show the fit obtained at each value of h for the memory lifetime τ_L estimated using the original data set (no resampling). Fig. A.3 shows the plots of b as a function of T for the corresponding values of h obtained after resampling from the original data set. The points (T_c^h, h) are used to construct the phase diagram of the sweep rule shown in Fig. 2.10.

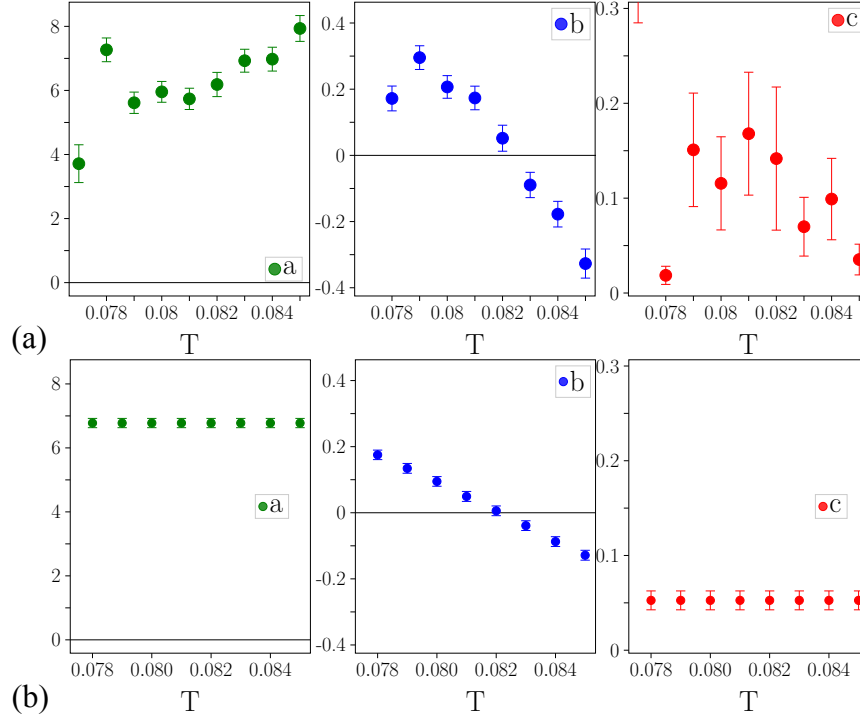


Figure A.1: (a) Estimating parameters a, b, c from the values obtained by fitting resampled data sets at $h = 0.5$ into the ansatz Eq. (2.6). (b) By constraining a, c to be constant over the range of temperatures we obtain a smoother plot of b as a function of T and, subsequently, can identify the phase transition as the point where $b = 0$, i.e., $T_c^{0.50} = 0.082 \pm 0.001$.

A.1.4 Analysis of Toom's rule on square lattices

We also report the same numerical analysis (see Fig. A.5) of memory lifetime for Toom's rule on square lattices and obtain the corresponding phase diagram (see Fig. A.4). This phase diagram corroborates the prediction in Ref. [52] of the existence of a region of non-zero measure in the (T, h) plane where two stable phases coexist.

A.2 Details of statistical-mechanical analysis

We simulate the sweep rule at $h = 0$ for $2^{26} - 1$ time steps, divided into consecutive windows of size $W \in \{2^0, 2^1, \dots, W_{\max} = 2^{25}\}$ for linear lattice sizes $L = \{12, 16, 24, 32, 48, 64, 96\}$.

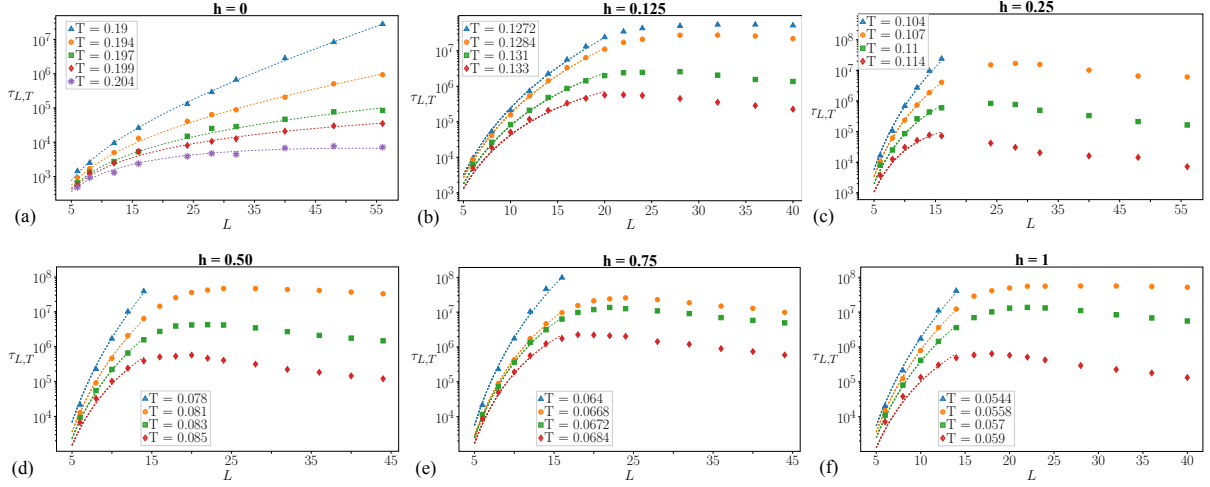


Figure A.2: Plots of memory lifetime $\tau_{L,T}$ of the sweep rule as a function of L . The dashed lines indicate the best fit into the ansatz in Eq. 2.6. The data in these plots have been fit by taking into account the statistical errors in the numerical estimation of memory lifetime. To avoid clutter, we do not display the corresponding asymmetric error bars.

The states of the system at each time step in the last window of the simulation (from $t = 2^{25}$ to $t = 2^{26} - 1$) together constitute an ensemble of states sampled from the stationary distribution of the PCA. We estimate ensemble averages of a property A of the PCA as

$$\langle A \rangle = \frac{1}{W_{\max}} \sum_{t=W_{\max}}^{2W_{\max}-1} A(t), \quad (\text{A.3})$$

where $A(t)$ is the value of the property A of the system at time step t . In Sec. 2.4, we study the ensemble averages of the magnetization M , magnetic susceptibility χ and the two-point correlation function ξ (see Fig. 2.11). These ensemble averages are estimated as follows

$$\langle M \rangle = \frac{1}{W_{\max}} \sum_{t=W_{\max}}^{2W_{\max}-1} |M(t)|, \quad (\text{A.4})$$

$$\langle \chi \rangle = \frac{1}{2L^2} \frac{\langle M^2 \rangle - \langle M \rangle^2}{T}, \quad (\text{A.5})$$

$$\langle \xi \rangle = \frac{1}{2 \sin(k_{\min}/2)} \sqrt{\frac{\langle \chi_m(\vec{0}) \rangle}{\langle \chi_m(\vec{k}_{\min}) \rangle} - 1}, \quad (\text{A.6})$$

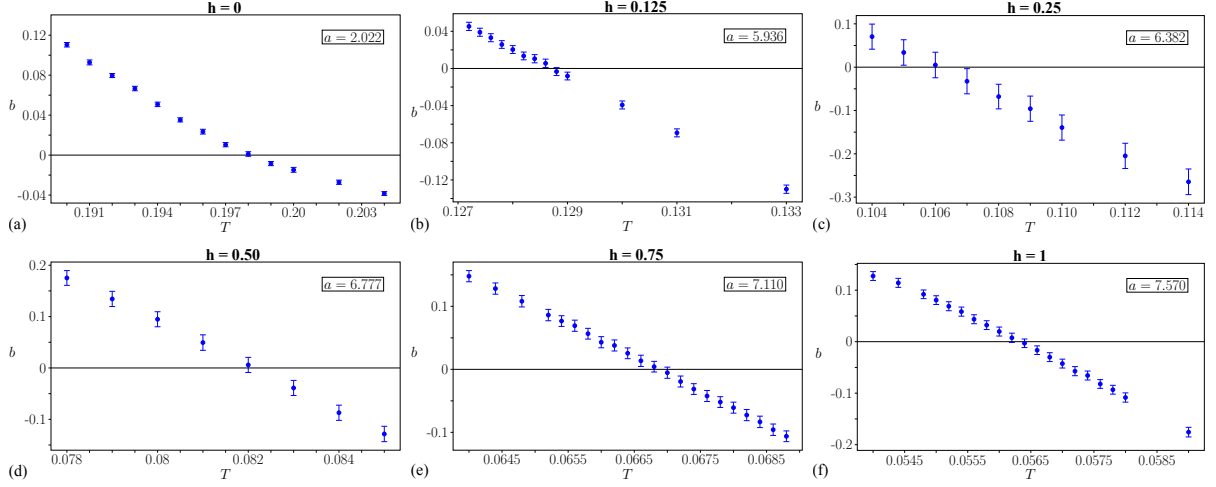


Figure A.3: Plot of parameter b as a function of temperature T for respective values of h corresponding to the fit obtained in Fig. A.2. We identify the critical temperature T_c^h at each value of h as the point where $b = 0$. In each case, we constrain the parameters a, c to be fixed in the neighborhood of T_c^h and report the values of a (polynomial scaling). The critical temperatures thus obtained are $T_c^0 = 0.198 \pm 0.001$, $T_c^{0.125} = 0.129 \pm 0.001$, $T_c^{0.25} = 0.106 \pm 0.002$, $T_c^{0.50} = 0.082 \pm 0.001$, $T_c^{0.75} = 0.0670 \pm 0.0004$ and $T_c^{1.00} = 0.0564 \pm 0.0004$.

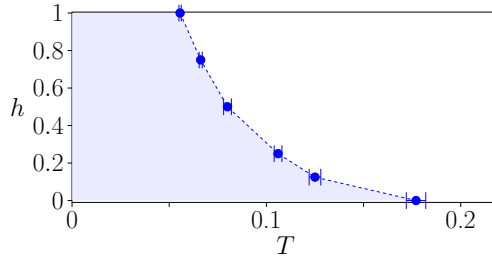


Figure A.4: Phase diagram of the Toom's rule in the (T, h) plane. The shaded region corresponds to a region where two stable phases coexist and the Toom's rule can work as memory. The diagram is symmetric with respect to the $h = 0$ axis.

where $\vec{k}_{\min}$ is the shortest wave vector of the lattice,

$$\chi_m(\vec{k}) = \frac{1}{N} \left\langle \sum_{i,j} s_i s_j e^{i\vec{k} \cdot \vec{R}_{i,j}} \right\rangle, \quad (\text{A.7})$$

and $\vec{R}_{ij}$ is the displacement vector from the spin at the i^{th} cell to the spin at the j^{th} cell of the lattice. Note that although we represent the triangular lattice (see Fig. 2.3) as a

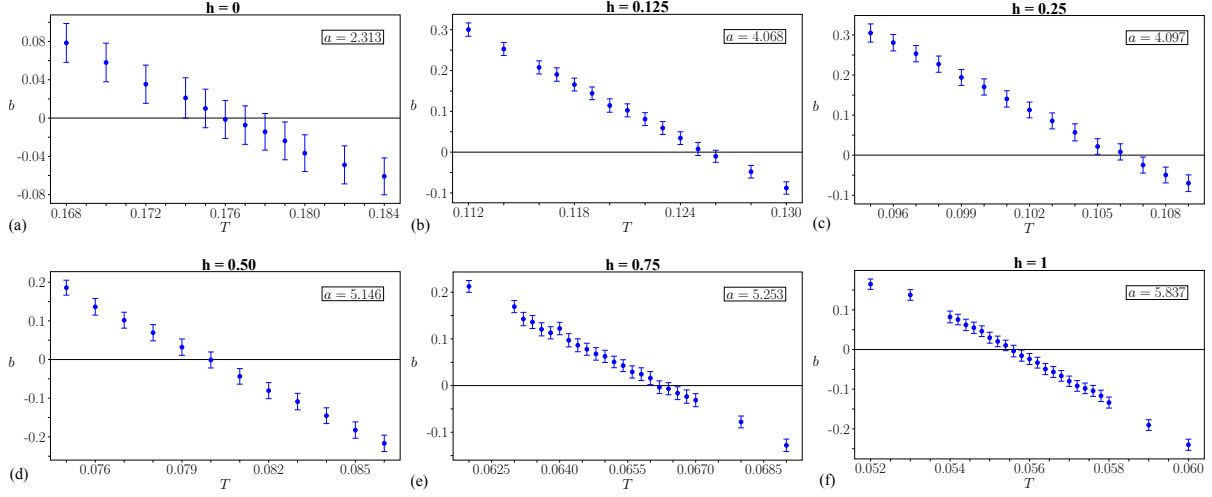


Figure A.5: Plots of parameter b as a function of temperature T obtained from fitting memory lifetime of Toom's rule into the ansatz in Eq. (2.6). The critical temperatures obtained are $T_c^0 = 0.177 \pm 0.005$, $T_c^{0.125} = 0.125 \pm 0.003$, $T_c^{0.25} = 0.106 \pm 0.002$, $T_c^{0.50} = 0.080 \pm 0.002$, $T_c^{0.75} = 0.0662 \pm 0.0008$ and $T_c^{1.00} = 0.0556 \pm 0.0006$.

square lattice with double the number of faces for convenience, we are actually studying the triangular lattice of equilateral triangles (see Fig. A.6) with periodic boundaries. This is reflected in the way the displacement vector $\vec{R}_{i,j}$ is calculated to estimate the ensemble average of the two-point correlation function ξ [141].

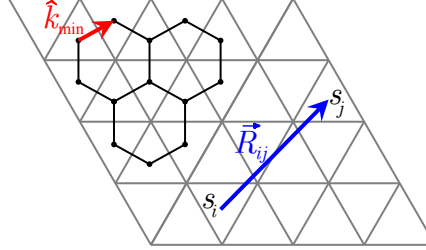


Figure A.6: The sweep rule is implemented on a triangular lattice with periodic boundary conditions. A lattice of linear lattice size $L = 4$ is shown here. The $N = 32$ spins situated at the face centres of the triangles form a honeycomb lattice (black edges) whose wave vector lies along the unit vector $\hat{k}_{\min}$ (red). The displacement vector $\vec{R}_{i,j}$ from s_i to s_j is depicted in blue.

A.3 Critical exponents of variants of the sweep rule

As discussed in Sec. 2.2, sweep rule can be implemented synchronously or asynchronously. Similarly, Toom's rule (which is equivalent to the sweep rule acting on square lattices) can also be implemented both synchronously and asynchronously. In Table A.1, we report the critical temperature T_c^0 and critical exponents ν, β, γ obtained from the statistical-mechanical analysis of each of these variants at $h = 0$ (see Figs. A.7, A.8 and A.9). The critical exponents we report for Toom's rule are in agreement with those reported previously in Ref. [50]. In case of the synchronous variant of both sweep rule and Toom's rule, the ratios β/ν and γ/ν are equal to those of the 2D Ising model. Therefore, they belong to the weak universality class of the Ising model [50]. The asynchronous variants of both rules have the same critical exponents as the 2D Ising model (with some discrepancies in our reported values of γ).

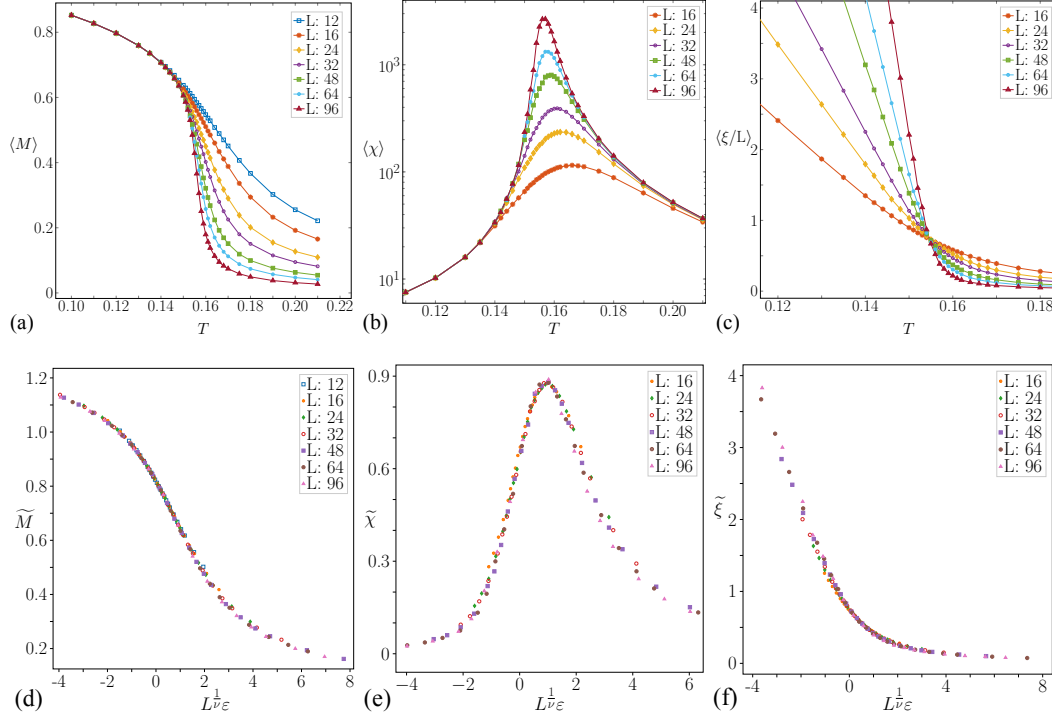


Figure A.7: Critical behavior of the asynchronous sweep rule at $h = 0$. In (a)-(c), we plot ensemble averages of magnetization M , magnetic susceptibility χ and two-point correlation function ξ/L , whereas in (d)-(f), we plot their respective data collapses.

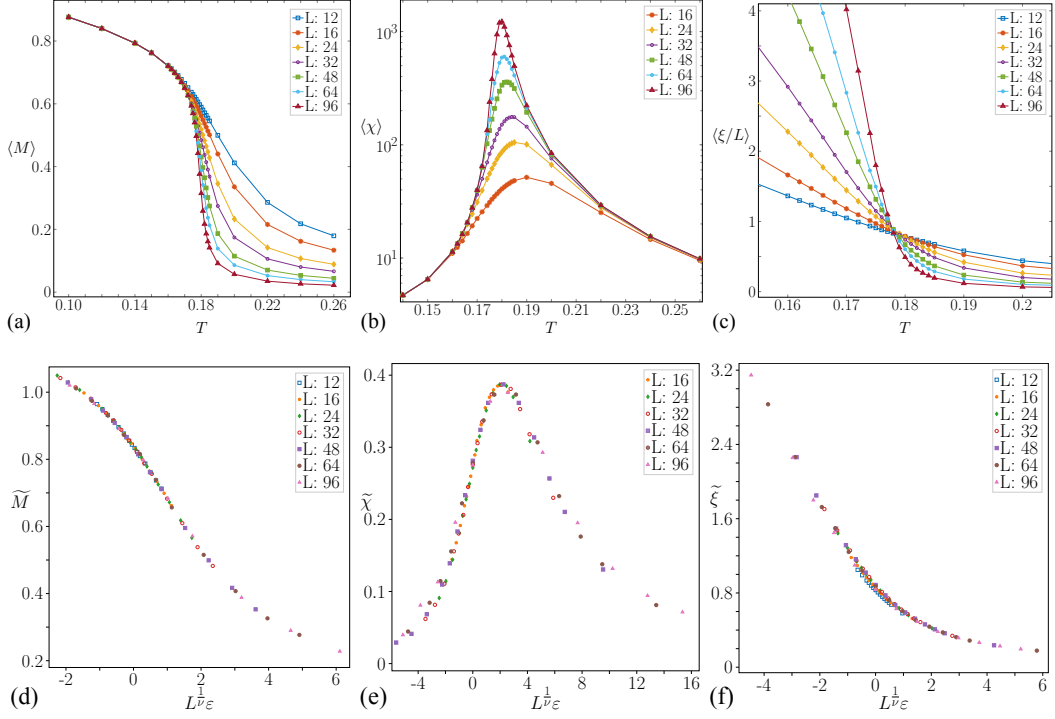


Figure A.8: Critical behavior of synchronous Toom's rule at $h = 0$. In (a)-(c), we plot ensemble averages of magnetization M , magnetic susceptibility χ and two-point correlation function ξ/L , whereas in (d)-(f), we plot their respective data collapses.

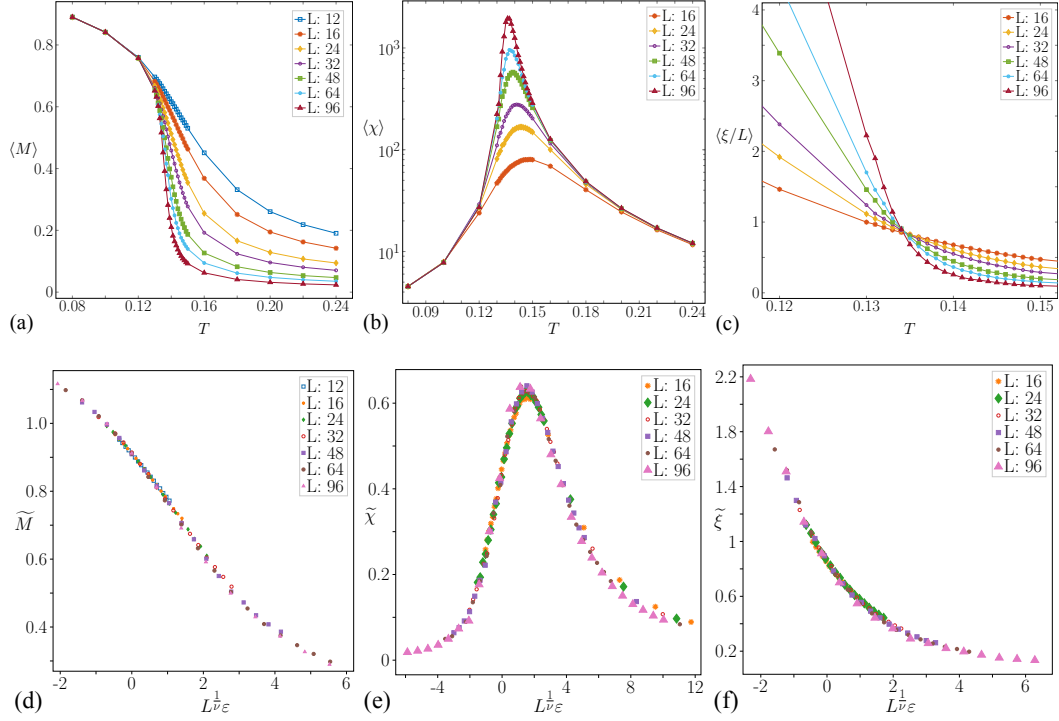


Figure A.9: Critical behavior of asynchronous Toom's rule at $h = 0$. In (a)-(c), we plot ensemble averages of magnetization M , magnetic susceptibility χ and two-point correlation function ξ/L , whereas in (d)-(f), we plot their respective data collapses.

PCA	ν	β	γ	T_c^0
synchronous sweep	0.924(8)	0.120(2)	1.62(1)	0.1973(1)
asynchronous sweep	1.06(3)	0.132(6)	1.88(3)	0.1548(1)
synchronous Toom's	0.89(3)	0.112(6)	1.54(6)	0.1779(1)
asynchronous Toom's	1.08(2)	0.130(6)	1.85(6)	0.1342(1)

Table A.1: Critical exponents of the synchronous and asynchronous variants of the sweep rule (on triangular lattices) and Toom's rule (on square lattices).

Appendix B

Additional details of the demonstration of $[[8,3,2]]$ code

B.1 Post-selection rates

Here, we provide the post-selection rates (proportion of accepted shots) for our demonstrations. Two-qubit gates and measurements are the noisiest operations in superconducting and trapped-ion devices [35, 36], hence we can approximate the error model in these devices with ideal state preparation and one-qubit gates, but noisy two-qubit gates and measurements (1% error rate). Suppose that any two-qubit gate error or measurement error causes us to discard the run, then (to first order) we would expect a post-selection rate of $1 - 0.01(n_m + n_g)$, where n_m is the number of measurements and n_g is the number of two-qubit gates. By counting the relevant locations in Figs. 3.2, 3.3, we therefore estimate a post-selection rate of 69% and 86% for the circuits with $|+++ \rangle$ and $|\text{GHZ}\rangle$ state preparation, respectively. These estimates are reasonably close to the values we observe in our demonstrations; see Tables B.1, B.2, B.3.

B.2 Additional hardware details

State	Gate	Measurement	Device	Post-selection rate
$ +++ \rangle$	I	X basis	ionq-11q	78%
$ +++ \rangle$	CZ_{12}	X basis	ionq-11q	78%
$ +++ \rangle$	CZ_{13}	X basis	ionq-11q	69%
$ +++ \rangle$	$CZ_{12}CZ_{13}$	X basis	ionq-11q	76%
$ +++ \rangle$	CZ_{23}	X basis	ionq-11q	80%
$ +++ \rangle$	$CZ_{12}CZ_{23}$	X basis	ionq-11q	77%
$ +++ \rangle$	$CZ_{13}CZ_{23}$	X basis	ionq-11q	79%
$ +++ \rangle$	$CZ_{12}CZ_{13}CZ_{23}$	X basis	ionq-11q	80%
$ +++ \rangle$	CCZ	X basis	ionq-11q	75%
$ +++ \rangle$	$CZ_{12}CCZ$	X basis	ionq-11q	81%
$ +++ \rangle$	$CZ_{13}CCZ$	X basis	ionq-11q	74%
$ +++ \rangle$	$CZ_{12}CZ_{13}CCZ$	X basis	ionq-11q	84%
$ +++ \rangle$	$CZ_{23}CCZ$	X basis	ionq-11q	75%
$ +++ \rangle$	$CZ_{12}CZ_{23}CCZ$	X basis	ionq-11q	83%
$ +++ \rangle$	$CZ_{13}CZ_{23}CCZ$	X basis	ionq-11q	77%
$ +++ \rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	X basis	ionq-11q	77%
$ +++ \rangle$	I	Z basis	ionq-11q	73%
$ +++ \rangle$	CZ_{12}	Z basis	ionq-11q	78%
$ +++ \rangle$	CZ_{13}	Z basis	ionq-11q	71%
$ +++ \rangle$	$CZ_{12}CZ_{13}$	Z basis	ionq-11q	74%
$ +++ \rangle$	CZ_{23}	Z basis	ionq-11q	72%
$ +++ \rangle$	$CZ_{12}CZ_{23}$	Z basis	ionq-11q	72%
$ +++ \rangle$	$CZ_{13}CZ_{23}$	Z basis	ionq-11q	73%
$ +++ \rangle$	$CZ_{12}CZ_{13}CZ_{23}$	Z basis	ionq-11q	71%
$ +++ \rangle$	CCZ	Z basis	ionq-11q	74%
$ +++ \rangle$	$CZ_{12}CCZ$	Z basis	ionq-11q	74%
$ +++ \rangle$	$CZ_{13}CCZ$	Z basis	ionq-11q	70%
$ +++ \rangle$	$CZ_{12}CZ_{13}CCZ$	Z basis	ionq-11q	74%
$ +++ \rangle$	$CZ_{23}CCZ$	Z basis	ionq-11q	72%
$ +++ \rangle$	$CZ_{12}CZ_{23}CCZ$	Z basis	ionq-11q	73%
$ +++ \rangle$	$CZ_{13}CZ_{23}CCZ$	Z basis	ionq-11q	74%
$ +++ \rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	Z basis	ionq-11q	73%

Table B.1: Post-selection rates for the results shown in Figs. 3.5, 3.6, where the percentage indicates the proportion of shots that were accepted. The average post-selection rate was 75%.

State	Gate	Measurement	Device	Post-selection rate
$ \text{GHZ}\rangle$	I	X basis	ionq-11q	90%
$ \text{GHZ}\rangle$	CZ_{12}	X basis	ionq-11q	86%
$ \text{GHZ}\rangle$	CZ_{13}	X basis	ionq-11q	90%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}$	X basis	ionq-11q	78%
$ \text{GHZ}\rangle$	CZ_{23}	X basis	ionq-11q	83%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}$	X basis	ionq-11q	81%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}$	X basis	ionq-11q	82%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}$	X basis	ionq-11q	79%
$ \text{GHZ}\rangle$	CCZ	X basis	ionq-11q	80%
$ \text{GHZ}\rangle$	$CZ_{12}CCZ$	X basis	ionq-11q	78%
$ \text{GHZ}\rangle$	$CZ_{13}CCZ$	X basis	ionq-11q	82%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CCZ$	X basis	ionq-11q	87%
$ \text{GHZ}\rangle$	$CZ_{23}CCZ$	X basis	ionq-11q	83%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}CCZ$	X basis	ionq-11q	83%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}CCZ$	X basis	ionq-11q	82%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	X basis	ionq-11q	84%
$ \text{GHZ}\rangle$	I	Z basis	ionq-11q	86%
$ \text{GHZ}\rangle$	CZ_{12}	Z basis	ionq-11q	81%
$ \text{GHZ}\rangle$	CZ_{13}	Z basis	ionq-11q	84%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}$	Z basis	ionq-11q	85%
$ \text{GHZ}\rangle$	CZ_{23}	Z basis	ionq-11q	81%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}$	Z basis	ionq-11q	90%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}$	Z basis	ionq-11q	84%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}$	Z basis	ionq-11q	81%
$ \text{GHZ}\rangle$	CCZ	Z basis	ionq-11q	86%
$ \text{GHZ}\rangle$	$CZ_{12}CCZ$	Z basis	ionq-11q	80%
$ \text{GHZ}\rangle$	$CZ_{13}CCZ$	Z basis	ionq-11q	87%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CCZ$	Z basis	ionq-11q	75%
$ \text{GHZ}\rangle$	$CZ_{23}CCZ$	Z basis	ionq-11q	74%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}CCZ$	Z basis	ionq-11q	83%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}CCZ$	Z basis	ionq-11q	75%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	Z basis	ionq-11q	85%

Table B.2: Post-selection rates for the results shown in Figs. 3.4, 3.7 for the IonQ device, where the percentage indicates the proportion of shots that were accepted. The average post-selection rate was 83%.

State	Gate	Measurement	Device	Post-selection rate
$ \text{GHZ}\rangle$	I	X basis	ibm_mumbai	76%
$ \text{GHZ}\rangle$	CZ_{12}	X basis	ibm_mumbai	69%
$ \text{GHZ}\rangle$	CZ_{13}	X basis	ibm_mumbai	72%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}$	X basis	ibm_mumbai	79%
$ \text{GHZ}\rangle$	CZ_{23}	X basis	ibm_mumbai	69%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}$	X basis	ibm_mumbai	76%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}$	X basis	ibm_mumbai	78%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}$	X basis	ibm_mumbai	80%
$ \text{GHZ}\rangle$	CCZ	X basis	ibm_mumbai	72%
$ \text{GHZ}\rangle$	$CZ_{12}CCZ$	X basis	ibm_mumbai	73%
$ \text{GHZ}\rangle$	$CZ_{13}CCZ$	X basis	ibm_mumbai	79%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CCZ$	X basis	ibm_mumbai	76%
$ \text{GHZ}\rangle$	$CZ_{23}CCZ$	X basis	ibm_mumbai	73%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}CCZ$	X basis	ibm_mumbai	79%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}CCZ$	X basis	ibm_mumbai	79%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	X basis	ibm_mumbai	72%
$ \text{GHZ}\rangle$	I	Z basis	ibm_mumbai	74%
$ \text{GHZ}\rangle$	CZ_{12}	Z basis	ibm_mumbai	70%
$ \text{GHZ}\rangle$	CZ_{13}	Z basis	ibm_mumbai	74%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}$	Z basis	ibm_mumbai	77%
$ \text{GHZ}\rangle$	CZ_{23}	Z basis	ibm_mumbai	71%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}$	Z basis	ibm_mumbai	71%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}$	Z basis	ibm_mumbai	75%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}$	Z basis	ibm_mumbai	73%
$ \text{GHZ}\rangle$	CCZ	Z basis	ibm_mumbai	76%
$ \text{GHZ}\rangle$	$CZ_{12}CCZ$	Z basis	ibm_mumbai	78%
$ \text{GHZ}\rangle$	$CZ_{13}CCZ$	Z basis	ibm_mumbai	78%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CCZ$	Z basis	ibm_mumbai	77%
$ \text{GHZ}\rangle$	$CZ_{23}CCZ$	Z basis	ibm_mumbai	78%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{23}CCZ$	Z basis	ibm_mumbai	77%
$ \text{GHZ}\rangle$	$CZ_{13}CZ_{23}CCZ$	Z basis	ibm_mumbai	77%
$ \text{GHZ}\rangle$	$CZ_{12}CZ_{13}CZ_{23}CCZ$	Z basis	ibm_mumbai	71%

Table B.3: Post-selection rates for the results shown in Figs. 3.4, 3.7 for the IBM device, where the percentage indicates the proportion of shots that were accepted. The average post-selection rate was 75%.

IonQ

The **ionq-11q** device has 11 qubits and all-to-all connectivity. The characterization data provided by IonQ for the dates that the demonstrations were carried out are given in Table B.4.

Date	1q fid.	2q fid.	SPAM fid.	T_1	T_2	1q gate	2q gate	Readout	Reset
26/07/23	0.9958	0.9652	0.99752	10 000 s	0.2 s	10 μ s	200 μ s	130 μ s	20 μ s
15/08/23	0.9976	0.9906	0.99752	10 000 s	0.2 s	10 μ s	200 μ s	130 μ s	20 μ s

Table B.4: Characterization data for **ionq-11q**. The demonstrations shown in Figs. 3.5,3.6 were conducted on 26/07/23 and the demonstrations shown in Figs. 3.4,3.7 were conducted on 17/08/23 (the data for 15/08/23 shown in the table is the nearest in time to when the relevant demonstrations were carried out). The fidelities are mean values. SPAM stands for state-preparation and measurement.

IBM

The layout of the `ibm_mumbai` device is shown in Table B.1. The demonstrations were carried out on 14/04/23. The 1-qubit and 2-qubit gate characterization data provided by IBM for this date are given in Tables B.5, B.6. The qubit characterization data provided by IBM for this date are given in Table B.7 and the qubit reset time was 3612 ns for all qubits.

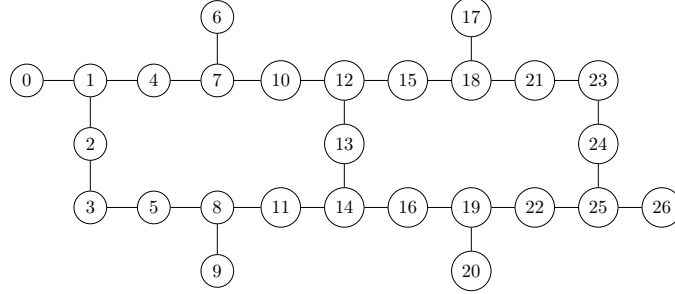


Figure B.1: Layout of the `ibm_mumbai` device. Vertices represent qubits and edges represent the availability of entangling gates between the two endpoints. The mapping from the qubits of the $[[8, 3, 2]]$ code to the qubits of the device was $(q_0, 7)$, $(q_1, 4)$, $(q_2, 1)$, $(q_3, 10)$, $(q_4, 2)$, $(q_5, 12)$, $(q_6, 13)$, $(q_7, 3)$.

Qubit	Gate	Error	Time
0	$I, X, \sqrt{X}$	0.0002951	35.56 ns
1	$I, X, \sqrt{X}$	0.0001877	35.56 ns
2	$I, X, \sqrt{X}$	0.0001816	35.56 ns
3	$I, X, \sqrt{X}$	0.0002697	35.56 ns
4	$I, X, \sqrt{X}$	0.0003707	35.56 ns
5	$I, X, \sqrt{X}$	0.0002801	35.56 ns
6	$I, X, \sqrt{X}$	0.000267	35.56 ns
7	$I, X, \sqrt{X}$	0.0001768	35.56 ns
8	$I, X, \sqrt{X}$	0.0001585	35.56 ns
9	$I, X, \sqrt{X}$	0.0003164	35.56 ns
10	$I, X, \sqrt{X}$	0.0002868	35.56 ns
11	$I, X, \sqrt{X}$	0.0002381	35.56 ns
12	$I, X, \sqrt{X}$	0.0001866	35.56 ns
13	$I, X, \sqrt{X}$	0.0001609	35.56 ns
14	$I, X, \sqrt{X}$	0.000182	35.56 ns
15	$I, X, \sqrt{X}$	0.0004084	35.56 ns
16	$I, X, \sqrt{X}$	0.0001818	35.56 ns
17	$I, X, \sqrt{X}$	0.002198	35.56 ns
18	$I, X, \sqrt{X}$	0.0002243	35.56 ns
19	$I, X, \sqrt{X}$	0.0001976	35.56 ns
20	$I, X, \sqrt{X}$	0.0002066	35.56 ns
21	$I, X, \sqrt{X}$	0.0004652	35.56 ns
22	$I, X, \sqrt{X}$	0.000166	35.56 ns
23	$I, X, \sqrt{X}$	0.0003324	35.56 ns
24	$I, X, \sqrt{X}$	0.0001602	35.56 ns
25	$I, X, \sqrt{X}$	0.0002232	35.56 ns
26	$I, X, \sqrt{X}$	0.0002714	35.56 ns

Table B.5: 1-qubit gate characterization data for `ibm_mumbai` on 14/04/23.

Qubits	Gate	Error	Time
3,2	CNOT	0.008789	433.8 ns
2,3	CNOT	0.008789	469.3 ns
14,11	CNOT	0.005348	391.1 ns
11,14	CNOT	0.005348	426.7 ns
5,8	CNOT	0.01016	604.4 ns
8,5	CNOT	0.01016	640.0 ns
12,13	CNOT	0.00516	547.6 ns
13,12	CNOT	0.00516	583.1 ns
13,14	CNOT	0.004559	320.0 ns
14,13	CNOT	0.004559	355.6 ns
22,19	CNOT	0.004825	327.1 ns
19,22	CNOT	0.004825	362.7 ns
3,5	CNOT	0.01117	476.4 ns
5,3	CNOT	0.01117	512.0 ns
18,21	CNOT	0.01269	497.8 ns
21,18	CNOT	0.01269	533.3 ns
25,22	CNOT	0.005982	448.0 ns
22,25	CNOT	0.005982	483.6 ns
2,1	CNOT	0.01161	704.0 ns
1,2	CNOT	0.01161	739.6 ns
8,11	CNOT	0.01248	604.4 ns
11,8	CNOT	0.01248	640.0 ns
10,12	CNOT	0.007643	604.4 ns
12,10	CNOT	0.007643	640.0 ns
10,7	CNOT	0.007437	398.2 ns
7,10	CNOT	0.007437	433.8 ns
20,19	CNOT	0.005657	369.8 ns
19,20	CNOT	0.005657	405.3 ns
23,21	CNOT	0.008342	362.7 ns
21,23	CNOT	0.008342	398.2 ns
6,7	CNOT	0.007037	248.9 ns
7,6	CNOT	0.007037	284.4 ns
17,18	CNOT	0.01205	248.9 ns
18,17	CNOT	0.01205	284.4 ns
4,7	CNOT	0.009587	604.4 ns
7,4	CNOT	0.009587	640.0 ns

8,9	CNOT	0.008003	604.4 ns
9,8	CNOT	0.008003	640.0 ns
16,19	CNOT	0.01609	682.7 ns
19,16	CNOT	0.01609	718.2 ns
24,23	CNOT	0.01297	604.4 ns
23,24	CNOT	0.01297	640.0 ns
4,1	CNOT	0.006135	312.9 ns
1,4	CNOT	0.006135	348.4 ns
15,18	CNOT	0.006877	305.8 ns
18,15	CNOT	0.006877	341.3 ns
16,14	CNOT	0.005754	291.6 ns
14,16	CNOT	0.005754	327.1 ns
26,25	CNOT	0.006879	312.9 ns
25,26	CNOT	0.006879	348.4 ns
0,1	CNOT	0.007626	419.6 ns
1,0	CNOT	0.007626	455.1 ns
15,12	CNOT	0.006138	369.8 ns
12,15	CNOT	0.006138	405.3 ns
24,25	CNOT	0.007386	433.8 ns
25,24	CNOT	0.007386	469.3 ns

Table B.6: 2-qubit gate characterization data for `ibm_mumbai` on 14/04/23.

Qubit	T_1	T_2	Frequency	Anharmonicity	Readout error	$p(0 1)$	$p(1 0)$	Readout time
0	100.5 μ s	125.9 μ s	5.071 GHz	-0.3285 GHz	0.0435	0.0714	0.0156	3576 ns
1	116.4 μ s	163.8 μ s	4.93 GHz	-0.3313 GHz	0.0401	0.0476	0.0326	3576 ns
2	94.52 μ s	114.8 μ s	4.67 GHz	-0.3369 GHz	0.0153	0.0206	0.01	3576 ns
3	103.2 μ s	168.3 μ s	4.889 GHz	-0.3312 GHz	0.057	0.0622	0.0518	3576 ns
4	105.9 μ s	47.41 μ s	5.021 GHz	-0.3302 GHz	0.03	0.0424	0.0176	3576 ns
5	86.9 μ s	151.6 μ s	4.969 GHz	-0.3298 GHz	0.1603	0.178	0.1426	3576 ns
6	122.5 μ s	81.42 μ s	4.966 GHz	-0.3293 GHz	0.0169	0.0236	0.0102	3576 ns
7	125.2 μ s	164.3 μ s	4.894 GHz	-0.331 GHz	0.0162	0.0232	0.0092	3576 ns
8	212.2 μ s	256.6 μ s	4.792 GHz	-0.3326 GHz	0.018	0.0192	0.0168	3576 ns
9	149.8 μ s	119.8 μ s	4.955 GHz	-0.3306 GHz	0.0119	0.0184	0.0054	3576 ns
10	99.87 μ s	115.1 μ s	4.959 GHz	-0.3309 GHz	0.0211	0.0288	0.0134	3576 ns
11	148.1 μ s	137.1 μ s	4.666 GHz	-0.3326 GHz	0.032	0.0422	0.0218	3576 ns
12	151.6 μ s	249.3 μ s	4.743 GHz	-0.333 GHz	0.0348	0.0414	0.0282	3576 ns
13	192.4 μ s	236.7 μ s	4.889 GHz	-0.3281 GHz	0.0124	0.0184	0.0064	3576 ns
14	156.7 μ s	223.5 μ s	4.78 GHz	-0.3326 GHz	0.0214	0.0276	0.0152	3576 ns
15	111.2 μ s	36.64 μ s	4.858 GHz	-0.3332 GHz	0.024	0.0364	0.0116	3576 ns
16	131.2 μ s	187.7 μ s	4.98 GHz	-0.3299 GHz	0.0634	0.075	0.0518	3576 ns
17	63.8 μ s	114.2 μ s	5.003 GHz	-0.3299 GHz	0.0206	0.0314	0.0098	3576 ns
18	132.7 μ s	172.1 μ s	4.781 GHz	-0.3331 GHz	0.0787	0.0852	0.0722	3576 ns
19	169.2 μ s	209.2 μ s	4.81 GHz	-0.3321 GHz	0.0306	0.0316	0.0296	3576 ns
20	96.24 μ s	218.2 μ s	5.048 GHz	-0.328 GHz	0.0137	0.0226	0.0048	3576 ns
21	105.7 μ s	162.3 μ s	4.943 GHz	-0.3313 GHz	0.0418	0.0666	0.017	3576 ns
22	161.2 μ s	194.7 μ s	4.911 GHz	-0.3318 GHz	0.0145	0.0232	0.0058	3576 ns
23	102.2 μ s	166.8 μ s	4.893 GHz	-0.3315 GHz	0.0471	0.0582	0.036	3576 ns
24	139.5 μ s	41.14 μ s	4.671 GHz	-0.3359 GHz	0.0152	0.0226	0.0078	3576 ns
25	146.9 μ s	182.3 μ s	4.759 GHz	-0.3336 GHz	0.0159	0.0184	0.0134	3576 ns
26	135.8 μ s	244.6 μ s	4.954 GHz	-0.3295 GHz	0.018	0.0214	0.0146	3576 ns

Table B.7: Qubit characterization data for `ibm.mumbai` on 14/04/23. $p(i|j)$ is the probability of measuring a $|i\rangle$ state, given that a $|j\rangle$ state was prepared.

Appendix C

Sampling complexity of quasiprobabilistic algorithms

C.1 Proof of Lemma 1

Proof. Let I be a random variable that takes values in $\{1, \dots, D\}$ with probability

$$\Pr[I = i] = \frac{q_i}{\gamma}. \quad (\text{C.1})$$

Let $L_I \in \{0, 1\}$ be the indicator variable that is 1 when the operation $\mathcal{B}_I$ is successfully implemented, and let Y_I denote the outcome of measuring the observable A . Note that

$$\Pr[L_i = 1] = 1, \text{ if } \mathcal{B}_i \text{ is trace preserving,} \quad (\text{C.2})$$

$$\Pr[L_i = 1] = \text{Tr}[\mathcal{B}_i(\rho)] \leq 1, \text{ if } \mathcal{B}_i \text{ is non-trace preserving.} \quad (\text{C.3})$$

Now, we define the random variables

$$F = \gamma \text{sgn}(q_I) L_I Y_I \quad G = \gamma \text{sgn}(q_I) L_I. \quad (\text{C.4})$$

Their expectations can be computed as

$$\mathbb{E}[F] = \gamma \sum_{i=1}^D \sum_{L_i} \mathbb{E}_{Y_i} \text{sgn}(q_i) \Pr[I = i] \Pr[L_i = 1] Y_i \quad (\text{C.5})$$

$$= \gamma \sum_{i=1}^D \frac{|q_i|}{\gamma} \text{sgn}(q_i) \text{Tr}[A \mathcal{B}_i(\rho)] \quad (\text{C.6})$$

$$= \text{Tr}[A\mathcal{T}(\rho)], \quad (\text{C.7})$$

$$\mathbb{E}[G] = \gamma \sum_{i=1}^D \sum_{L_i} \text{sgn}(q_i) \text{Pr}[I = i] \text{Pr}[L_i = 1] \quad (\text{C.8})$$

$$= \gamma \sum_{i=1}^D \frac{|q_i|}{\gamma} \text{sgn}(q_i) \text{Tr}[\mathcal{B}_i(\rho)] \quad (\text{C.9})$$

$$= \text{Tr}[\mathcal{T}(\rho)], \quad (\text{C.10})$$

Thus we see that the rescaled expectation value is given by

$$\langle A \rangle_{\mathcal{T}} = \frac{\mathbb{E}[F]}{\mathbb{E}[G]} \quad (\text{C.11})$$

proving the correctness of Algorithm 2.

We use standard concentration inequalities to bound the number of measurements needed to estimate $\langle A \rangle_{\mathcal{T}}$ within a given precision. We note that both F and G lie between $\pm\gamma$. For N independent trials (where each trial consists of sampling an index i , running the quantum circuit corresponding to element $\mathcal{B}_i$, post-selecting, and measuring A), we define $\bar{F} = \frac{1}{N}(F_1 + F_2 + \dots + F_N)$ and $\bar{G} = \frac{1}{N}(G_1 + G_2 + \dots + G_N)$. Then, Hoeffding's inequality implies

$$\Pr [|\bar{F} - \mathbb{E}[F]| \geq \epsilon] \leq \exp \left(-\frac{N\epsilon^2}{2\gamma^2} \right) \quad (\text{C.12})$$

$$\Pr [|\bar{G} - \mathbb{E}[G]| \geq \epsilon] \leq \exp \left(-\frac{N\epsilon^2}{2\gamma^2} \right). \quad (\text{C.13})$$

Choosing

$$N = \frac{2\gamma^2}{\epsilon^2} \log(1/\delta) \quad (\text{C.14})$$

ensures that the RHS's of Eq. C.12 are at most δ . Then, we have with probability at least $(1 - \delta)$

$$\left| \frac{\bar{F}}{\bar{G}} - \frac{\mathbb{E}[F]}{\mathbb{E}[G]} \right| = \left| \frac{\mathbb{E}[G](\bar{F} - \mathbb{E}[F]) + \mathbb{E}[F](\mathbb{E}[G] - \bar{G})}{\bar{G}\mathbb{E}[G]} \right| \quad (\text{C.15})$$

$$\leq \left| \frac{\mathbb{E}[G](\bar{F} - \mathbb{E}[F])}{\bar{G}\mathbb{E}[G]} \right| + \left| \frac{\mathbb{E}[F](\mathbb{E}[G] - \bar{G})}{\bar{G}\mathbb{E}[G]} \right| \quad (\text{C.16})$$

$$= \frac{1}{|\bar{G}|} \left(\epsilon + \epsilon \left| \frac{\mathbb{E}[F]}{\mathbb{E}[G]} \right| \right) \quad (\text{C.17})$$

$$= \frac{2\epsilon}{|\bar{G}|} \quad (\text{C.18})$$

where we have used Eqs. C.7, C.10 and that $\frac{\mathbb{E}[F]}{\mathbb{E}[G]} \leq 1$. Moreover

$$\frac{1}{|\bar{G}|} \leq \frac{1}{|\mathbb{E}[G] - \epsilon|} \leq \frac{1}{|\mathbb{E}[G]|} \left(1 + \frac{\epsilon}{|\mathbb{E}[G]|} \right). \quad (\text{C.19})$$

Using Eqs. C.18, C.19 and recalling Eq. C.7 we see that

$$\left| \frac{\bar{F}}{\bar{G}} - \langle A \rangle_{\mathcal{T}} \right| \leq \frac{2\epsilon}{|\text{Tr}[\mathcal{T}(\rho)]|} \cdot \left(1 + \frac{\epsilon}{|\text{Tr}[\mathcal{T}(\rho)]|} \right) \quad (\text{C.20})$$

with probability at least $(1 - \delta)$ for choice of N as in Eq. C.14. Therefore, our algorithm requires the claimed number of measurements to return an estimate within precision $\epsilon \cdot (\text{Tr}[\mathcal{T}(\rho)])^{-1}$ under the condition $\epsilon \leq |\text{Tr}[\mathcal{T}(\rho)]|$. $\square$

C.2 Proof of Lemma 2

Proof. Let us denote the ITE operation of the Hamiltonian H by $T = e^{-\beta H}$ and its 1st order Trotter approximation (see Eq. 4.16) by $P = (e^{-\beta H_1} e^{-\beta H_2} \dots e^{-\beta H_L})$. Let $\mathcal{T}(\cdot)$ and $\mathcal{P}(\cdot)$ denote the corresponding maps. From [142] we know that

$$\|T - P\| = \mathcal{O} \left(\left(\sum_{l=1}^L \|H_l\| \beta \right)^2 e^{\sum_{l=1}^L \|H_l\| \beta} \right) \quad (\text{C.21})$$

$$= \mathcal{O} (c^2 L^2 \beta^2 e^{cL\beta}), \quad (\text{C.22})$$

where $\|H_l\| \leq c$ for each H_l . For an input state ρ , the trotterization error in the estimate of $\text{Tr}[Ae^{-\beta H} \rho e^{-\beta H}]$ for an observable A is given by—

$$|\text{Tr}[AT(\rho)] - \text{Tr}[AP(\rho)]| = |\text{Tr}[AT\rho T] - \text{Tr}[AP\rho P]| \quad (\text{C.23})$$

$$= |\text{Tr}[AT\rho T] - \text{Tr}[AT\rho P] + \text{Tr}[AT\rho P] - \text{Tr}[AP\rho P]| \quad (\text{C.24})$$

$$= |\text{Tr}[AT\rho(T - P)] + \text{Tr}[A(T - P)\rho P]| \quad (\text{C.25})$$

$$\leq |\text{Tr}[AT\rho(T - P)]| + |\text{Tr}[A(T - P)\rho P]| \quad (\text{C.26})$$

$$\leq \|A\| \cdot \|T\| \cdot \|(T - P)\| + \|A\| \cdot \|(T - P)\| \cdot \|P\| \quad (\text{C.27})$$

$$\leq 2\|e^{-\beta H}\| \cdot \mathcal{O}(c^2 L^2 \beta^2 e^{cL\beta}) \quad (\text{C.28})$$

$$= \mathcal{O}(c^2 L^2 \beta^2 e^{2cL\beta}) \quad (\text{C.29})$$

Similarly,

$$|\text{Tr}[\mathcal{T}(\rho)] - \text{Tr}[\mathcal{P}(\rho)]| = \mathcal{O}(c^2 L^2 \beta^2 e^{2cL\beta}) \quad (\text{C.30})$$

Using analysis similar to that in the Eq. C.15 of Proof of Lemma 1, the error in estimating the normalized quantity $\frac{\text{Tr}[Ae^{-\beta H} \rho e^{-\beta H}]}{\text{Tr}[e^{-\beta H} \rho e^{-\beta H}]}$ using the first-order Trotter approximation is

$$\frac{2\epsilon_{\text{trot}}}{\text{Tr}[\mathcal{T}(\rho)]} \left(1 + \frac{\epsilon_{\text{trot}}}{\text{Tr}[\mathcal{T}(\rho)]}\right), \quad (\text{C.31})$$

where $\epsilon_{\text{trot}} = \mathcal{O}(c^2 L^2 \beta^2 e^{2cL\beta})$ □

C.3 Proof of Lemma 3

Proof. The proof essentially follows that of Lemma 1 with more involved notation. First note that for a ξ given by R repeated applications of $\mathcal{T}$ we have

$$\text{Tr}[A\xi(\rho)] = \sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_R=1}^D \text{sgn}(q_{i_1} q_{i_2} \cdots q_{i_R}) |q_{i_1} q_{i_2} \cdots q_{i_R}| \cdot \text{Tr}[A \mathcal{B}_{i_R}(\cdots \mathcal{B}_{i_2}(\mathcal{B}_{i_1}(\rho)))]. \quad (\text{C.32})$$

We define I_{i_r} be a random variable that takes values in $\{1, \dots, D\}$ with probability

$$\Pr[I_{i_r} = i] = \frac{q_i}{\gamma}. \quad (\text{C.33})$$

Similar to the proof of Lemma 1, for each i_r we define $L_{i_r} \in \{0, 1\}$ to be the indicator variable that is 1 when the operation $\mathcal{B}_{i_r}$ is successfully implemented. We have $L_{i_r} = 1$ if $\mathcal{B}_{i_r}$ is trace-preserving and

$$\Pr[L_{i_r} = 1] = \frac{\text{Tr}[\mathcal{B}_{i_r}(\mathcal{B}_{i_{r-1}}(\cdots \mathcal{B}_{i_1}(\rho)))]}{\text{Tr}[\mathcal{B}_{i_{r-1}}(\cdots \mathcal{B}_{i_1}(\rho))]}, \quad (\text{C.34})$$

if $\mathcal{B}_{i_r}$ is non trace-preserving. We let Y be the random variable denoting measurement outcome of the observable A and define

$$F := \gamma^R \sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_R=1}^D \text{sgn}(q_{i_1} q_{i_2} \cdots q_{i_R}) \cdot L_{i_R} \cdots L_{i_2} L_{i_1} Y \quad (\text{C.35})$$

$$G := \gamma^R \sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_R=1}^D \text{sgn}(q_{i_1} q_{i_2} \cdots q_{i_R}) \cdot L_{i_R} \cdots L_{i_2} L_{i_1}. \quad (\text{C.36})$$

The expectation values of F and G are given by

$$\begin{aligned} \mathbb{E}[F] &= \sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_R=1}^D \text{sgn}(q_{i_1} q_{i_2} \cdots q_{i_R}) |q_{i_1} q_{i_2} \cdots q_{i_R}| \cdot \text{Tr}[\mathcal{B}_{i_R} (\cdots \mathcal{B}_{i_2} (\mathcal{B}_{i_1}(\rho)))] \\ &= \text{Tr}[A\xi(\rho)], \end{aligned} \quad (\text{C.37})$$

$$\begin{aligned} \mathbb{E}[G] &= \sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_R=1}^D \text{sgn}(q_{i_1} q_{i_2} \cdots q_{i_R}) |q_{i_1} q_{i_2} \cdots q_{i_R}| \cdot \text{Tr}[\mathcal{B}_{i_R} (\cdots \mathcal{B}_{i_2} (\mathcal{B}_{i_1}(\rho)))] \\ &= \text{Tr}[\xi(\rho)] \end{aligned} \quad (\text{C.38})$$

and thus

$$\langle A \rangle_\xi = \frac{\mathbb{E}[F]}{\mathbb{E}[G]} \quad (\text{C.39})$$

proving correctness of Algorithm 3. Now, both F and G lie between $\pm\gamma^R$ (recall that $\|A\| \leq 1$). For N independent trials (each trial consists of sampling a multiset of indices $\{i_1, i_2, \dots, i_R\}$, running the corresponding quantum circuit and accepting/rejecting, and finally measuring A), we define $\bar{F} = \frac{1}{N}(F_1 + F_2 + \cdots + F_N)$ and $\bar{G} = \frac{1}{N}(G_1 + G_2 + \cdots + G_N)$. Then Hoeffding's inequality implies

$$\Pr [|\bar{F} - \mathbb{E}[F]| \geq \epsilon] \leq \exp \left(-\frac{N\epsilon^2}{2\gamma^{2R}} \right) \quad (\text{C.40})$$

$$\Pr [|\bar{G} - \mathbb{E}[G]| \geq \epsilon] \leq \exp \left(-\frac{N\epsilon^2}{2\gamma^{2R}} \right). \quad (\text{C.41})$$

Following the same analysis as in the proof of Eq. 1 (c.f. Eq. C.14 to Eq. C.20), we see that choosing

$$N = \frac{2\gamma^{2R}}{\epsilon^2} \log(1/\delta) \quad (\text{C.42})$$

ensures that

$$\left| \frac{\bar{F}}{\bar{G}} - \langle A \rangle_\xi \right| \leq \frac{2\epsilon}{|\text{Tr}[\xi(\rho)]|} \times \max \left\{ 1, \frac{\epsilon}{|\text{Tr}[\xi(\rho)]|} \right\} \quad (\text{C.43})$$

with probability at least $(1 - \delta)$.

□